

**SYSTEM DESIGN FOR EXPRESS AIRLINES**

**BY**

**MICHAEL R. FISHER, JR.**

**FLIGHT TRANSPORTATION LABORATORY**

**MASSACHUSETTS INSTITUTE OF TECHNOLOGY**

**OCTOBER 9, 1987**

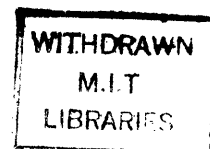
**1**

MASSACHUSETTS INSTITUTE  
OF TECHNOLOGY

FEB 04 1983

LIBRARIES

**Aero**



SYSTEM DESIGN FOR EXPRESS AIRLINES

by

MICHAEL R. FISHER, JR.

B.S., Christian Brothers College, 1975

M.S., Memphis State University, 1977

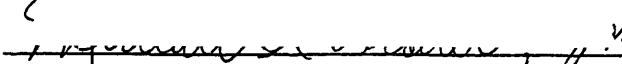
Doctoral Dissertation in Flight Transportation

submitted in partial fulfillment of the requirements for the degree of


DOCTOR OF PHILOSOPHY at the MASSACHUSETTS INSTITUTE OF  
TECHNOLOGY

December 1987

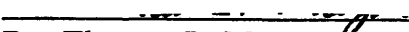
©Michael R. Fisher, Jr., 1987

Signature of Author: 


Department of Aeronautics and Astronautics, December 1987

Certified by: 

Dr. Robert W. Simpson, Thesis Supervisor  
Professor of Aeronautics and Astronautics  
Director, Flight Transportation Laboratory

Certified by: 

Dr. Thomas L. Magnanti  
George Eastman Professor of Management Science,  
Sloan School of Management  
Codirector, Operations Research Center

Certified by: 

Dr. Antony Kong  
Federal Express Corporation

Accepted by: 

Dr. Harold Y. Wachman,  
Professor of Aeronautics and Astronautics  
Chairman, Departmental Graduate Committee

The author hereby grants to M.I.T. permission to reproduce and to distribute copies of this thesis document in whole or in part.

# Contents

<b>1 EXPRESS CARRIERS</b>	<b>5</b>
1.1 Introduction . . . . .	5
1.2 The Single-Hub Single-Turn System . . . . .	7
1.3 Indices and Index Sets . . . . .	10
1.4 Regional Multiple-Hub Systems . . . . .	25
1.5 The Jet Bleed and Trunk Hubs . . . . .	39
1.6 Lower Priority Products . . . . .	43
1.7 Double-Turn Systems . . . . .	47
<b>2 RELATED PROBLEMS AND OPTIMIZATION-BASED SOLUTION METHODS</b>	<b>52</b>
2.1 Transportation . . . . .	52
2.2 Communications . . . . .	55
2.3 Solution Methodologies . . . . .	58
2.4 Lagrangian Relaxation . . . . .	59
2.5 Benders Decomposition . . . . .	61
2.6 Dual Ascent . . . . .	62
2.7 Cutting Planes . . . . .	64
2.8 Summary . . . . .	64
<b>3 THE SINGLE-HUB, SINGLE-TURN PROBLEM</b>	<b>66</b>
3.1 Additional Formulations and Benders Decomposition . . . . .	66
3.2 Building Route Complexes – Some Examples . . . . .	89
3.3 Route Complexes of Higher Order . . . . .	97
<b>4 AN ANALYSIS OF THE COMPLICATING CONSTRAINTS – FORMING A SOLUTION APPROACH</b>	<b>100</b>
4.1 Dualizing the Placement Constraints . . . . .	102
4.2 Setting Up the Matching Problem . . . . .	110
4.3 Constructing the Lagrangian Dual . . . . .	117
4.4 Determining Lagrange Multipliers from the Benders Sub-problem . . . . .	120
4.5 The Aircraft Availability Constraints and the Symmetric Problem . . . . .	126
4.6 The Column-Joining Constraints . . . . .	139
<b>5 COMPUTATIONAL RESULTS, CONCLUSIONS, AND SUGGESTIONS FOR FUTURE RESEARCH</b>	<b>141</b>
5.1 Conclusions and Suggestions for Further Research . . . . .	164
<b>APPENDIX A</b>	<b>169</b>
<b>APPENDIX B</b>	<b>173</b>
<b>APPENDIX C</b>	<b>177</b>
<b>REFERENCES</b>	<b>186</b>

## ACKNOWLEDGEMENTS

There are many people without whose help and guidance this thesis would not have been possible. First, I wish to thank the members of my committee. My thesis supervisor, Professor Robert W. Simpson of the Flight Transportation Laboratory, provided me with an open mind when I wished to pursue unorthodox goals, and a tempering hand to guide me when I needed focus. Professor Thomas L. Magnanti of the Sloan School not only showed a consistent willingness to share his time and thoughts with me, but also demonstrated an enthusiasm for my efforts that singularly helped me to feel that it was all worthwhile. Dr. Antony Kong of the Federal Express Corporation acted well beyond the usual call of duty with his trips to Cambridge from Memphis to sit on my committee.

At Federal Express, I wish to thank two people in particular. Mr. Ted Weise, as a Sr. Vice President, provided me with the necessary encouragement and sanction for my endeavors. Also, I wish to extend a special thanks to Mr. Joe D. Hinson, Managing Director of Operations Research, for conceiving the idea of my sojourn in Cambridge and supporting me throughout. He more than any other made this thesis possible. Thanks also to the many others at Federal Express who shared their ideas with me, including Jose Andrade, Yen Soun, John Murphy, and Darren Smith.

I owe a great debt to Professor Ulrich Derigs of the University of Bayreuth, not only for his interest, advice, and support, but also for graciously allowing me to use his matching code in my computations. I likewise wish to thank Professor James Ho of the University of Tennessee for the use of his revised simplex code.

Heading the list of those who showed patience and support beyond any reasonable limit is my wife Linda, who endured not only my long days of work but also my frequent absences during my extended commute between Cambridge and Memphis. I also wish to thank her for the typing she did for me, making my whole situation much more tenable, and for her stamina in the face of frequently onerous tasks. I hereby promise, "Never again". Also of great help were Abby Crear, not only in the early stages of my thesis, but also throughout my time at M.I.T., and Lyman Hazelton, whose generous help with  $\text{\LaTeX}$  was invaluable. Finally, I wish to thank my parents, for their many sacrifices and anxieties over me, and for never failing to remind me to dress warmly.

I wish to dedicate this thesis to my father, who, when I was young, taught me about electrons and how to meet the ball instead of trying to kill it.

# SYSTEM DESIGN FOR EXPRESS AIRLINES

by

MICHAEL R. FISHER, JR.

Flight Transportation Laboratory  
Massachusetts Institute of Technology

October 9, 1987

## ABSTRACT

In this thesis we investigate and analyze express airlines for the purpose of system design. Chapter 1 contains a taxonomy for express carriers that is built around elemental system components, distinguishable from one another with a two-variable classification scheme. We describe how overnight carriers operate, what their basic philosophy of operation is, and how they might choose to develop their networks to best serve that philosophy. In addition, we present mathematical formulations for several systems.

Chapter 2 is a review of research into similar problems and of solution techniques that might be applicable to express system design problems. In Chapter 3 we focus on the simplest express network problem, the Single-Hub, Single-Turn System Design Problem, SHP. We develop several models for SHP, both to expose the structure of the problem and to find a tractable formulation. The emergent concept of the chapter is the *route complex*. Using this approach to route expression, we choose a formulation that is essentially a set partitioning problem with side constraints.

In Chapter 4 we explore the dualization of the side constraints and develop a solution procedure. There are three types of complicating constraints: aircraft availability, placement (for ferry flights), and column-joining (for transforming a pure set partitioning problem into a nonbipartite matching problem with side constraints). We use a minimum weight, nonbipartite matching problem as the core of our solution procedure for SHP, focusing on obtaining feasible solutions directly from a Lagrangian relaxation, rather than using branch-and-bound. In Chapter 5 we report our computational results and offer suggestions for further research.

# Chapter 1

## EXPRESS CARRIERS

### 1.1 Introduction

In the 1970's a new class of air cargo service evolved in the United States, precipitated by the increasing need for time-critical parcel conveyance. This new industry is now autonomous with respect to all other categories of air transport. It specializes in the pickup and overnight delivery of cargo and is represented by such companies as Federal Express, Emery, Airborne, DHL, Purolator Courier, and United Parcel Service. Federal Express first developed the market, and alone of these firms was incorporated solely for this purpose. The other companies that compete in the time-critical market all entered from other closely related areas, such as air freight forwarding, and continue to maintain their strong presences there. The growth of the industry in terms of annual revenues has been rapid. Federal Express, the industry leader, drew \$3.2 billion in revenues in fiscal 1987, its eleventh year of operation [A3].

A standard measure of productivity in the analysis of the air cargo industry as a whole is ton-miles. Using this gauge as an industry-wide norm today is misleading, however. This is because the overnight services' share of ton-miles is quite low, although their share of the total revenues is substantial (see de Neufville, [D2]). The major reason for this anomaly is that time-critical items are usually very small when compared with the cargo flown by an air freight carrier such as Flying Tigers, which offers a service for relatively time-insensitive items. Therefore, a measure of productiv-

ity that much more accurately reflects the importance of express services within the industry is annual revenues. Because of this, optimal system design for these carriers is essential to the efficient performance of a significant segment of the air transport industry.

The purpose of this paper is to build a foundation for modeling and designing the different types of systems that could facilitate the services offered by this multifaceted, burgeoning segment of flight transportation. We will see that this effort entails addressing some intensely interesting and challenging operations research problems.

Our focus in this chapter will be on systems whose primary purpose is to transport cargo for the highest class of service – that is, overnight carriage and morning delivery of parcels. We properly refer to this as *express* service, but we sometimes use the terms *overnight*, *time-critical*, or *high-priority*. The motivation for this restriction is that we wish to avoid confusion of a pure express system with any other system that may have been originally designed for transporting cargo at a slower rate and is now being used to provide express service as well. This will clarify our exposition.

In keeping with this focus, we will use two types of indicator quantities that will enable us to characterize the express system under consideration as being one of five elemental overnight systems. Any actual system is probably a hybrid of two or more of these or even other, nonexpress, systems, but our classification will allow us to see the basic building blocks from which express systems are likely to be formed. The two indicators are:

1.  $\lambda_i$
2.  $\gamma_{ij}$

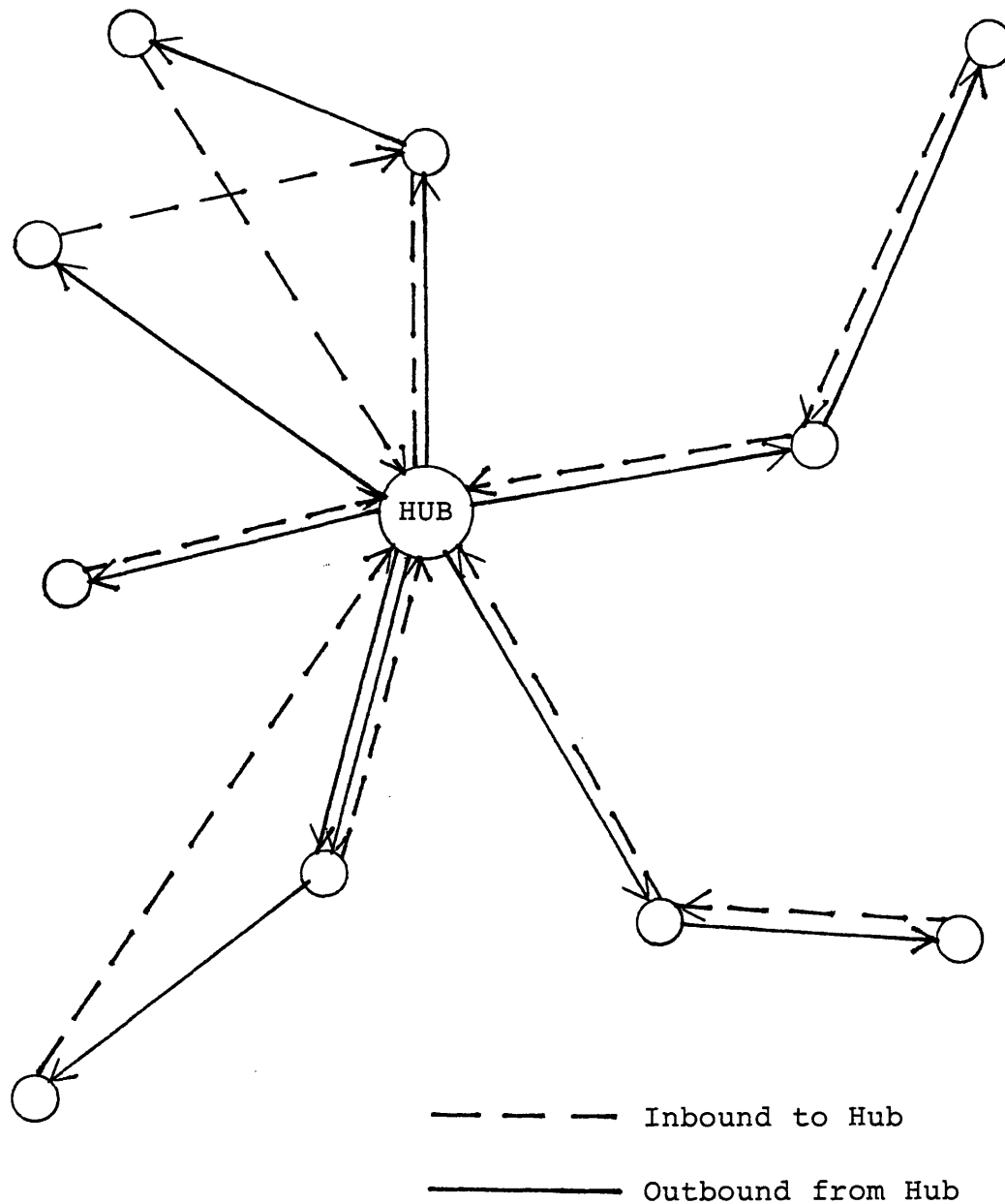
We define the first indicator as the air cargo that flows through airport  $i$ . The quantity  $\gamma_{ij}$  is defined as the air cargo that flows between airport  $i$  and airport  $j$ . We use the term “air cargo” to emphasize that we are concerned with cargo transport through an *airline* network. Thus, we will assume that any cargo that moves by truck or other means has already been removed from consideration. We will sometimes refer to this system characterization scheme as the  $\lambda - \gamma$  method [A1].

## 1.2 The Single-Hub Single-Turn System

We begin our system descriptions with a scenario for the simplest express operation, that of a single *hub*. In this type of system,  $\lambda_i$  is very large for the index  $i$  that denotes the hub. All other indicators are quite small in comparison. Unlike air freight forwarding companies, the express carriers operate their own aircraft, so they can tailor their systems to operate with the lowest possible cost, while maintaining their service commitments. Thus, these companies tend toward the use of hubs as sorting and distribution centers. Typically, for a *single-hub* carrier the location is somewhere near the center of the service region, as weighted by business activity. By using the hub as a break-bulk point, a carrier can serve many more city-pairs than it could afford to do with direct service. For example, the traffic demand between St. Louis and Phoenix might not be large enough to justify a direct flight between the two cities. However, by flying all cargo into a hub for sorting, and then flying it out from the hub to final destinations, the carrier can serve such a city-pair. In fact, if  $n$  cities are in such a system, then all  $n(n-1)/2$  city-pairs can be served with relatively few aircraft. Figure 1-1 shows a conceptual system map.

The simplest single-hub express system operates in the following manner. If someone wishes to ship a package, a courier stops at the customer's place of business, picks up the package, and transports it to an assigned airport later in the day. One of the company's aircraft is parked at the airport, and when enough cargo has arrived, the aircraft is loaded. Later in the evening, this aircraft and the rest of the fleet begin to converge on the hub from all points in the system. At the hub, all the cargo is offloaded from the aircraft and is sorted by destination in a sorting facility. The aircraft are then loaded, and they depart from the hub very early in the morning in order to meet the company's delivery commitment. When an aircraft arrives at an airport in the morning, the appropriate cargo is offloaded and sorted, and trucks and vans transport the parcels to their final destinations. The aircraft stays parked all day at the last stop on its route, until the evening when the whole process is repeated. Thus, the process has a period of one day, or, a *single-turn*.





**Figure 1-1. Conceptual System Map For a High-Priority Carrier**

An aircraft often has more than one stop to make, both when flying into the hub and out of the hub. When flying into the hub, an aircraft only picks up cargo at intermediate stops, and when flying out of the hub, it only drops off cargo. Express carriers employ these restrictions because onloading and offloading cargo in the same stop is not considered to be worth the effort and time that is required. We therefore distinguish between *delivery routes* and *pickup routes*.

Because of the express carrier's delivery commitment, an aircraft on a delivery route must arrive at an airport before a certain time in the morning, so that those communities that are served from the airport can be reached early enough. This time, which may vary from airport to airport, is the *morning or delivery cutoff time*. In the evening, aircraft must wait at an airport long enough for businesses to send as many parcels as they wish. However, the aircraft must also leave the airport soon enough to complete its route and arrive at the hub so that sorting can be performed in time for the morning cutoff times to be met. A balance is therefore required between meeting the restrictions of sorting time and delivery commitment on the one hand and generating as much business as possible on the other hand. Generally speaking, an aircraft will remain at the airport until a prescribed time, before which it cannot leave. This time is the *evening or pickup cutoff time*. Thus, delivery routes are critical with respect to arrival times, and pickup routes are critical with respect to departure times.

We now state the problem that a single-hub system scheduler faces – given an aircraft fleet, a set of airport locations, a set of cutoff times, and a business demand forecast, select the aircraft and design the most efficient routes that meet all the constraints. A fleet planner faces a similar problem – given all of the above except a fleet, determine which fleet produces the most efficient system. This can be accomplished by solving the system scheduler's problem for several different fleet mixes over a predetermined planning horizon, and selecting the best fleet. The system planner must not only deal with route structures and fleet choices, but also with the locations and sizes of facilities throughout the system. We shall investigate and mathematically model several different approaches to system design for

single-hub express carriers. The first of these is the most simple, having one landing and one takeoff per day per aircraft at the hub. We refer to such a system as a *single-hub single-turn system*, SHP. This notation should not be confused with the particular model or formulation of SHP that we may be discussing. In the course of this paper we will present several formulations of SHP. In all cases, those formulations will be denoted in parentheses, e.g. (SDP).

Rewording the design problem, we have the following given a set of aircraft from which to choose, a set of airports, a business demand matrix, a set of cutoff times, and a hub, construct an airline system such that:

1. The demand for pickup and delivery is met at every airport.
2. Aircraft adhere to all cutoff times.
3. All cargo flows through the hub.
4. The system uses only available aircraft.
5. The period of the schedule is one day.
6. The cost is minimized.

We formulate this problem and all other problems in the chapter using a node-arc technique similar to that used for many network design problems (see, for example, Magnanti and Wong [M5], or Gavish [G1]). This method will produce very large and, for the most part, intractable formulations. However, by taking this finely-grained approach, we hope to appreciate the richness of the problems and gain insight into what could be more tractable formulations and solution techniques.

The variables are defined as follows:

### 1.3 Indices and Index Sets

$i, j =$  node indices, indicating airports served (node 0 is the hub)

$k$  = route orientation (time period)

$$k = \begin{cases} 2 & \text{if delivery (morning)} \\ 1 & \text{if pickup (evening)} \end{cases}$$

$I^o = \{0, \dots, n-1\}$  This set represents all airports, including the hub.

$I = \{1, \dots, n-1\}$

$m$  = aircraft index, representing *individual* aircraft. For example,  $m$  could be a tail number.

$T(m)$  = the type of aircraft  $m$ , e.g.; B727-100

$M$  = the set of all aircraft, considered individually

$A$  = the set of aircraft types

### Constants

$n-1$  = number of airports; node  $n$  is an artificial sink

$a_j^k$  = cutoff time at node  $j$  for time period  $k, j = 2, \dots, n-1$ . All times are in minutes after time zero.

$K_m$  = capacity of aircraft  $m$ , in pounds

$c_{ijm}$  = cost of flying aircraft  $m$  from  $i$  to  $j$

$c_{im}$  = cost of locating ground equipment for handling aircraft  $m$  at node  $i$

$\delta_{ij}$  = volume to be sent from  $i$  to  $j$

$d_i^k$  = demand at location  $i$ , time period  $k$

$$= \begin{cases} -\sum_{j=0}^{n-1} \delta_{ji}, k = 2 \\ \sum_{j=0}^{n-1} \delta_{ij}, k = 1 \end{cases}$$

$g_{im}$  = ground time required after landing for aircraft  $m$  at node  $i$

$h^k$  = hub cutoff time in minutes, time period  $k$

$Q_{T(m)}$  = number of aircraft type  $T(m)$  available

$t_{ijm}$  = time required for aircraft type  $T(m)$  to fly from  $i$  to  $j$

### Decision Variables

$a_{im}^k$  = delivery ( $k = 2$ ) or pickup ( $k = 1$ ) departure time of aircraft  $m$ , from node  $i$ , in minutes after time zero

$f_{ijm}^{pqk}$  = flow that originates at  $p$  and is destined for  $q$  that moves from  $i$  to  $j$  on aircraft  $m$  during period  $k$ . Either  $p$  or  $q$  can be the hub.

$f_{ijm}^k = \sum_{p \in I^0} \sum_{q \in I^0} f_{ijm}^{pqk}$  = total flow from  $i$  to  $j$  on aircraft  $m$  during period  $k$

$y_{ijm}^k = \begin{cases} 1, & \text{if aircraft } m \text{ flies from } i \text{ to } j \text{ in time period } k \\ 0, & \text{otherwise.} \end{cases}$

$z_{im} = \begin{cases} 1, & \text{if aircraft } m \text{ lands at node } i. \text{ This decision} \\ & \text{variable models ground equipment necessary for handling the aircraft.} \\ 0, & \text{otherwise.} \end{cases}$

We assume that the demand is in pounds for this formulation, even though, due to containerization, aircraft are bulk, or cube, limited. We shall consider this to have been factored into the capacities  $K_m$ . This assumption is valid because high-priority parcels are usually small and dense enough to make weight the overriding factor. When we discuss the topic of sorting facility capacity, we must consider the demand in terms of numbers of parcels, because this is the way in which these capacities are measured.

We shall interchange the terms “node” and “airport”. Any node other than a hub is a *field* node. Also, when we discuss the topic of sorting facility capacities, we sometimes use the term “hub” to mean the sorting building itself. The context will make the meaning clear.

We set  $y_{inm}^1 \equiv y_{nim}^2 \equiv 0$  for all  $i \in I^0$ , and  $y_{0im}^1 \equiv y_{i0m}^2 \equiv 0$  for  $i \in I$ . We assign these values to ensure that the route structure implied by any

solution to the problem formulation is operationally consistent. We also set  $y_{im}^k \equiv 0$  for all  $i \in \{0, \dots, n\}$ ,  $m \in M$ , and  $k=1, 2$ .

Our formulation, which we designate (SDP), is

$$\text{minimize } \sum_{k=1}^2 \sum_{i \in I^o} \sum_{j \in I^o} \sum_{m \in M} c_{ijm} y_{ijm}^k + \sum_{i \in I} \sum_{m \in M} c_{im} z_{im} \quad (1.1)$$

subject to

$$\sum_{m \in M} \sum_{j=0}^n f_{ijm}^k - \sum_{m \in M} \sum_{j=0}^n f_{jim}^k = d_i^k \quad i \in I, k = 1, 2 \quad (1.2a)$$

$$\sum_{j=0}^n f_{ijm}^{pqk} - \sum_{j=0}^n f_{jim}^{pqk} = \begin{cases} 0, p \neq i, k = 1 \\ 0, q \neq i, k = 2 \end{cases} \quad p, q \in I^o, i \in I, m \in M \quad (1.2b)$$

$$f_{ijm}^k \leq K_m y_{ijm}^k \quad i, j \in I^o \cup \{n\}, m \in M, k = 1, 2 \quad (1.3a)$$

$$y_{ijm}^k \leq z_{jm} \quad i \in I^o \cup \{n\}, j \in I, m \in M, k = 1, 2 \quad (1.3b)$$

$$\sum_{j=0}^n y_{ijm}^k - \sum_{j=0}^n y_{jim}^k = 0 \quad i \in I, m \in M, k = 1, 2 \quad (1.4)$$

$$\sum_{m: T(m)=\alpha} (y_{inm}^2 - y_{nim}^1) = 0 \quad \alpha \in A, i \in I \quad (1.5)$$

$$\sum_{m: T(m)=\alpha} \sum_{i \in I} y_{0im}^2 \leq Q_\alpha \quad \alpha \in A \quad (1.6)$$

$$a_{jm}^k \geq (a_{im}^k + t_{ijm} + g_{jm}) - (1 - y_{ijm}^k) 1440 \quad i \in I^o, j \in I, m \in M, k = 1, 2 \quad (1.7a)$$

$$a_{0m}^2 \geq h^2 \quad m \in M \quad (1.7b)$$

$$a_{im}^1 + t_{i0m} \leq h^1 + (1 - y_{i0m}^1) 1440 \quad i \in I, m \in M \quad (1.7c)$$

$$a_{im}^1 \geq a_i^1 \quad i \in I, m \in M \quad (1.7d)$$

$$a_{im}^2 + t_{ijm} \leq a_j^2 + (1 - y_{ijm}^2) 1440 \quad j \in I, i \in I^o, m \in M \quad (1.7e)$$

$$a_{im}^k, f_{ijm}^{pqk} \geq 0, y_{ijm}^k \in \{0, 1\}, z_{im} \in \{0, 1\} \quad p, q \in I^o, m \in M, k = 1, 2 \quad (1.8)$$

The objective function (1-1) charges an operating cost  $c_{ijm}$  for every

segment  $i - j$  flown by aircraft  $m$ . We note that dealing with aircraft results in  $t_{ijm} \neq t_{jim}$ , due to prevailing winds, and therefore  $c_{ijm} \neq c_{jim}$ , since  $c_{ijm}$  depends on  $t_{ijm}$ . (Typically,  $c_{ijm}$  will be determined by factors such as fuel burn, maintenance expense, and crew cost, all of which are dependent on  $t_{ijm}$ .) Therefore, (SDP) lacks symmetry in this respect. A daily ownership cost is assessed by allocating it to the first segment flown out of the hub. Thus, costs of the form  $c_{0im}$  will be quite high in comparison with other segment costs. A cost  $c_{im}$  is assessed for using aircraft  $m$  at a node. This cost covers the purchase of loading/unloading equipment and other necessary hardware and spare parts.

Constraints (1-2a) are the mass balance equations and provide that the net flow out of a station in the evening ( $k=1$ ) and into the station in the morning ( $k=2$ ) matches the demand. We prevent cargo transfer between aircraft with (1-2b). Constraints (1-3a) are “forcing” constraints and ensure that the cargo flow along a segment does not exceed the total capacity of the aircraft flying that segment. These constraints also ensure that no cargo is flown when no aircraft is available (i.e., when  $y_{ijm}^k = 0$ ). We provide for necessary parts and loading equipment for aircraft  $m$  at node  $i$  with (1-3b).

Constraints (1-4) guarantee that, for each aircraft, morning or evening, the total number flown into a city is equal to the total number flown out of that city. Node  $n$  is used as an artificial sink for all flights. We ensure that the last stops on all delivery routes match the starting points on all pickup routes, both with regard to airport and aircraft type, with constraints (1-5). Constraints (1-6) provide that the total number of available aircraft of type  $T(m)$  is not exceeded. In a fleet planning situation where there is possibly no limit to the number of some particular type available,  $T(m)$  can be set to some very large number.

Inequalities (1-7a) through (1-7e) are all time constraints. Departure times are guaranteed to be physically feasible by (1-7a) (see Golden and Magnanti, [G7]). These constraints set the departure time of aircraft  $m$  from node  $j$  to be at least as great as its departure time from  $i$  (provided the aircraft departed  $i$ , i.e.,  $y_{ijm}^k=1$ ), plus the ground processing time at  $j$ , plus the flying time between  $i$  and  $j$ . If aircraft  $m$  did not land at node  $i$ ,

then the constraint is satisfied for any  $a_{jm}^k \geq 0$ . The number 1440 is used because it is the number of minutes in a day, or the period of the total schedule. This formulation uses ground times that are aircraft-dependent and node-dependent to account for possible variations at different airports. We could have  $g_{im}=g_{jm}$  for all  $i, j$ , and  $m$ , however.

In constraints (1-7b) we provide that the departure time from the hub for an aircraft takes place only after the sort is finished. We ensure that any flight into the hub (e.g.  $y_{i0m}^1 = 1$  for some  $i \neq 1$  and some  $m \in M$ ) leaves  $i$  early enough to arrive before the sort, in constraints (1-7c). As with constraints (1-7a), this inequality is easily satisfied if  $y_{i0m}^1 = 0$ . Constraints (1-7d) and (1-7e) enforce pickup and delivery cutoff times, respectively, for all field nodes. Arrival and departure times and arc flows are non-negative continuous variables, and service arcs are 0-1 variables, as constraints (1-8) stipulate.

In today's domestic express systems, time constraints prohibit routes of many stages. For our application, we will assume that no route longer than three stages is feasible. This assumption will allow us to specialize our solution approach in a (potentially) much more effective way, and for most systems should be a very practical constraint on the problem.

The system that (SDP) models is actually a simplified version of the problem that is faced by the planner. Additional complexities of an express system that (SDP) assumes away are landing and takeoff spacing, a positive lower limit on the numbers of certain aircraft used, and transloading. In general, *transloading* is the transfer of cargo from one aircraft to another at a point away from the hub. Also, we have made some simplifying assumptions about payload-range characteristics and flight plans. We shall assume that all aircraft involved in the model can fly any distance that any (SDP) solution requires, at full payload. This usually should present no difficulty, but it might if small jets are employed, for example. Such aircraft are limited by payload-range factors, and must stop for refueling on long flights. Nonetheless, they have the speed required for meeting cutoff times at distant points, so that without payload-range feasibility constraints, the model could attempt an invalid assignment.



At all times we assume that any cargo that can be carried from origin to destination by truck has been subtracted from the demand matrix  $[\delta_{ij}]$ . We will thus be concerned only with the airside of the system, with two exceptions. In the next section we discuss a special transloading operation that uses *feeder* aircraft. These aircraft are not required to come into the hub. By treating trucks as feeder aircraft, we may include them in this special set. Also, trucks that use the hub as their cargo transfer point, just as aircraft do, may be treated as aircraft.

Our formulation implicitly assumes that the hub has enough sorting capacity to process all cargo in the system. This is not always the case, and capacity must be built into the hub as the system grows. While it is not necessary to include this in our model, more complex system design problems require explicit consideration of sorting capacity. Therefore, we defer a discussion of this topic until we develop these formulations.

A final qualification to our approach is that we will not address the subject of recursively linking periodic (SDP) solutions over a planning horizon. Realistically, we probably would not have a certain fleet one year and a significantly different fleet the next year. Thus, obtaining a set of solutions that fit together over a planning horizon is a very real problem for the planner. Our approach will focus on obtaining a solution for one period only.

Before proceeding with an examination of other systems, we discuss the nature of constraints (1-5), that we will call the *end-node constraints*. The requirement that the last stop on an aircraft's delivery route be the first stop on the same aircraft's pickup route is actually rather restrictive. Although most overnight carriers probably prefer that such a matchup occurs, it might actually be the case that allowing a ferry, or *placement*, flight from the last delivery stop to the pickup route start-node would result in an improved solution. There is time during the day for an aircraft to do exactly that, since it would have several hours of idle time otherwise. To model such a flight, which need not adhere to cutoff times, we introduce a new decision variable,  $x_{ija}$ . We let

$x_{ija}$  = the number (nonnegative integer) of placement flights from node  $i$

to node  $j$  of aircraft type  $a$ .

We now replace constraints (1-5) with (1-5a) and (1-5b):

$$(1-5a) \quad \sum_{j \in I^o} x_{ija} = \sum_{m: T(m)=a} y_{inm}^2 \quad i \in I, a \in A$$

$$(1-5b) \quad \sum_{i \in I^o} x_{ija} = \sum_{m: T(m)=a} y_{njm}^1 \quad j \in I, a \in A$$

Henceforth, we refer to (1-5a) and (1-5b) as the *placement constraints*. If we use placement constraints instead of (1-5) in (SDP), we must add the operating cost of flying aircraft type  $a$  from node  $i$  to node  $j$ ; the new term in the objective function is

$$\sum_{a \in A} \sum_{(i,j) \in I^o \times I^o} c'_{ija} x_{ija}.$$

Our examination of the placement constraints reveals that, given values for the  $y$  variables, the remaining system is essentially a transportation problem for each aircraft type  $a \in A$ . To see this, we form two node sets, one for accommodating delivery flights and one for accommodating pickup flights. Let  $J = I$  and  $J^o = I^o$ . We can then write the placement constraints for a given vector  $\hat{y}$  as

$$\sum_{j \in J^o} x_{ija} = \sum_{m: T_m=a} \hat{y}_{inm}^2 \quad i \in I, a \in A \quad (1-5a)$$

$$\sum_{i \in I^o} x_{ija} = \sum_{m: T_m=a} \hat{y}_{njm}^1 \quad j \in J, a \in A \quad (1-5b)$$

For each pair  $(i, j) \in I \times J$  and each index  $a \in A$ , the variable  $x_{ija}$  appears in exactly two constraints, one of the form (1-5a) and one of the form (1-5b). However, any variable of the form  $x_{ioa}$  or  $x_{oja}$  appears in only one constraint. (We note that this and multiplying (1-5b) by  $-1$  show that the constraint matrix for the  $x_{ija}$  is totally unimodular.) In addition, for any given  $a \in A$ , it is possible that

$$d_a^2 = \sum_{i \in I} \sum_{m: T_m=a} \hat{y}_{inm}^2 \neq \sum_{j \in J} \sum_{m: T_m=a} \hat{y}_{njm}^1 = d_a^1.$$

This would occur, for example, if a placement flight originated at the hub to compensate for a demand imbalance at a node that is a net producer. If not

for these two exceptions, (1-5a) and (1-5b) would form  $|A|$  transportation problems for any given feasible  $y$ -vector. We will now demonstrate that the special structure of our problem allows us to classify this as a set of transportation problems anyway. To do this, we elaborate on how the costs  $c'_{ija}$  are computed for each  $x_{ija}$ , and on what the variable  $x_{ija}$  means physically.

Suppose first that  $i$  and  $j$  represent the same physical location. Then the cost  $c'_{ija} \equiv 0$  for each  $a \in A$ , and setting  $x_{ija} > 0$  simply means that an aircraft of type  $a$  sits on the ground at location  $i=j$  during the idle daytime period. Next, suppose that  $i$  and  $j$  represent different locations, and that neither index denotes the hub. Then  $c'_{ija}$  is the cost of operating aircraft  $a$  between points  $i$  and  $j$ , and  $x_{ija} > 0$  implies that an aircraft of type  $a$  flies from  $i$  to  $j$ . Such a flight could possibly stop at an intermediate location or locations, including the hub. Wherever these intermediate points are, they are chosen so as to minimize the cost of flying from  $i$  to  $j$ .

Now suppose that either  $i$  or  $j$  denotes the hub, but not both. (If both, an aircraft never flies at all, but always sits at the hub.) Suppose first that  $i$  represents the hub. In such a case, we would like for  $x_{ija} > 0$  to imply that we are ferrying a type  $a$  aircraft from the hub to location  $j$ . This is distinct from a placement flight from some other airport to location  $j$  that uses the hub as an intermediate point. We now examine the conditions under which we would make such a flight.

Figure 1-2 depicts a problem of our kind where the nodes on the left represent nodes on the delivery side, and the nodes on the right represent nodes on the pickup side. A node on the pickup side and the node directly opposite it on the delivery side represent the same physical location. A delivery node and a pickup node that are directly opposite each other represent the same physical location. This is a variation of Simpson's [S4] technique of using different nodes to represent the same airport at different times of the day. We consider the problem for aircraft type  $a'$ . The hub is the bottom node in each case, and only nodes where there is a supply or a demand are shown (except the hub). Note that supply exists at the last node of a delivery route, and demand exists at the first node of a pickup

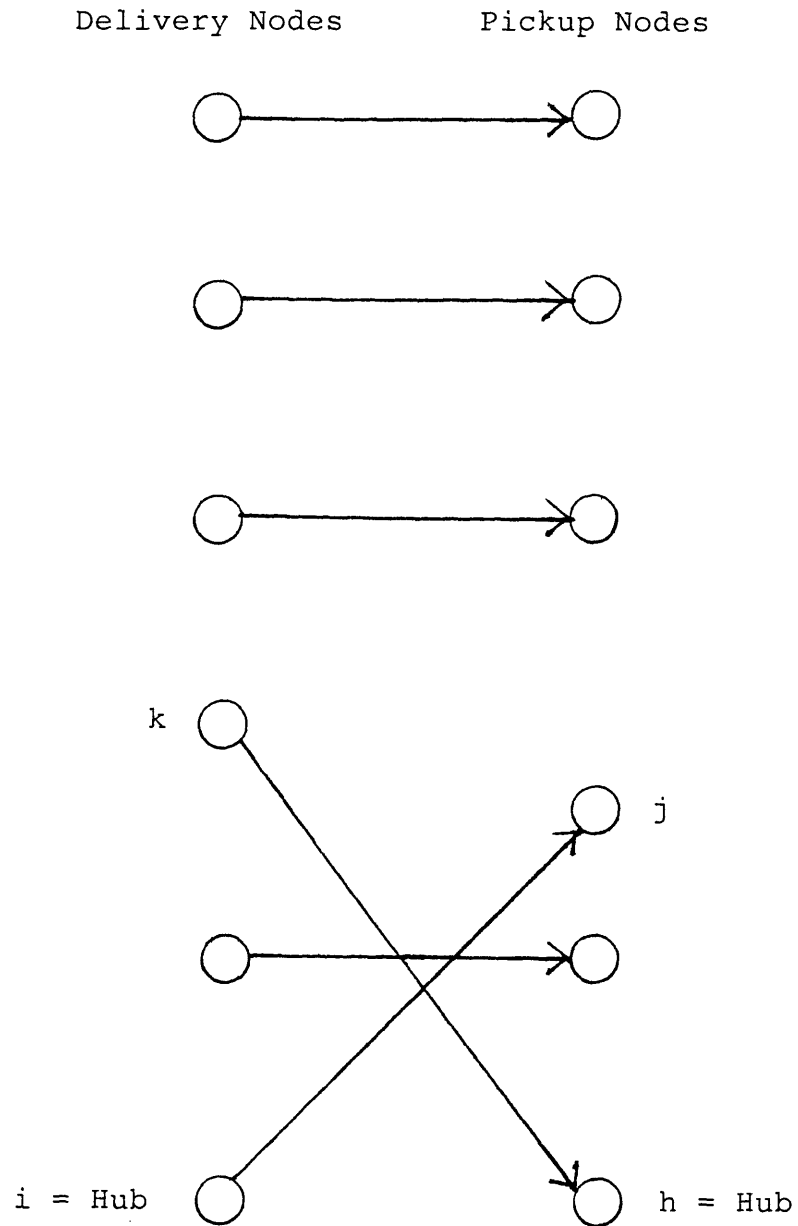
route. Arcs are drawn in where flow actually occurs. A horizontal arc denotes an aircraft remaining on the ground. In case 1, we set  $d_a^2 \geq d_a^1$ . Note that if  $x_{ija'} > 0$ , then there must be a flight from some delivery node  $k$  into the hub in order to satisfy constraints (1-5a), since  $d_a^2 \geq d_a^1$ . Physically, we can interpret this as a single flight from node  $k$  to the hub and then to node  $j$ . But such a flight must be at least as expensive as the cheapest ferry flight from node  $k$  to node  $j$ . Thus, dropping the arcs  $i-j$  and  $k-h$  replacing them with arc  $k-j$  is just as inexpensive.

We would like to guarantee that the cost  $c'_{kja'}$  is strictly less than  $c'_{ija'} + c'_{kha'}$ , where  $i$  represents the hub on the delivery side, and  $h$  represents the hub on the pickup side. We can accomplish this by judicious allocation of ownership costs. Up to this point, all placement flight costs have been operating costs. If we allocate half of the ownership cost of an aircraft to any flight segment in or out of the hub, whether the flight is a placement flight or not, the desired strict inequality will hold. Moreover, this technique gives the model validity and provides a cost-allocation scheme that is at least intuitively appealing, if not practical, from the standpoint of finding a solution. We therefore adopt this cost-allocation methodology.

It follows from the above that no  $x_{ija'}$  will be positive if  $i$  is the hub node index and  $d_a^2 \geq d_a^1$ . Also because  $d_a^2 \geq d_a^1$ , we must have a flow of at least  $d_a^2 - d_a^1$  into the hub node on the pickup side. If this is also the maximum optimal flow into this node, we will have established the desired result for the case  $d_a^2 \geq d_a^1$ . But  $d_a^2 - d_a^1$  must in fact be the maximum flow into the hub on the pickup side, since any more than this amount would require flow out of the hub on the delivery side in order to satisfy the constraints. As we have just noted in the above paragraph, this will not occur in an optimal solution because of our cost allocation. In case 2, we suppose that  $i$  is the hub and  $d_a^2 \leq d_a^1$ . A similar argument yields the desired result. Also, similar arguments yield the same result if node  $j$  is the hub.

From this discussion we see that, given  $\hat{y}$  satisfying all other constraints, we have the following equivalent transportation problem for each  $a \in A$ .

$$\text{minimize } \sum_{i \in I^o} \sum_{j \in J^o} c_{ija} x_{ija}$$



**Figure 1-2.      The Placement Flight Problem For The Case**  
 $d_a^2 \geq d_a^1$

subject to

$$\sum_{j \in J^o} x_{ija} = \sum_{m: T_m=a} \hat{y}_{irm}^2 = d_{ai}^2 \quad i \in I$$

$$\sum_{i \in I^o} x_{ija} = \sum_{m: T_m=a} \hat{y}_{njm}^1 = d_{aj}^2 \quad j \in J$$

$$\sum_{j \in J} x_{0ja} = \max(0, d_a^1 - d_a^2) = d_{a0}^2$$

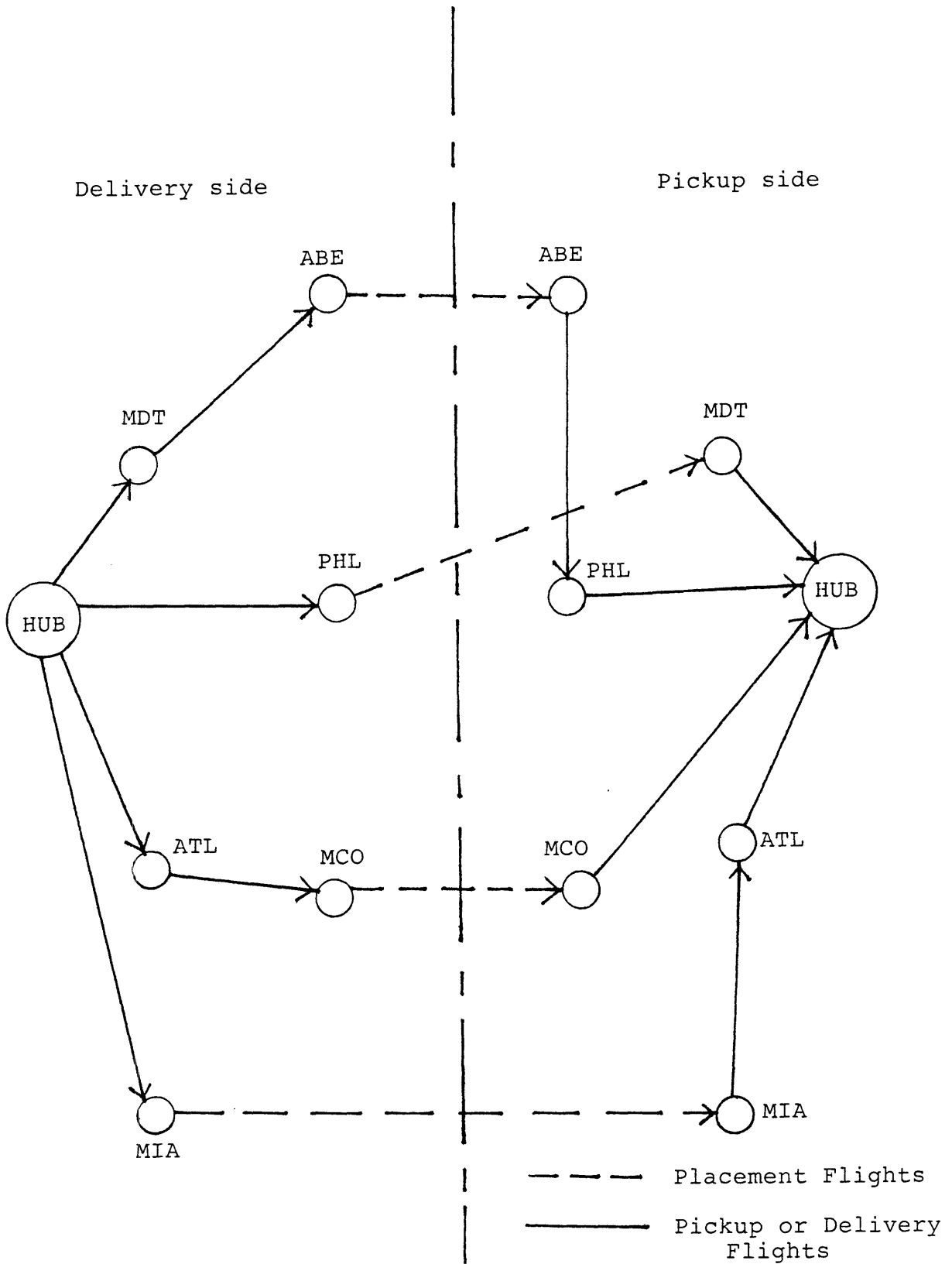
$$\sum_{i \in I} x_{i0a} = \max(0, d_a^2 - d_a^1) = d_{a0}^1$$

Formalizing the result, we have

**Lemma 1.1:** For any feasible  $\hat{y}$ , constraints (1-5a) and 1-5b) form a transportation problem for each aircraft type that is used.

This development affords us an opportunity to decompose (SDP) (indeed, probably most formulations of SHP) into two parts – the “delivery side” and the “pickup side”. (See Figure 1-3.) Independent solutions for each of these problems can then be joined into an overall feasible solution for the entire problem. We will use this fact and the fact that the constraint matrix for the  $x_{ija}$ , given  $\hat{y}$ , is totally unimodular, thus allowing us to drop the integrality requirement, in developing a solution procedure. (We note here that a different formulation of the placement constraints may or may not have the totally unimodular property.) Also, we will use the end-node constraints in the original formulation to construct solutions, and we will compare results.

Additional complexities are possible for the configuration of an express system. We have made some implicit assumptions about the distribution and level of demand across the system that have bearing on the suitability of the model (SDP) for achieving the carrier’s goals. At least some aircraft with jet speed are required to connect all points in the system and to satisfy the cutoff-time restrictions. If all points served offer enough volume to justify the use of a large jet, then (SDP) may indeed be the appropriate model. However, in areas of sparse volume, a superior approach might be to use less expensive turboprops or small jets to aggregate cargo at a facility for further transport by other aircraft. We shall refer to aircraft that serve



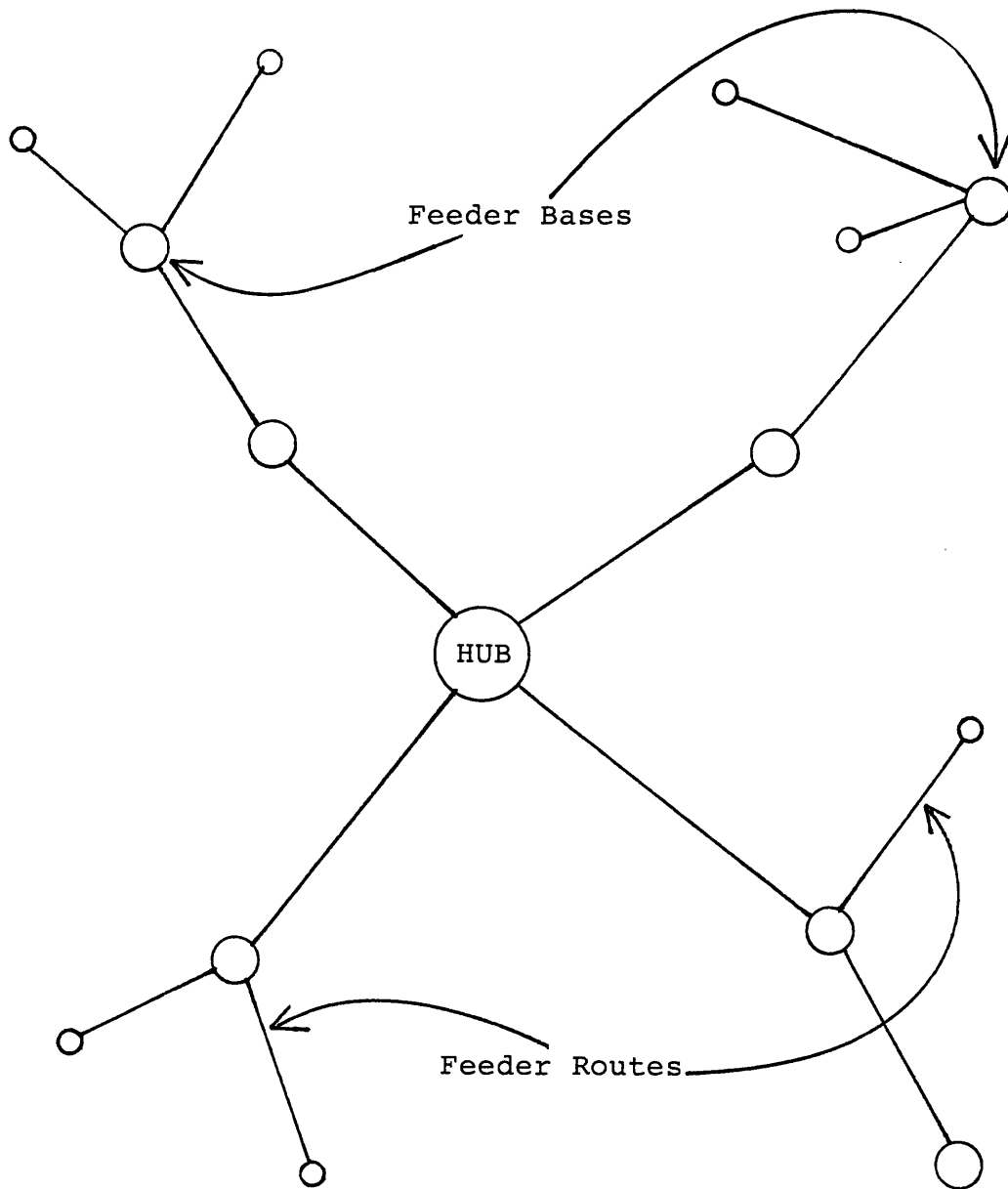
**Figure 1-3. The Two Sides of the Single-Hub Problem**

a region in this fashion as *feeder* aircraft. Figure 1-4 shows a conceptual diagram of a feeder system. All of the constraints that apply to the simple single-hub problem also apply to the system with feeders. In addition, we must consider that aircraft designated as feeders need not fly into the hub. However, the implication of this is that small crew and maintenance bases, sorting and other facilities may need to be established at outlying points. Since the startup cost for such a base could be high, and because the logistics of manning feeder aircraft might require it, the carrier could choose to establish feeder flights based out of a node only if a minimum number of such aircraft are used at that node. Also, the node may have an upper limit to the number of aircraft that it can process. We omit a node-arc formulation for the single-hub feeder system due to its length and complexity, but in essence it is quite like (SDP). For this problem, its intricacies probably dictate starting with a simpler formulation approach.

Relating the feeder system concept to the  $\lambda - \gamma$  classification scheme, we might be unable to detect whether or not an operation employed feeder aircraft by simply observing the relative sizes of the  $\lambda_i$ . This is because a feeder by definition would carry a relatively small amount of cargo to its base. The increment to  $\lambda_i$  (that the total amount of such flow would cause) could be indistinguishable when compared to a nonfeeder flight passing through node  $i$  from even a moderately sized node  $j$ . However, if  $\gamma_{ij}$  is positive for several nodes  $j$ , a feeder base is likely to exist at node  $i$ , since several routes into  $i$  are indicated. Thus, suppose only one  $\lambda_i$  is very large relative to the other  $\lambda_j$  (thus representing a single hub), and for certain nodes  $k$ ,  $\gamma_{kj}$  is positive, (but not exceptionally large) for a few, up to several, indices  $j$ . Then a single hub system with feeders based at the nodes  $k$  likely exists.

As the number of positive  $\gamma_{ij}$ 's grows for some node  $i$ , but the size of such  $\gamma_{ij}$ 's is moderate,  $\lambda_i$  itself grows until it likely becomes distinguishably large relative to other  $\lambda_k$ . In this case, the express system has evolved into a multiple-hub design.





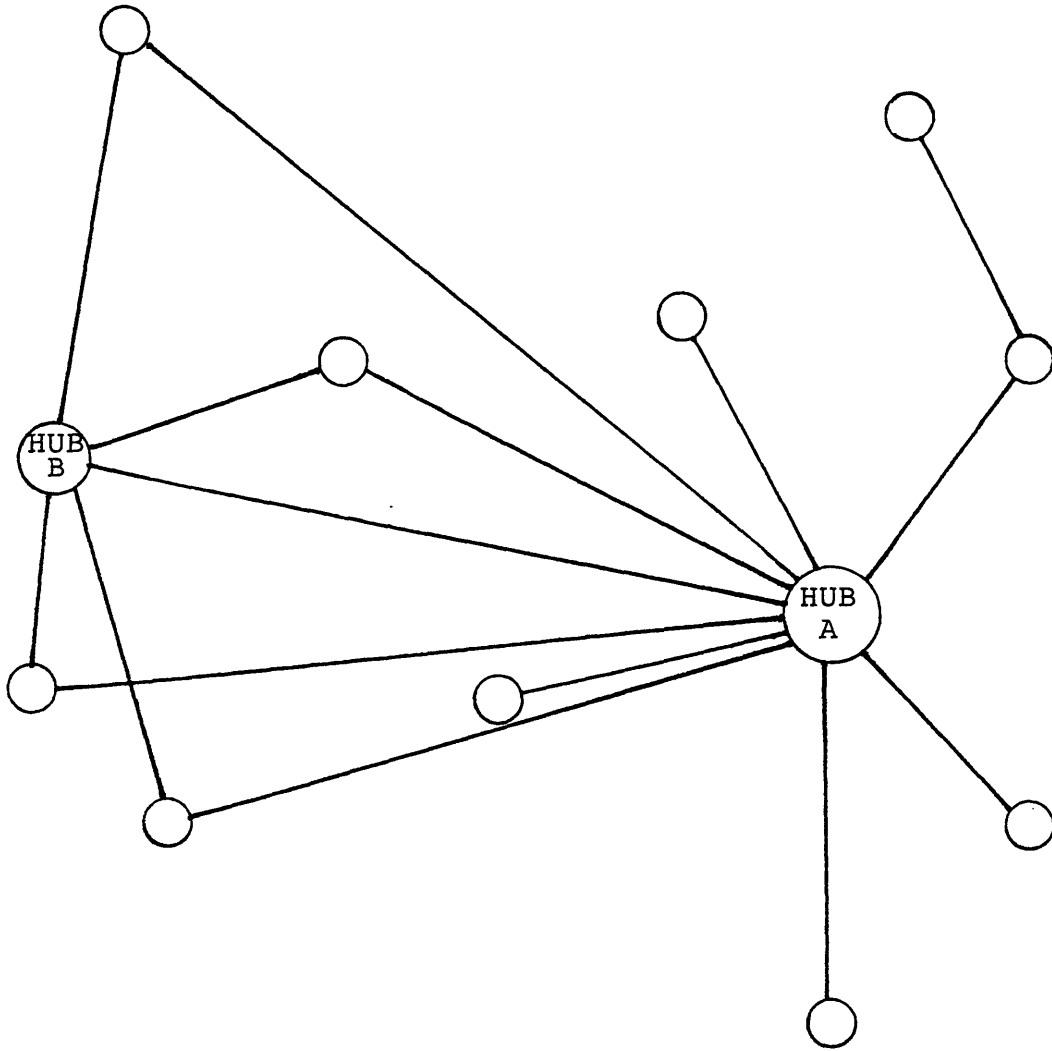
**Figure 1-4. Feeder Concept for a Single-Hub System**

## 1.4 Regional Multiple-Hub Systems

The distribution of demand across a system could be such that a multiple-hub design is preferable to a single hub operation. For example, if the business activity within a region is substantial, establishing and operating a hub that is central to the region could be more cost-effective than flying all of the regional traffic to a single distant hub. Figure 1-5 illustrates a possible multiple-hub design. In this section we develop a model for the Regional Multiple-Hub System Design Problem (RDP). In such a system, one hub is capable of serving all nodes in the network, but one or more regional hubs each serves the nodes of a region, processing cargo that originates in and is destined for points within the same region. The operational constraints that apply to a single-hub also apply to a multiple-hub system. However, additional complexities arise in the formulation of the regional multiple-hub system because any regional hub node is a field node with respect to the principal hub node. Thus, all regional hubs must have their own pickup and delivery cutoff times in addition to their own sort cutoff times. This causes the time constraints to become somewhat more complicated. (In a single-hub system, the hub can be thought of as having pickup and delivery cutoff times equal to the sort departure and sort arrival cutoff times, respectively.)

A further implication of treating hub nodes as field nodes is that a pickup flight can start at a hub node, and a delivery flight can end at a hub node. Indeed, we may have whole flights that only visit hub nodes. In the single-hub formulation, we insisted on end-node matching on an aircraft-type basis at field nodes only. In (RDP), we need to extend the same stipulation to hub nodes as well.

We define  $I_H = \{i : i \text{ is a hub node}\}$ , and  $I_F = \{i : i \text{ is a field node}\}$ . Nodes 0 and  $n$  will be used as artificial sinks. We shall assume at first that all hubs are constructed and are capable of processing all flights and cargo that any solution demands. We wish to



**Figure 1-5. A Possible Two-Hub System. In this system all nodes feed into Hub A. Hub B serves nodes on a regional basis.**

$$\text{minimize } z = \sum_{k \in K} \sum_{i \in I^0} \sum_{j \in I^0} \sum_{m \in M} c_{ijm} y_{ijm}^k + \sum_{i \in I} \sum_{m \in M} c_{im} z_{im} \quad (2-1)$$

subject to

$$\sum_{m \in M} \left( \sum_{j=0}^n f_{ijm}^k - \sum_{j=0}^n f_{jim}^k \right) = d_i^k \quad i \in I \quad (2-2a)$$

$$\sum_{j=0}^n f_{ijm}^{pqk} - \sum_{j=0}^n f_{jim}^{pqk} = \begin{cases} 0, & p \neq i, k=1 \\ 0, & q \neq i, k=2 \end{cases} \quad p, q, i \in I, m \in M \quad (2-2b)$$

$$\sum_{m \in M} \left( f_{i0m}^{pq1} - f_{0im}^{pq2} \right) = 0 \quad p, q \neq i, i \in I_H \quad (2-2c)$$

$$\sum_{p \in I} \sum_{q \in I} f_{ijm}^{pqk} \leq K_m y_{ijm}^k \quad i, j \in I \cup \{0, n\}, m \in M \quad (2-3a)$$

$$y_{ijm}^k \leq z_{jm} \quad i \in I \cup \{0, n\}, j \in I_F, m \in M, k \in K \quad (2-3b)$$

$$\sum_{j=0}^n y_{ijm}^k - \sum_{j=0}^n y_{jim}^k = 0 \quad i \in I, m \in M \quad (2-4)$$

$$\sum_{m: T(m)=\alpha} \left( y_{inm}^2 - y_{nim}^1 \right) = 0 \quad i \in I_F, m \in M \quad (2-5a)$$

$$\sum_{m: T(m)=\alpha} \left( y_{0im}^2 - y_{i0m}^1 \right) = 0 \quad i \in I_H, m \in M \quad (2-5b)$$

$$\sum_{i \in I_F} \sum_{m: T(m)=\alpha} y_{inm}^k \leq Q_\alpha \quad \text{for all distinct } \alpha \quad (2-6a)$$

$$\sum_{i \in I_H} y_{0im}^2 \leq 1 \quad (2-6b)$$

$$a_{jm}^k \geq \left( a_{im}^k + t_{ijm} + g_{jm} \right) - \left( 1 - y_{ijm}^k \right) 1440 \quad i, j \in I, m \in M \quad (2-7a)$$

$$a_{im}^2 \geq h_i^2 - \left( 1 - y_{0im}^2 \right) 1440 \quad i \in I_H, m \in M \quad (2-7b)$$

$$a_{im}^1 + t_{ijm} \leq h_j^1 + (2 - y_{ijm}^1 - y_{i0m}^1) 1440 \quad i \in I, j \in I_H, m \in M \quad (2-7c)$$

$$a_{im}^1 \geq a_i^1 \quad i \in I_F, m \in M \quad (2-7d)$$

$$a_{im}^2 + t_{ijm} \leq a_j^2 + (1 - y_{ijm}^2) 1440 \quad i \in I, j \in I_F, m \in M \quad (2-7e)$$

$$a_{im}^k, f_{ijm}^{pqk} \geq 0, y_{ijm}^k = 0 \text{ or } 1, \quad i, j \in I \cup \{0, n\}, p, q \in I, m \in M, k \in K \quad (2-8)$$

We set  $y_{i0m}^2 \equiv y_{nim}^2 \equiv y_{inm}^1 \equiv y_{i0m}^1 \equiv 0$  for all  $m \in M$  and  $i \in I$ . In the context of our constraint set, the following interpretations of the above identities hold. The first identity says that delivery routes cannot end at node 0 and must not pass through node  $n$ . The third identity states that pickup routes cannot end at node  $n$  and must not pass through node 0. We also set  $y_{i0m}^2 \equiv y_{i0m}^1 \equiv 0$  for  $m \in M$  and  $i \in I_F$ . This identity states that delivery routes cannot begin at a field node and pickup routes cannot end at a field node. Also,  $y_{0nm}^k \equiv y_{n0m}^k \equiv y_{iim}^k \equiv 0$  for all  $i, k$ , and  $m$ .

Constraints (2-2a) guarantee that the net difference in pickup or delivery flow through any node is equal to the demand at that node. Transloading during a pickup route or delivery route is prevented by (2-2b). Constraints (2-2c) provide that the pickup flow of commodity  $pq$  into any hub  $i \left( \sum_{j=0}^n f_{ij}^{pq1} \right)$  equals the delivery flow of commodity  $pq$  out of the same hub  $i \left( \sum_{j=0}^n f_{ij}^{pq2} \right)$ , provided neither  $p$  nor  $q$  is node  $i$ . This physically necessary condition is not guaranteed by (2-2a) or (2-2b), since only pickup flow or only delivery flow appears in any one of these equations. For the single hub problem this condition is implicitly satisfied by constraints (1-2a), which are equivalent to (2-2a) with  $I_H = \{1\}$ .

The forcing constraints (2-3a) and (2-3b) and aircraft conservation constraints (2-4) are the same as for SDP. Constraints (2-5a) ensure that an aircraft begins a pickup route at node  $i$  if and only if an aircraft of same type ends a delivery route at node  $i$ . We enforce a complementary condition for hubs with constraints (2-5b). These state that a pickup route ends

at a hub if and only if an aircraft of the same type begins a delivery route out of that hub. The fleet constraints (2-6a) and (2-6b) are essentially the same as for (SDPF).

We now consider the time constraints (2-7a) through (2-7e). The cycle-breaking constraints (2-7a) are exactly the same as (1-7a). The hub cutoff time constraints (2-7b) and (2-7c) each have extra terms to account for the differences between sort cutoff times and pickup and delivery cutoff times. In (2-7b) the term  $(1 - y_{oim}^2) 1440$  implies that an aircraft on a delivery route has to wait to depart until the sort-down time for that hub only if it begins its route from that hub (i.e.,  $y_{oim}^2 = 1$ ). Thus, an aircraft may use some hub node as an intermediate stop on a delivery route without adhering to the hub's departure cutoff time. In all probability, however, if an aircraft had to wait to depart until the sort-down time at the origin node  $A$  of a delivery route, it could never manage to fly to another hub  $B$ , be processed, and be ready to take off before the sort-down time at  $B$ . This is especially true when one considers that, because as much cargo as is possible is trucked, hubs are quite likely to be very far apart. It is thus probable that we could write (2-7b) in exactly the same form as (1-7b) without unduly constraining the system.

The hub cutoff time constraints (2-7c) require that an aircraft must leave node  $i$  in time to meet the cutoff time at hub  $j$  only if it is flying the segment  $i - j$  (i.e.,  $y_{ijm}^1 = 1$ ) and ending its pickup route at hub  $j$  (i.e.,  $y_{jom}^1 = 1$ ). Constraints (2-7d) and (2-7e) behave exactly as do (1-7d) and (1-7e). We require that all pickup and delivery cutoff times be adhered to, regardless of node type.

Some remarks concerning hub cutoff times and hub capacities are now in order. We once again consider the single-hub, single-turn system, with a given, built-in, sorting capacity. In a SHP, we know a priori exactly how much cargo has to be handled by the sorting facility. Given cutoff times for the hub, it is easy to determine how much capacity is needed to sort the known amount of cargo. We express capacity as pieces sorted per hour. Fortunately, the nature of the simple single-hub system in the United States and the business commitment itself provide us with "good" values for  $h^1$

and  $h^2$ , as we will see.

In order to meet the morning cutoff time at some of the extreme points, aircraft will have to leave by  $h^2$  and fly nonstop to these cities. The easternmost cities are potentially the most constraining in setting  $h^2$  because of time zone considerations – when flying east we “lose” time, while when flying west we “gain” time. Some of the most important eastern cities, such as New York, Miami, and Boston, are also the farthest away from a centrally located hub. Thus, if we choose  $h^2$  such that these points can be reached in time by a nonstop flight, many of the interior nodes in the system can be transit nodes on multiple-stop flights.

Now we consider setting hub cutoff time  $h^1$ . The westernmost cities are critical in this regard, because the time loss effect due to flying east forces these aircraft to leave much earlier in the day, locally, than aircraft in more easterly time zones. However, setting  $h^1$  is not quite as clear-cut as setting  $h^2$ . As we noted in the beginning of the chapter, there is a tradeoff between leaving the farthest nodes late and straining the capacity of the hub, and leaving early and losing business. Generally speaking, aircraft will depart as late as possible without straining the hub “too much”. An implication of this is that all flights must wait until some time at which it is deemed permissible to leave. For distant cities, leaving any later than this puts too great a burden on the hub’s resources, and aircraft at these points will leave at exactly the cutoff time.

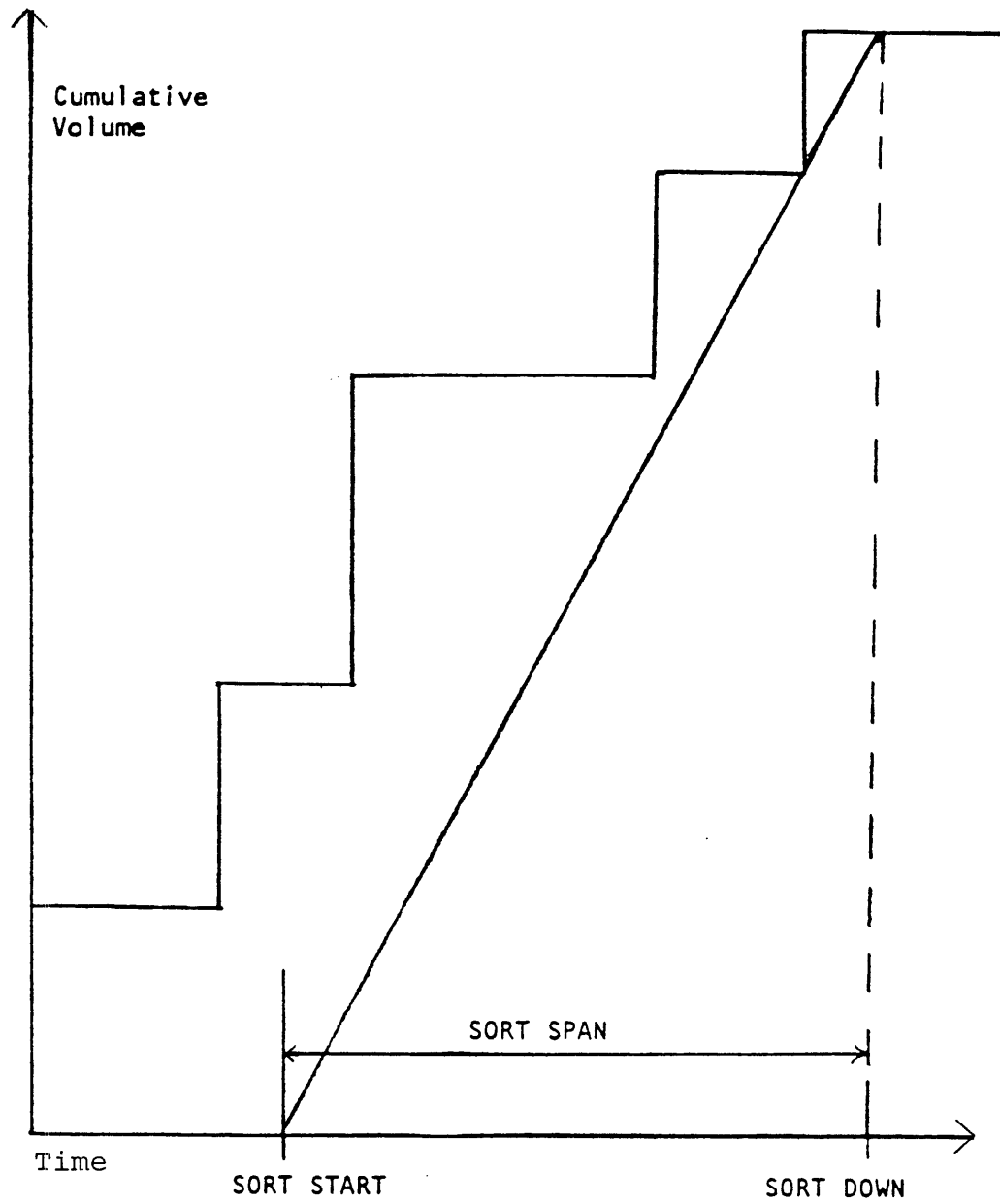
As the system matures, cutoff times can become quite fixed. There are two reasons for this. First, customers become used to a given cutoff time and will be lost if earlier departure times are attempted. Second, because of this, encroachment on the hub sort time is somewhat irreversible, and when the absolute limit to constructible sorting capacity is reached, the system can handle no more traffic through the hub. Since a very high service level is the real product of the carrier, turning away business for any reason is singularly injurious. The hub will therefore wish to maintain a sort capacity at some maximum level, and force schedulers to respect this. Once we know this level,  $h^1$  can be set. In determining  $h^1$ , we must consider the arrival rate of cargo at the sorting facility.

Figure 1-6 shows a step function depicting a given cumulative availability of packages to be sorted versus time. Note that this is not a cumulative graph for arrival of cargo at the airport. The capacity planner must take into account the different unloading times for various aircraft. For example, a DC10-10 and a B727-100 arriving at the airport at the same time with the same payload might appear at different points on the graph because a DC10-10 takes longer to unload, thus delaying the availability of its cargo to be sorted. Graphs of this type are called *volume availability graphs* [M10].

The line in Figure 1-6 that intersects the time axis at the point marked *sort-start* determines the minimum capacity needed to finish by the time marked *sort-down*. The time from *sort-start* to *sort-down* is the *sort span*. The *sort-down* time is set by  $h^2$ , since aircraft must be loaded in time for the morning launch. However, the *sort-start* is actually determined by the slanted line. Notice that the slope of this line is a measure of the change in total packages available to the facility over the change in time, or, in other words, a desired sort rate. The line is determined as shown in the next figure. A vertical line is extended upward from the *sort-down* point. Consider the point where this vertical line intersects the graph of the step function, at point *B*. Imagine the line as hanging from this point, and swing it toward the steps at *A*, as shown in Figure 1-7. The slope of the slanted line *AB* thus formed is the minimum capacity needed to meet the sorting requirements of the given volume availability pattern. To see this, we consider the nature of the volume availability graph.

From Figure 1-7 we can see that the slope of line *AB* is equal to the total volume delivered to the facility divided by the sort span. If we process packages at this rate, we can finish the sort by the required *sort-down* time. However, this is possible only because the packages are available to be sorted at this time. Suppose we now swing our line farther into the step function to where the dotted line is in Figure 1-7. As can be seen, we now have an earlier *sort-start* time and a lower rate or needed sort capacity. However, there are periods of time when no packages are available for sorting. If we take the slope of this line as the sort capacity, we will not be able to finish





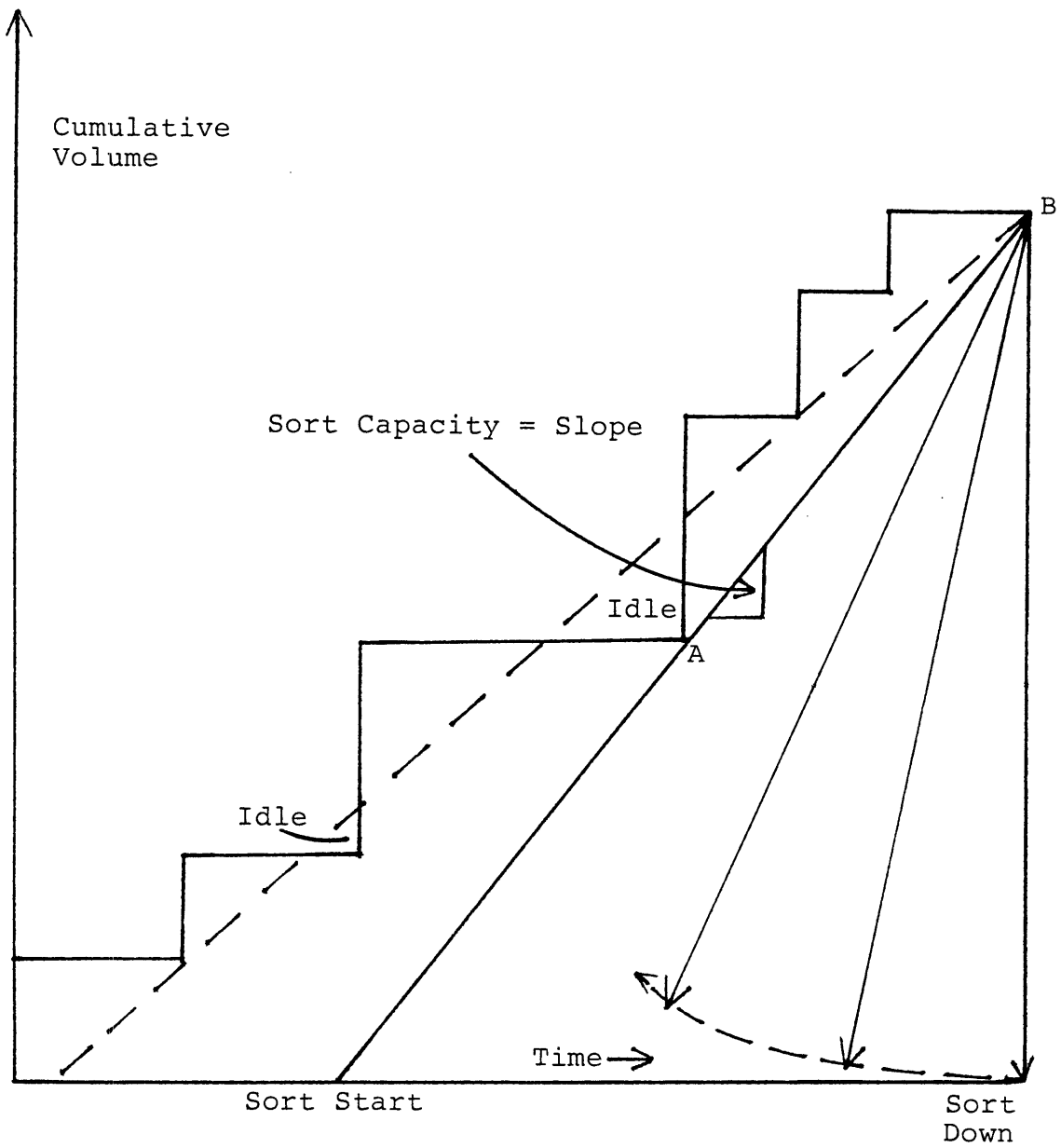
**Figure 1-6. Volume Availability Graph**

the sort on time, because then our capacity will be such that we must sort throughout the span in order to finish with the total demand on time. But the only time we can sort is if packages are available. Thus, no matter how the facility is constructed, we must be able to sort packages at a rate that is, on the average, no less than the slope of  $AB$ . Because the sort capacity calculated in this manner is the minimum required, we must consider what happens if cargo is late. Should this happen, it is possible that the sort facility will be idle. If we have built only the minimum required capacity into the facility, then we will not be able to recover. We must therefore allow for this occurrence and build extra capacity into the facility. This extra percentage of capacity is called the *peaking factor*. A rule of thumb that is used is to add 20% capacity. The same sort-start time is used, and packages are always sorted if present. In the event of late arrivals, the peaking factor will allow us to catch up.

It is important to note that the above discussion applies equally well to a system that uses trucks, vans, or any mix of delivery vehicles. The reasoning relies on the arrival times of packages at the facility and the required departure times, not on the system fleet mix or route structure away from the facility.

In this paper, we will assume that the sort-down time  $s_D$  is the same as the hub departure cutoff time, minus some small constant  $t_D$  (say, 15 minutes) required to finish loading at least some aircraft to begin the morning launch. By our discussion, we cannot make such a straightforward assumption for the hub arrival cutoff time relative to the sort-start time, since aircraft can arrive at the hub significantly later than the sort-start. However, if we are attempting to relate sort span to our model, it is helpful to establish a relationship between the sort-start and hub arrival cutoff time. Thus, we will assume that the hub arrival cutoff time  $h^1$  is equal to the sort-start time  $s_S$  plus a constant  $t_A$ , which could be on the order of 120 minutes. We can treat  $t_A$  as a constant and solve (SDP). By varying  $t_A$  over a small range of values, and then solving (SDP), we can evaluate which choice is the best.

The previous discussion involving volume availability shows that arrivals



**Figure 1-7. Finding the Minimum Required Sort Capacity**

at the hub need to be spread out over some time period in order for the sort to function properly. Also, departures from the hub must be staggered due to airport capacity limitations. We will not address this facet of the problem in this thesis. Instead, we will assume that these conditions are met sufficiently well by any solution.

We have assumed a given cumulative cargo arrival pattern, but in actuality any cumulative rate that produces a line  $AB$  whose slope is acceptably less than the hub's designed sorting capacity will suffice. The important concept is that, given  $h^2$  and the existing sort capacity,  $h^1$  can be determined. Suppose now that we have some latitude in designing the sort capacity. For a simple single-hub system, limits on the necessary sorting capacity at the hub are easily determined with a given  $h^2 - h^1$ , from considerations in our foregoing discussion. However, in a multiple-hub system, we may have no a priori knowledge of how much cargo must flow through a hub. Therefore, knowledge of  $h_i^k$  cannot be used to determine the necessary sort capacity if none or very little exists, and we may wish to test this capacity as a design variable. If we do not know the minimum allowable level for the sort spans in a multiple-hub system, we may also wish to consider the  $h_i^k$  to be design variables. Alternatively, we might wish to set cargo flow levels at some fixed  $\lambda_i$  and calculate a sort capacity from this value. We may have a good idea of approximately what  $\lambda_i$  should be based on gross demand distribution figures.

We first discuss the topic of designing the capacity of a hub, given values for the  $h_i^k$ . We may broadly decompose the cost for capacity into two components. The first of these is the startup cost associated with a facility. This includes possibly buying or leasing land, constructing the initial facility, constructing ramps or runways, and staffing. The second component is that of the cost of improvements to the hub. This cost is not a continuous function, because capacity improvements take the form of discrete construction or machinery additions.

We divide capacity itself into two contributing components – those from the sorting facility and those from ramps and runways. The contribution from the sorting facility is easily translated into sorting capacity. We need

only consider that sorting capacity is expressed in terms of pieces/hour. Therefore, we divide the flow for each commodity at a hub by the average weight-per-piece for that commodity. Unfortunately, there is no direct translation of ramp space into cargo-handling capability. As an extreme example, 10 Dassault-20 Falcon Fanjets require more ramp space than a McDonnell Douglas DC10-30, yet combined they can carry only about half the revenue payload. (The related topic of runway capacity is potentially much more complex, and we will not discuss this here.) We can achieve the desired expression of ramp space, in addition to those that relate cargo flow to sorting capacity. We give the following definitions.

#### Constants

$A_i$  = present parking area at hub  $i$

$B_i$  = the sort capacity at hub  $i$ , multiplied by the sort span

$N_i^R$  = total number of ramp improvements possible or under consideration

$N_i^S$  = total number of sorting facility improvements possible or under consideration

$A_{i\alpha}$  = parking area increase from improvement  $\alpha$  at hub  $i$

$B_{i\beta}$  = sort capacity increase resulting from improvement  $\beta$  at hub  $i$

$c_{i\alpha}^R$  = cost of ramp improvement  $\alpha$  at hub  $i$

$c_{i\beta}^S$  = cost of sorting improvement  $\beta$  at hub  $i$

$P_m$  = total space needed for parking aircraft  $m$

$W_{pq}$  = the average weight per piece of commodity  $pq$

#### Decision Variables

$$z_{i\alpha}^R = \begin{cases} 1 & \text{if ramp improvement } \alpha \text{ is made at hub } i \\ 0 & \text{if not} \end{cases}$$

$$z_{i\beta}^S = \begin{cases} 1 & \text{if sorting improvement } \beta \text{ is made at hub } i \\ 0 & \text{if not} \end{cases}$$

The decision variables  $z_{i\alpha}^R$  and  $z_{i\beta}^S$  could represent improvements that are mutually exclusive. For example, one hub improvement might be to build an additional sort building and fill half of the space inside with sorting equipment. The additional space could be filled with equipment later, thus freeing funds for interim investments. Another improvement might be to build the same additional building and fill it completely with sorting equipment. Clearly, both improvements will not be made. We shall insist that at most one  $z_{i\alpha}^R$  and at most one  $z_{i\beta}^S$  is chosen, and that these variables represent configurations that are realizable.

We can now state the regional hub problem with improvements. We wish to

$$\begin{aligned} \text{minimize } z = & \sum_{k=1}^2 \sum_{i,j \in I} \sum_{m \in M} c_{ijm} y_{ijm}^k + \sum_{i \in I_F} \sum_{m \in M} c_{im} z_{im} + \\ & \sum_{i \in I_H} \left( \sum_{\alpha=1}^{N_i^R} c_{i\alpha}^R z_{i\alpha}^R + \sum_{\beta=1}^{N_i^S} c_{i\beta}^S z_{i\beta}^S \right) \end{aligned}$$

subject to (2-2) through (2-7e), and

$$a_{im}^k, f_{ijm}^{pqk} \geq 0, y_{ijm}^k, z_{i\gamma}^R, z_{i\beta}^S = 0 \text{ or } 1 \quad (2-8')$$

$$\sum_{m \in M} \sum_{j \in I} P_m y_{ijm}^1 \leq A_i + \sum_{\alpha=1}^{N_i^R} A_{i\alpha} z_{i\alpha}^R \quad i \in I_H \quad (2-9a)$$

$$\sum_{p \in I} \sum_{q \in I} \sum_{m \in M} \frac{f_{0im}^{pq1}}{W_{pq}} \leq B_i + \sum_{\beta=1}^{N_i^S} B_{i\beta} z_{i\beta}^S \quad i \in I_H \quad (2-9b)$$

$$\sum_{\alpha=1}^{N_i^R} z_{i\alpha}^R \leq 1 \quad i \in I_H \quad (2-10a)$$

$$\sum_{\beta=1}^{N_i^S} z_{i\beta}^S \leq 1 \quad i \in I_H \quad (2-10b)$$

Constraints (2-9a) are the forcing constraints for relating parking requirements to available ramp space. As can be seen, as long as the total parking requirements are less than the available space, we need not add any improvements (i.e.,  $\sum_{\gamma=1}^{N_i^R} z_{i\gamma}^R = 0$ ). We must begin considering improvements, however, as soon as  $A_i$  is exceeded for any hub  $i$ . It is possible that

$A_i = 0$  for some cases, where we are testing a node  $i$  for its potential benefit as a hub, and have not yet begun operations there. A similar discussion applies to constraints (2-9b), where we relate cargo flow to sorting capacity.

We note that constraints (2-9a) are an approximation to an actual situation. In order for the constraints to be physically valid, the regions of available and potential ramp space must be "reasonably" convex, so that if the constraint is satisfied, then the aircraft selected can, in fact, be parked. We shall assume that this is the case. Constraints (2-10a) and (2-10b) enforce any mutual exclusiveness relationships among improvements for parking and sorting, respectively.

The addition of these constraints to (RDP) creates a very imposing formulation. However, there is the strong likelihood that the number of potential improvements at any hub is small. Moreover, the number of potential hubs itself is likely to be small. It is therefore a reasonable hope that the problem is manageable and that a detailed analysis can be carried out for all or nearly all possibilities.

We are now in a position to treat the hub cutoff times  $h_i^k$  as decision variables. The capacities in constraints (2-9a) are expressed in terms of packages. We must therefore first know the sort span  $s_D^i - s_S^i$  before we can determine these constants from the sort rates that are given by the various improvements under consideration. Since we no longer have the sort span as a given value, we must use the rates given by facility improvements to express constraints (2-9b). Let  $R_i$  be the presently-existing sort rate available at hub  $i$ , and let  $R_{i\beta}^S$  be the rates available from improvements. We have

$$\sum_{p \in I} \sum_{q \in I} \sum_{m \in M} f_{oim}^{pq1} \leq (s_D^i - s_S^i) \left[ R_i + \sum_{\beta=1}^{N_i^S} R_{i\beta}^S z_{i\beta}^S \right] \quad (2-9b')$$

$$h_i^1 = s_S^i + t_A \quad (2-9c')$$

$$h_i^2 = s_D^i + t_D \quad (2-9d')$$

With this modification, we can both relate cargo flow to sort capacity and express the  $h_i^k$  as decision variables. Using (2-9b') we can shrink  $s_D^i - s_S^i$  for any hub  $i$  and properly assess the cost for doing so, or if the time

constraints permit, we can enlarge the span and possibly forestall a costly improvement. The hub cutoff times expressions (2-9c') and (2-9d') derive from our earlier discussion on volume availability.

We have yet to discuss the matter of airport capacity. Usually the only nodes of concern are hubs, since other airports will have very few aircraft flying in and out. Moreover, these flights are at off-hours so that obtaining a slot is not too difficult. Fortunately, hubs are used in the middle of the night, so that the carrier can essentially land and take off freely. However, it is possible that the carrier's operations reach the airport's capacity limits even though they are virtually the only ones. In this event, the carrier may wish to consider runway and taxiway additions or improvements. This technique could be used as an alternative to opening a new hub.

Reformulating (SDP) or (RDP) to include runway improvements is much more complicated than allowing for parking or sorting facility improvements. In this case, we must invoke separation rules for takeoffs and landings to relate airport capacity to the schedule. We do not include a formulation for this problem. However, the number of possible runway additions at a hub is quite likely to be very small, or even zero. Thus, we may be able to enumerate the possibilities and analyze each separately.

This concludes our discussion of the Regional Multiple-Hub System Design Problem. We have considered (RDP) with hub cutoff times given and sufficient sorting capacity, parking space, and airport capacity assumed. We then reformulated the problem using design variables for sorting capacity and ramp space. Next we recast the sorting capacity constraints to allow hub cutoff times to be decision variables, and finally discussed the introduction of runway design variables. We now consider a method of dealing with dense systems that departs from the regional hub design strategy.

## 1.5 The Jet Bleed and Trunk Hubs

When the entries of the business demand matrix  $[\delta_{ij}]$  grow large relative to the capacities of available aircraft, it becomes attractive to consider

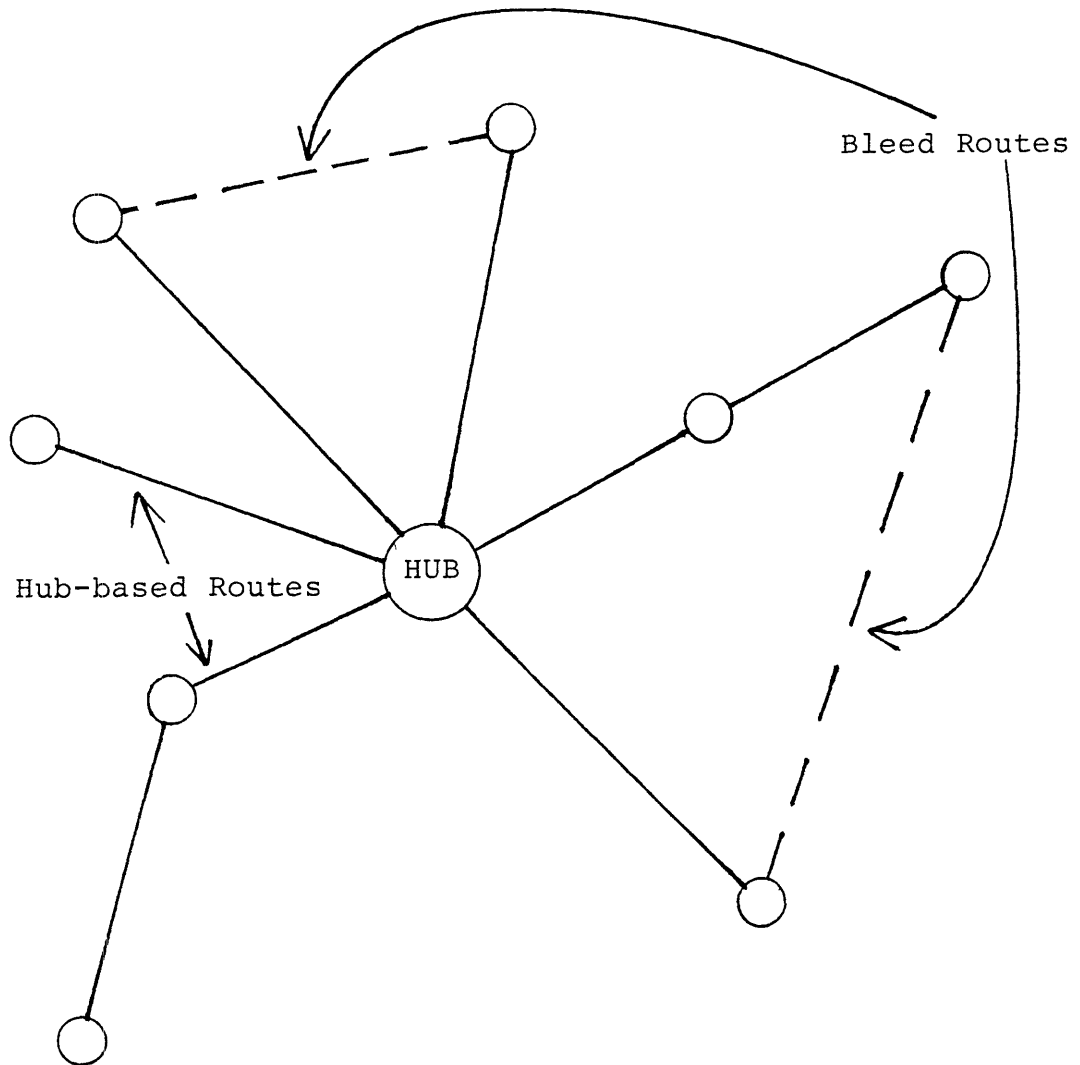


the use of direct, or *jet bleed*, service between nodes  $i$  and  $j$ . We use the term “bleed” since a route that does not pass through a hub can be said to bleed the cargo flow off from the main sorting facility. A hub being pushed to its limits quite possibly could benefit from a substantial diversion of cargo. It might also be the case that further construction at the hub is not possible, and that another hub or some form of bypass is required. Figure 1-8 illustrates a single-hub system overlaid with bleed routes. For our discussion, we shall assume that direct service that can be accomplished by truck has been determined, and that the matrix  $[\delta_{ij}]$  has been revised accordingly. It is possible to discern a jet bleed system using the  $\lambda - \gamma$  indicators, but the distinction is subtle. Suppose that the  $\gamma_{ij}$  are moderately-sized, but using previously discussed criteria the only property that we can deduce about the system at hand is that it is a single-hub system. Thus, the tests for feeder bases and regional hubs are negative. Then the following test will determine if a jet bleed is in force. Suppose that  $\lambda_1$  is the flow through the hub. If the hub processes all cargo we must have

$$\lambda_1 = \sum_{i,j \in I} \delta_{ij}.$$

If the sum above is greater than  $\lambda_1$ , then a jet bleed is the only possibility remaining for explaining the cargo diversion. So far, we have been modeling a system that is segmented with respect to time and with respect to route function. Thus, aircraft make deliveries only in the morning and pickups only at night. When considering bleed flights, we no longer insist that a route be purely delivery or purely pickup. Furthermore, we allow cyclic routes.

We may consider the bleed system to be a limiting case of a second type of multiple-hub system. The new design arises in a mathematical sense as the  $\gamma_{ij}$  grow large. We refer to it as the *trunk* multiple-hub system, so named for the trunk flights between hubs. It would be easy to include this design type with regional multiple-hubs under the one general classification of multiple-hub systems. However, there are some marked distinguishing characteristics between the two types, relative to the  $\lambda - \gamma$  indicators. First,



**Figure 1-8. Map of Bleed System Concept**

the regional hub concept grows out of the feeder system design as the  $\lambda_i$  increase beyond some threshold value. The internodal traffic rates  $\gamma_{ij}$  remain low throughout this evolution, though. In the trunk hub system, the  $\gamma_{ij}$  become large along with the  $\lambda_i$ .

A second distinguishing characteristic between the two system types is the route classification scheme. With feeder and regional hub systems, we can always classify a route as either pickup or delivery. For jet bleed and trunk hub systems, there are routes that are "in-between" pickup and delivery. In the former operation, these are the bleed routes, and in the latter system they are the trunk flights. This new type of route can be thought of as corresponding directly to the significant increases in the  $\gamma_{ij}$ .

A third differentiating aspect of trunk hub systems is that the service that a carrier offers can be inferred at least in part from observing the  $\gamma_{ij}$  and the relative locations of  $i$  and  $j$ . If nodes  $i$  and  $j$  are very distant from each other, then the amount of flow  $\gamma_{ij}$  is not likely to be morning delivery express cargo. This is because the time required to sort the cargo twice (at  $i$  and  $j$ ), plus the time required to fly from  $i$  to  $j$ , is likely to render the highest delivery service prohibitive. Thus, a flow of  $\gamma_{ij}$  between distant nodes is indicative of a different product offering by the carrier, possibly a late next-day service or a second-day service. In the next sections of this chapter we discuss system design from the perspective of facilitating these additional services.

Just as a large distance between  $i$  and  $j$  (for example, from New York to Denver) implies lower priority services, short distances for  $\gamma_{ij}$  (say New York to Chicago) are more likely indicative of a system designed for express cargo, since the time for two sorts plus transit might be small enough.

The final difference between the two hub system types is that of the probable fleet mix employed by each. The regional hub system behaves in a way that keeps the flow of cargo between any two points relatively small. Thus, smaller aircraft are best suited for such an operation. On the other hand, the trunk hub system acts to consolidate cargo and create large flows between certain pairs of nodes. In this case, large aircraft such as A300's, MD-11's, or even B747's are called for. Table 1-1 summarizes our discussion

interrelating the cargo flow parameters  $\lambda_i$  and  $\gamma_{ij}$ , the system type, and the fleet mix.

With this we conclude our discussion of U.S. domestic airline systems specialized for morning express cargo. In the interest of focusing our development, we omit formulations for the jet bleed and trunk-hub systems. Next, we consider how such systems can accommodate other classes of service.

## 1.6 Lower Priority Products

If a carrier offers a service that guarantees delivery a day or more later than the highest-priority service, the system must still satisfy the demand, but some flexibility in operations is allowed. For example, Federal Express offers a service that guarantees delivery a day after the highest priority (P1) service. Because of this, if there is not room for priority-two (P2) cargo on a flight, it can be left on the ground and carried on the following night.

To formulate the system with P2 cargo we introduce new variables for P2 flow and for P2 cargo that remains on the ground overnight.

### Decision Variables

$f_{ijm}^{pqk}$  = P1 flow with the usual interpretation

$\theta_{ijm}^{pqk}$  = P2 flow with the usual interpretation

$\Delta_{\ell}^{pq}$  = the amount of commodity  $pq$  that is P2, and that is left on the ground at  $p$  on day  $\ell$ .

### Constants

$\delta_{pq}^1$  = P1 demand

$\delta_{pq}^2$  = P2 demand

The problem as stated below will be understood for day  $\ell$ . We have a recursive formulation for  $\ell = 1, \dots, 7$ , where day 7 is Sunday. Because there would be no pickups or very little demand in the case of Sunday pickup operations, it is quite unlikely that any cargo would be left on the

$\lambda - \gamma$  Classification Scheme for Express Systems

	SINGLE HUB	SINGLE HUB WITH FEEDERS	REGIONAL HUB	JET BLEED	TRUNK HUB
$\lambda_k$	one large $\lambda_k$ , where $k$ is the hub	one large $\lambda_k$ , where $k =$ the hub	more than one $\lambda_k$ of significant size	$\lambda_k < \sum_{i,j} \delta_{ij}$	more than one large $\lambda_k$
$\gamma_{ij}$	small $\gamma_{ij}$ with no pattern	for several $m$ , $\gamma_{mj} > 0$ , for up to several $j$	for each $\lambda_k \gg 0$ , $\gamma_{kj} > 0$ for several $j$	possibly a few $\gamma_{ij}$ of noticeable size	at least a few large $\gamma_{ij}$ , where $\lambda_i$ and $\lambda_j$ are both large - i.e., hubs

**Table 1-1. Classification Scheme for Express Systems**

ground that day. Thus, we can set  $\Delta_0^{pq} \equiv 0$  for initial conditions. We now formulate the Single-Hub System Design Problem (SDP2) with P2 flow.

$$\text{minimize } z = \sum_{k=1}^2 \sum_{i \in I} \sum_{j \in I} \sum_{m \in M} c_{ijm} y_{ijm}^k + \sum_{i \in I_p} \sum_{m \in M} c_{im} z_{im}$$

subject to

$$\left( \sum_{m \in M} \sum_{j \in I} f_{ijm}^{pqk} - \sum_{j \in I} f_{jim}^{pqk} \right) = \begin{cases} \delta_{pq}^1 & i = p, & k = 1 \\ -\delta_{pq}^1 & i = q, & k = 2 \\ 0 & \text{otherwise} \end{cases} \quad (3-2a)$$

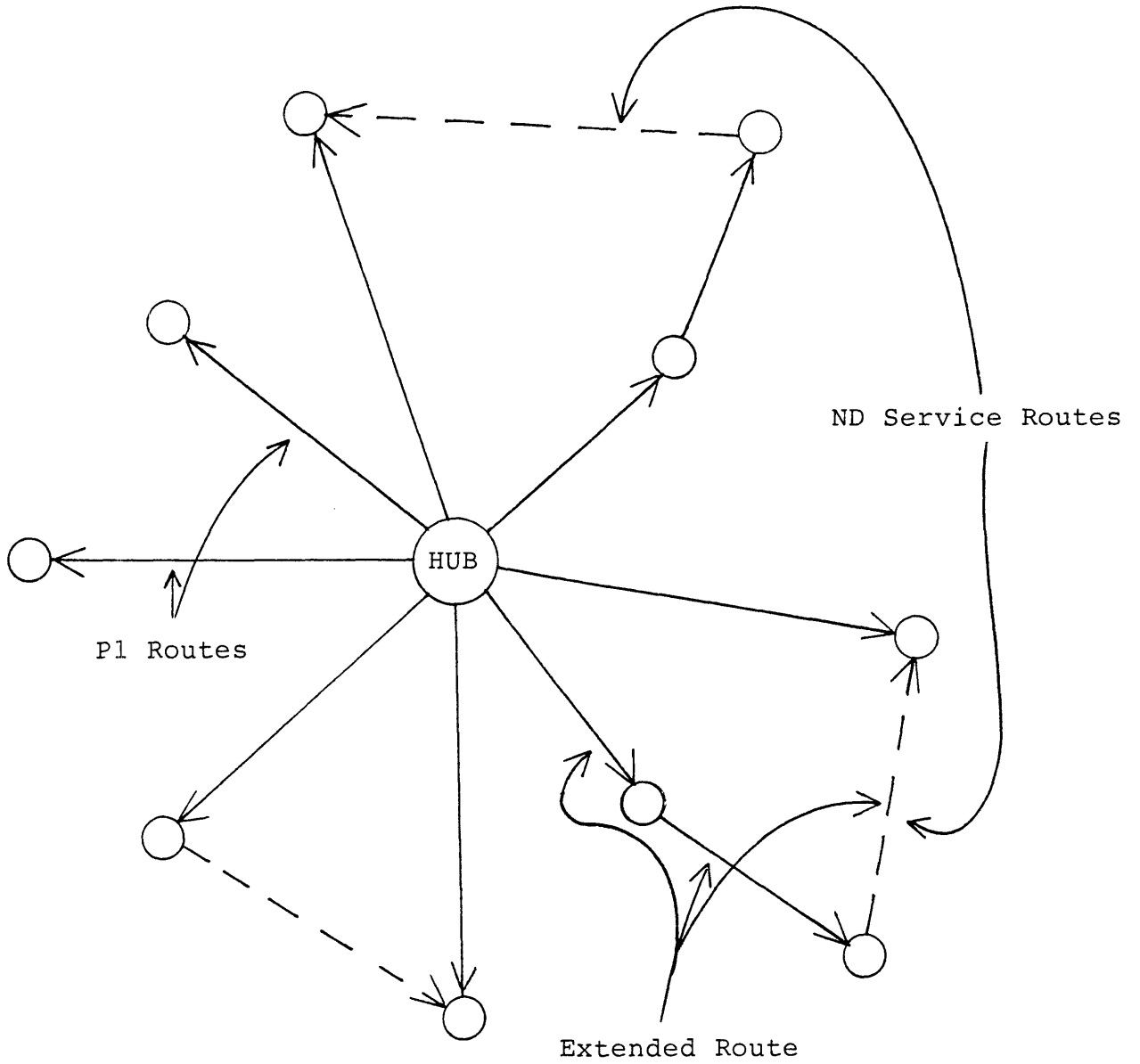
$$\sum_{m \in M} \left( \sum_{j \in I} \theta_{ijm}^{pqk} - \sum_{j \in I} \theta_{jim}^{pqk} \right) = \begin{cases} \delta_{pq}^2 - \Delta_{\ell}^{pq} + \Delta_{\ell-1}^{pq} & i = p, k = 1 \\ -\delta_{pq}^2 + \Delta_{\ell}^{pq} - \Delta_{\ell-1}^{pq} & i = q, k = 2 \\ 0 & \text{otherwise} \end{cases} \quad (3-2b)$$

$$\sum_{p \in I} \sum_{q \in I} \left( f_{ijm}^{pqk} + \theta_{ijm}^{pqk} \right) \leq K_m y_{ijm}^k \quad i, j \in I \cup \{0, n\}, m \in M \quad (3-3)$$

and (1-4a) through (1-9)

We have formulated the problem for second day delivery, but other possibilities exist. Even lower-priority products could be offered, for example, and the formulation of such a system would be an extension of constraints (3-2b). A product with more complex implications is a "next-day" (ND) service that guarantees delivery by the end of the next business day (5:00 p.m.), instead of a 12:00 noon or 10:30 a.m. delivery commitment for the highest-priority service.

There are a number of operational options for ND service. First, we could include ND cargo along with P1 as part of the same system with possible ND route extensions that meet later allowable cutoff times. Figure 1-9 illustrates such a possibility. If ND demand became high enough, it might be appropriate to evaluate special ND flights. These aircraft could both depart later from points along pickup routes and arrive later at points along delivery routes than their counterpart aircraft with P1 cargo. However, a more attractive solution for heavy ND demand, and also for heavy P2 demand, could be a double-turn system.



**Figure 1-9. P1 Route Extension for ND Service**

## 1.7 Double-Turn Systems

The systems that we have discussed thus far all have the property that hub-based flights come into and depart from the hub exactly once in a twenty-four hour period. As we mentioned earlier the operation of landing at a hub, exchanging cargo, and taking off, is referred to as a turn. Thus, the systems we have been analyzing are *single-turn systems*. The discussion of the previous section indicates that it might be economically desirable to design a system that entails more than one turn.

The feasibility of such a system arises from the fact that many aircraft spend at least 12 continuous hours on the ground at a field station in a single-turn system. While aircraft utilization for a high-priority carrier must be thought from a different perspective than for a passenger carrier (i.e., revenues do not necessarily increase with utilization), we nonetheless should note that this amount of time can offer the planner some extra freedom in system design. As an example, consider the P2 service of the last section, which guarantees the delivery of parcels one day later than P1 service. The limitations of a single-turn system under heavy P2 demand are easily seen. If we cannot move P2 cargo even after leaving some of it on the ground and waiting until the next day, then we must purchase additional aircraft. However, we can use a double-turn system and avoid this capital expenditure. The operation takes place during the idle period of a single-turn system. We choose a set of nodes, fly the previous night's excess P2 demand in this set to the hub, perform a sort, and fly back out to the field. It is not necessary during this turn to exhaust all the excess P2 on the ground at all points. It is only necessary to process enough of it so that the following night's P1 flights can handle the remainder.

We now formulate the double-turn system for P2. As a preface, we note that the P2 double-turn system formulation can be modified quite easily to handle the next-day (ND) service we discussed in the previous section. We use the same variables for P2 flow that we defined in the last section, with one exception. We define

$\Delta^{pq}$  = amount of P2 destined for  $q$  that is left on the ground at  $p$  in the



first turn.

We use  $k = 3$  for second-turn pickup routes and  $k = 4$  for second-turn delivery routes. We omit the time constraints, but they are essentially the same as for (SDP).

Although we need not insist that second-turn flights exhaust all excess P2, we present a formulation that insists that they do, for simplicity. The extension to a recursive formulation analogous to (SDP2) is straightforward but cumbersome. We now state the formulation of the Single-Hub Double-Turn System Design Problem for P2 (SDDP2).

$$\text{minimize } z = \sum_{k=1}^4 \sum_{i \in I} \sum_{j \in I} \sum_{m \in M} c_{ijm} y_{ijm}^k + \sum_{i \in I_F} \sum_{m \in M} c_{im} z_{im} \quad (4-1)$$

subject to

$$\sum_{m \in M} \sum_{j=0}^n f_{ijm}^{pqk} - \sum_{m \in M} \sum_{j=0}^n f_{jim}^{pqk} = \begin{cases} \delta_{pq}^1 & i = q, k = 1 \\ -\delta_{pq}^1 & i = q, k = 2 \\ 0 & \text{otherwise} \end{cases} \quad p, q \in I, i \in I_F \quad (4-2a)$$

$$\sum_{m \in M} \sum_{j=0}^n \theta_{ijm}^{pqk} - \sum_{m \in I} \sum_{j=0}^n \theta_{jim}^{pqk} = \begin{cases} \delta_{pq}^2 - \Delta^{pq}, & i = p, k = 1 \\ -\delta_{pq}^2 + \Delta^{pq}, & i = q, k = 2 \\ 0 & \text{otherwise} \end{cases} \quad p, q \in I, i \in I_F \quad (4-2b)$$

$$\sum_{m \in M} \sum_{j=0}^n \theta_{ijm}^{pqk} - \sum_{m \in M} \sum_{j=0}^n \theta_{jim}^{pqk} = \begin{cases} \Delta^{pq} & i = p, k = 3 \\ -\Delta^{pq} & i = q, k = 4 \\ 0 & \text{otherwise} \end{cases} \quad p, q \in I, i \in I_F \quad (4-2c)$$

$$\sum_{j=0}^n f_{ijm}^{pqk} - \sum_{j=0}^n f_{jim}^{pqk} = \begin{cases} 0 & p \neq i, k = 1 \\ 0 & q \neq i, k = 2 \end{cases} \quad i, p, q \in I, m \in M \quad (4-2d)$$

$$\sum_{j=0}^n \theta_{ijm}^{pqk} - \sum_{j=0}^n \theta_{jim}^{pqk} = \begin{cases} 0, & p \neq i, k = 1, 3 \\ 0, & q \neq i, k = 2, 4 \end{cases} \quad i, p, q \in I, m \in M \quad (4-2e)$$

$$\sum_{p \in I} \sum_{q \in I} (f_{jim}^{pqk} + \theta_{ijm}^{pqk}) \leq K_m y_{ijm}^k \quad i, j \in I \cup \{0, n\}, k = 1, 2, m \in M \quad (4-3a)$$

$$\sum_{p \in I} \sum_{q \in I} \theta_{ijm}^{pqk} \leq K_m y_{ijm}^k \quad i, j \in I \cup \{0, n\}, k = 3, 4, m \in M \quad (4-3b)$$

$$y_{ijm}^k \leq z_{jm} \quad i \in I \cup \{0, n\}, j \in I_F, k = 1, \dots, 4, m \in M \quad (4-3c)$$

$$\sum_{j=0}^n y_{ijm}^k - \sum_{j=0}^n y_{jim}^k = 0 \quad i \in I, k = 1, \dots, 4, m \in M \quad (4-4)$$

$$\sum_{m \in M} y_{inm}^{k+1} - \sum_{m \in M} y_{nim}^k = 0 \quad i \in I_F, k = 1 \text{ or } 3 \quad (4-5)$$

$$\sum_{m \in M} y_{inm}^2 - \sum_{m \in M} y_{nim}^3 \geq 0 \quad i \in I_F \quad (4-5b)$$

$$\sum_{m: T(m)=\alpha} \sum_{i \in I_F} y_{nim}^1 \leq N_\alpha \quad \forall \text{ distinct types } \alpha \text{ plus time constraints} \quad (4-6)$$

$$a_{im}^k, f_{ijm}^{pqk}, \theta_{ijm}^{pqk}, \Delta^{pq} \geq 0, \quad y_{ijm}^k, z_{im} = 0 \text{ or } 1 \quad (4-7)$$

$$i, j \in I \cup \{0, n\}, p, q \in I, m \in M, k = 1, \dots, 4$$

Constraints (4-2b) determine the excess P2 cargo from the first turn and (4-2c) require that the second turn deliver the excess. The flow in turn 1 is capacitated by (4-3a) and in turn 2 by constraints (4-3b). The only other constraint set that differs significantly from the (SDP2) formulation is (4-5b). They effectively force the endpoints of the second-turn routes to be a subset of the endpoints of the first-turn routes.

By treating  $\theta_{ijm}^{pqk}$  as a flow variable for the ND problem, we can model such a system with the same formulation. There will, however, be a much more tightly constrained second turn, since arrivals on delivery segments will have to occur in time to meet the “next day” delivery commitment. It is likely that a double-turn system for ND service would be better applied to a multiple-hub network, due to the need for a short turn span.

We have noted that the topic of hub facility design is important for multiple-hub systems, where some nodes have not yet been established as hubs. The double-turn system potentially has a significant impact on necessary hub capacity for any system, as can be seen by examining the hub capacity constraints. The principle constraints dealing with capacity for (MDDP2) are

$$\sum_{j=0}^n \sum_{m \in M} P_m y_{ijm}^k \leq A_i + \sum_{r=1}^{N_i^R} A_{ir} z_{ir}^R \quad k = 1 \text{ or } 3, i \in I_H \quad (4 - 9a)$$

$$\sum_{p \in I} \sum_{q \in I} \sum_{j=0}^n \sum_{m \in M} (f_{ijm}^{pq2} + \theta_{ijm}^{pq2}) \leq B_i + \sum_{\beta=1}^{N_i^S} B_{i\beta} z_{i\beta}^S \quad i \in I_H \quad (4 - 9b)$$

$$\sum_{p \in I} \sum_{q \in I} \sum_{j=0}^n \sum_{m \in M} \theta_{ijm}^{pq3} \leq B_i + \sum_{\beta=1}^{N_i^S} B_{i\beta} z_{i\beta}^S \quad i \in I_H \quad (4 - 9c)$$

We recall that a double-turn system implies fewer aircraft and fewer parcels per turn than for a single-turn system that gives the same service. Thus, there is a concomitant potential for less needed ramp space and less needed sorting capacity, respectively. This potential is illustrated by the above constraints. However, in a double-turn system that is completely dominated by the first turn, the savings may not be realized.

With this we conclude Chapter 1. We have identified and discussed five elemental system types for express cargo. They are

1. Single-Hub
2. Feeder
3. Regional Multiple-Hub
4. Jet Bleed
5. Trunk Multiple-Hub

Using the indicator values  $\lambda_i$  and  $\gamma_{ij}$  we have seen that the pure form of each of the above systems can be characterized, as well as the probable fleet makeup deduced. We have also discussed other product offerings by overnight carriers within the framework of the express system. In addition, we formulated several of these systems quantitatively.

The remainder of this thesis focuses on the single-hub single-turn problem. We discuss related problems and present a literature review in Chapter 2, and we investigate additional formulations in Chapter 3, searching for

one that seems likely to yield to efficient solution methods. Having decided on a formulation, we develop a solution approach in Chapter 4. Chapter 5 reports on the computational results of our approach and suggests future avenues for research in this extremely rich area of applied operations research.

## Chapter 2

# RELATED PROBLEMS AND OPTIMIZATION-BASED SOLUTION METHODS

### 2.1 Transportation

Conceptually and physically, the simple single-hub system design problem (SDP) resembles a sort of two-staged Multiple-Vehicle Routing Problem (MVP). First, on the “delivery side”, all points must be visited by at least one aircraft from the hub, and the correct amount of cargo must be delivered to each point. Once this is accomplished, the first stage is finished. The second stage occurs on the “pickup side” when all points are visited again by at least one hub-bound aircraft. Paralleling the delivery operations, the correct amount of cargo must be picked up at each point. The problem is not two independent routing problems because delivery route endpoints must match pickup route origin points. Also, we sometimes may require more than one vehicle to visit a node, whereas the classical MVP requires that each node be visited by exactly one vehicle. Moreover, we are solving more than merely a capacitated vehicle routing problem with time constraints because of the complication of locating auxiliary equip-

ment at appropriate stops. There is thus an aspect of facility location even in (SDP).

The literature on multiple-vehicle routing problems is immense in scope and continues to grow. Gavish and Graves [G2] have formulated many variants of the Traveling Salesman Problem (TSP) and relate some new formulations to past ones. Magnanti [M1] gives a thorough overview of vehicle routing and scheduling problems in various settings. He presents different formulation approaches for MVP and discusses a number of solution approaches, focusing especially on Lagrangian Relaxation and Benders Decomposition. Magnanti and Wong [M5] have shown that many classical combinatorial problems, including TSP, the Vehicle Routing Problem, and the Facility Location Problem, are variants of the Fixed Charge Network Design Problem. Bodin, et. al. [B4] have an comprehensive survey of the state of the art in vehicle routing and crew scheduling. See also Golden and Assad [G5], for a discussion of new developments.

Fisher, et al. [F2] address a single-hub truck routing problem for a liquefied gas company. The problem is very much like (SDP) except that the only operations made are deliveries. All of the liquefied gas is supplied by the hub. However, multiple deliveries are allowed at many nodes, and there are different vehicle capacities to contend with as well as time constraints. Yet another variant of the vehicle routing problem is one that includes *backhauling*, which allows the dual functions of pickup and delivery in [G6].)

The design problem for the feeder system (SDPF) obviously contains a facility location problem as one aspect. Tansel, Francis, and Lowe [T1] provide an excellent survey of facility location problems on networks. Our problem, however, is concerned more with locating facilities simultaneously with the determination of service arcs. We may recast a facility location problem as part of an overall network design problem by representing the potential facility node  $i$  as a directed arc  $(i', i'')$ . All arcs in the original network of the form  $(a, i)$  become arcs of the form  $(a, i')$ , and arcs of the form  $(i, b)$  become arcs of the form  $(i'', b)$ . Of course, our problem allows multiple service arcs, so we have arcs of the form  $(i, j, m)$ , where  $i$  and  $j$  are nodes and  $m$  is an aircraft. Nonetheless, this may represent a viable

avenue of formulation to explore, and others have done so successfully (see Wong [W1], for example).

O'Kelly [O1] has written one of the few papers on locating interacting hub facilities, which addresses our Trunk Hub Problem. The Trunk Hub Problem is also similar in some respects to a problem that was studied by Singhal [S5]. His paper deals with a truck routing and load planning problem for a major parcel delivery company, in which parcels often pass through more than one sort (hub) before proceeding to their final destinations. Briefly, the load planning problem is to determine how much of a given truck-load each hub must sort, and from where that load is to come. The problem also contains elements of the alternative-priority products which we have discussed. There are significant differences, however. One point of departure is the fact that a truck travels only one leg from a field node to a hub. Our model allows an aircraft to make multiple stops at a field node before (in the case of pickup routes) it reaches a hub. Also, it is possible that only lower-priority products could pass through more than one hub. The (RDP) formulation is aimed primarily at high-priority products which might not have the time to pass through more than one sort. A case in point is a past Federal Express study of the possibility of establishing multiple hubs instead on the single hub at Memphis. The plan was to establish a system in which packages would often go through two sorts. This plan was abandoned as being too complex and expensive at the time [H2]. Thus, our considerations will focus on systems that have only one sort for any parcel. Nonetheless, we wish to emphasize that the load planning problem is quite important for trucking (see Powell and Sheffi [P1]) and rail freight (see Assad [A2]).

Within the airline industry itself, it appears that very little has been done that resembles the situation for express carriers. Simpson [S4] is one of the earliest to exploit the mathematical concept of a network and relate it to passenger airline system design. The problem in the case is significantly different from ours, however. Possibly the closest situation to the high-priority case is that of another cargo carrier, Flying Tiger. Marsten and Muller [M7] model the network design and fleet planning problems for

Flying Tiger in a succession from the single-hub case to the multiple-hub case, and finally to a double-turn case. Moreover, as is stated in their problem definition, for a single-turn system, aircraft leave outlying nodes in the evening with cargo that is to be delivered in the morning. Their double-turn system has a second turn during daylight hours, just as for the express carriers. There are, however, some significant differences.

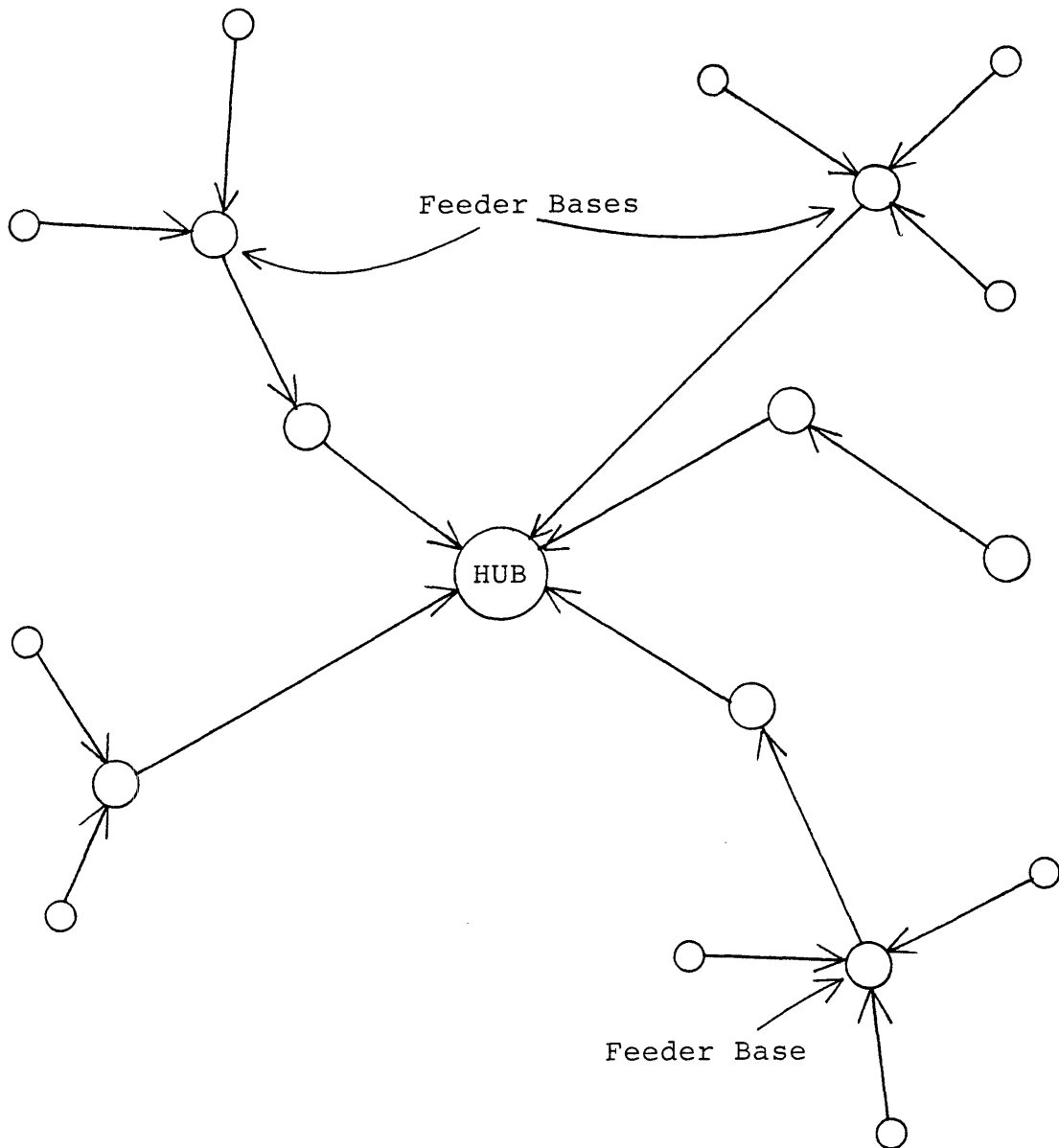
First, the design of the network, in terms of permissible service arcs, is determined a priori by management to the extent that any node has a unique path to any hub (if it has a path to a hub). This simplifies the problem to a very great degree. The second major difference is that not all of the demand must be carried. The objective function of the problem formulation is to maximize profit. Conceivably, this could mean not serving a demand point at all. This demonstrates a major philosophical difference between the two types of business that is reflected in the design problem. As we have noted, the overnight carrier must handle all the demand because fast, *highly reliable* service is its product. Customers cannot be served only some of the time.

Interestingly, this facet of Flying Tiger's operation provides a type of logical link between scheduled passenger airlines and express cargo carriers. That is, scheduled passenger carriers generally allow for some business lost due to insufficient capacity. (One rare exception to this is the Eastern Airlines Shuttle, where every effort is made never to turn passengers away. Here, as with the express systems, the product is reliability.)

## 2.2 Communications

Figure 2-1 shows a possible pickup route structure for the Single-Hub System Design Problem with Feeders (SDPF). Alternatively, this might be the structure of a centralized data communications network. In such a diagram the arcs of the graph could represent data links, each with some capacity. A computer could be at the center of the network and terminals could be located at the nodes. Such networks are described in Schwartz [S2], for example. The principle difference between a communications network





**Figure 2-1. Pickup Routes of a Single-Hub Feeder System**

of this type and a system design for (SDP) or (SDPF) is the travel time between points. We elaborate on this point to illustrate the relationship between the two problems.

A reasonable approach to attacking (SDP) is to solve the pickup and delivery sides separately. Then, if constraints (1-5) are satisfied, we have a feasible solution to SDP. In another vein, the method of Hinson and Mulherkar [H3] was to reduce the whole problem to a "symmetric" system that was constrained so that any "one-sided" solution automatically solved the pickup and delivery sides simultaneously. In either case, the solution process (i.e., a solution of either pickup side or the delivery side or a solution of the symmetric problem) would involve solving only a "single-sided" problem. Now let  $t_{ijm} \equiv 0$  for all  $i, j$ , and  $m$ . An immediate consequence is that time constraints are unnecessary, so (1-7a) through (1-7e) may be discarded. Because we are now solving a single-sided system, the cost of ground equipment for a particular aircraft at a station can be absorbed into the cost of a service arc, and we can then discard constraints (1-3b). Moreover, if we are doing fleet planning with no limit on aircraft availability, (1-6) may be dropped. This leaves the following system:

$$\text{minimize } z = \sum_{i,j,m} c_{ijm} y_{ijm}^1$$

subject to

$$\sum_{m \in M} \sum_{j=0}^n f_{ijm}^1 - \sum_{m \in M} \sum_{j=0}^n f_{jim}^1 = d_i^1 \quad i \in I \quad (1-2a)$$

$$f_{ijm}^1 \leq K_m y_{ijm}^1 \quad i, j \in I \cup \{0, n\}, m \in M \quad (1-3a)$$

$$\sum_{j=0}^n y_{ijm}^1 - \sum_{j=0}^n y_{jim}^1 = 0 \quad i \in I, m \in M \quad (1-4)$$

$$f_{ijm}^1 \geq 0, y_{ijm}^1 = 0 \text{ or } 1, \quad i, j \in I \cup \{0, n\}, m \in M \quad (1-8)$$

This formulation is very similar to Gavish's formulation [G1], of a special case of the one-terminal Telpak problem (see Rothfarb and Goldstein [R3]), where line capacities can be purchased in bauds. The major difference is

that instead of (1-4), Gavish has

$$\sum_{j=0}^n \sum_{m \in M} y_{ijm}^1 = 1 \quad i \in I.$$

We will not develop this line of thought further, but we note that the wide variety of express systems could allow for a number of similar comparisons. (See for example, Mirzaian [M9].)

## 2.3 Solution Methodologies

The design problems that we have formulated are of a very complex nature, and to our knowledge, no formulations have been given for any of them. However, the related classical problems in transportation that we have noted are all NP-complete. Lenstra and Rinnooy Kan [L2] offer a concise overview of NP-complete problems in vehicle routing and scheduling. Among these are the Traveling Salesman Problem and Multiple Vehicle Routing Problem. Many related problems in the area of network design are also NP-complete. For example, the Budget Network Design Problem is NP-complete. (This problem puts a limit on the total number of arcs that can be included in any feasible solution; see Johnson, Lenstra, and Rinnooy Kan [J1]).

The apparent intractability of most variants of TSP and network design problems only naturally leads one to consider the use of heuristics. Indeed, such approaches have been very popular in the past and will doubtless continue to be. However, in recent years, researchers have made much progress in the development of optimization-based methods. Moreover, heuristics in and of themselves have limitations. Wong [W1] has shown that the problem of finding a heuristic for the Budget Network Design Problem (BNDP) that always comes within a factor of  $N^{(1-\epsilon)}$  of the optimal solution, where  $N$  is the number of nodes and  $0 < \epsilon < 1$ , is NP-complete. Thus, it is quite significant that getting "reasonably good" BNDP solutions is no easier than getting optimal solutions, from the standpoint of computational complexity theory. Extensions of this result to other network design variants should come as no surprise. Furthermore, as Magnanti [M1] points out, heuristics

have other disadvantages, including the difficulty of sensitivity analysis and the possible compounding of the inaccuracies of input data and heuristic solutions.

Another reason for investigating optimization-based procedures for our problems is that many of these approaches, such as Lagrangian Relaxation, generate solution bounds as part of the process. Thus, we can often stop with an intermediate solution with a guarantee as to its closeness to optimality. We therefore shall discuss some of the recently exploited algorithms for obtaining optimal results to problems that are related to ours.

## 2.4 Lagrangian Relaxation

One of the most popular techniques for solving large-scale linear mixed-integer programming problems is price-directive decomposition, or Lagrangian relaxation. One of the earliest successful uses of the technique was on the Traveling Salesman Problem, by Held and Karp [H1], in 1971. Use of the technique has since grown to include a number of applications. Cornuejols, Fisher and Nemhauser [C3] have used the technique on an uncapacitated facility location problem. Singhal [S5] used Lagrangian relaxation in addressing the load planning problem that we described earlier. Gavish [G1] has used it quite successfully on a number of capacitated spanning tree problems, and Srikanth and Gavish [G3] used the technique on the Multiple Traveling Salesman Problem. This list is far from complete, and the number of applications can be expected to grow substantially.

Fisher [F1] has a survey article on the methodology, explaining the theory behind it, its uses, and prospects for future development and applications. Additional treatments can be found in Shapiro [S3] and Magnanti [M1]. For an illustration of the method, see Appendix A.

There is, in general, a tradeoff between competing relaxations involving the ease of solving the relaxation and the size of the duality gap. Since we generally use Lagrangian relaxation for a branch-and-bound routine, reducing the duality gap can lead to more efficient performance in this stage of our total solution technique. The result might be better overall performance.

Also, as Fisher et al. [F2] have found, a Lagrangian solution that is close to optimal can be modified in many instances to be a feasible solution to the original problem with a near-optimal objective value. Thus, we can possibly terminate our procedure at such a point with an acceptable solution. It is clear from these considerations that careful planning is appropriate in choosing a good relaxation, although we might only be able to find the best relaxation empirically.

There are, broadly speaking, three methods for determining an optimal  $u$  for the Lagrangian dual. These are subgradient optimization, column generation, and multiplier adjustment techniques. Subgradient optimization has been used in many applications of Lagrangian relaxation to both classical and practical problems. Among the applications mentioned earlier in this section, Singhal [S5], Srikanth and Gavish [G3], and Gavish [G1] have reported excellent results using subgradient optimization. Gavish and Graves [G2] used the technique on the Lagrangian relaxation that produces the 1-arborescence that we describe in Appendix A.

Convergence can be slow, however, as Fisher et al. [F2] discovered in using the subgradient procedure on the truck routing problem for the liquefied gas company, which we have outlined. Their preliminary results showed that subgradient optimization resulted in good solutions, but that too much computer time was used. They report, however, that their efforts produced insights into the nature of good solutions that allowed them to short-cut the subgradient procedure through the development of a multiplier adjustment technique.

Multiplier adjustment methods are tailored to exploit the richness of a particular application. A sequence of multipliers is generated by the rule

$$u^{i+1} = u^i + t_i d_i,$$

where  $t_i$  is a positive scalar, and  $d_i$  is a direction, chosen from a small set of directions along which the directional derivative of  $Z^*(u)$  is simple to determine (see Fisher [F1]). Erlenkotter [E1] has used a multiplier adjustment procedure to achieve quite dramatic results in conjunction with a dual ascent procedure on the uncapacitated facility location problem.

Shapiro [S3] gives a treatment of a generalization of the primal-dual simplex method called the primal-dual ascent algorithm, which is used to solve the Lagrangian dual. This is an example of a column generation approach to finding an optimal Lagrange multiplier. Marsten [M6] also reports successful applications of a method called *BOXSTEP*.

This completes our discussion of Lagrangian relaxation. We now discuss now another method of addressing large-scale optimization, that of resource-directive decomposition.

## 2.5 Benders Decomposition

Benders Decomposition is a method of dividing up a formulation by removing variables that “complicate” the problem, instead of complicating constraints as in Lagrangian relaxation. Developments of the principle can be found in the texts of Lasdon [L1] and Shapiro [S3]. To date, the technique has not shown the promise of Lagrangian relaxation when applied to vehicle routing problems (see Magnanti [M1]). However, success has been reported in applying the method to other variants of network design problems. Geoffrion and Graves [G4] have applied the technique successfully to problems of multicommodity distribution system design. In the area of network design, Magnanti, Mireault, and Wong [M2] have used the technique to solve some of the largest problems on record. Also, Richardson [R2] solved some aircraft routing problems for QANTAS Airlines with Benders Decomposition. We must remark that the aircraft routing problem solved by Richardson is not geometrically similar to ours. Nonetheless, the problem is one of a vehicle routing flavor, and it makes sense to consider the method’s applicability to our problems. For an illustration of Benders Decomposition, see Appendix B.

For Benders Decomposition to be applied effectively, a straightforward approach may not suffice. Gavish [G1] found that such a tactic led to relatively-poor performance. However, it appears that some problem formulations do not lend themselves in *any* original way to solution using Benders’ method. Gavish and Graves [G2] demonstrate that Benders De-

composition applied to their formulation of TSP offers no theoretical computational advantage over the standard formulation that Dantzig, Fulkerson, and Johnson [D1] produced. Gavish and Graves demonstrate that Benders Decomposition, applied to the formulation of TSP that we gave earlier, results in Benders cuts that are merely the subtour-breaking constraints given by Dantzig et al. [D1]. Magnanti [M1] shows a similar result. However, these are also straightforward applications, and it could be that using techniques such as the generation of pareto-optimal cuts would produce more meaningful results. One new development that brings together Lagrangian relaxation and Benders Decomposition in a unified manner is cross decomposition. (See Van Roy [V1] and Van Roy [V2].) In any event, it is clear that the best approach is a synthesis of methods such as branch-and-bound, Benders Decomposition, and pareto-optimal cut generation, together with heuristics that take advantage of the special structure of the problem (see Magnanti et al. [M2]). One such heuristic uses linear programming duality theory as its basis. We discuss this technique next.

## 2.6 Dual Ascent

Some other recent developments in addressing problems related to ours have produced promising results. One of the most exciting is that of *dual ascent*. Methods of dual ascent are heuristics that take advantage of the special structure of the dual formulation of the LP relaxation of a primal problem. By designing a heuristic judiciously, very-quickly-obtained, tight lower bounds to an integer or mixed-integer primal can be used quite effectively in a branch-and-bound procedure.

The principle of dual ascent relies on the fact that for a minimization in an integer programming problem  $F$ , the LP relaxation of  $F$ ,  $LPF$ , is such that  $v(F) \geq v(LP F)$ , where  $v(*)$  is the optimal value of problem  $*$ . By LP-duality theory,  $v(LP F) = v(DLP F)$ , where  $DLP F$  is the LP dual of  $LP F$ . Moreover, if  $W_D$  is the value for any feasible solution to  $DLP F$ , then  $W_D \leq v(DLP F)$ , since  $DLP F$  is a maximization problem. Thus, a quickly-obtained value for  $W_D$  could be useful as a lower bound in a branch-

and-bound procedure for  $F$ .

From the foregoing discussion, it is evident that the ultimate purpose of using a dual ascent heuristic is the same as for using Lagrangian relaxation, that is, for branch-and-bound routines. In fact, it often happens that, as with Lagrangian relaxation, a dual ascent solution is not only close to optimal, but also is nearly feasible (and can be made so with manipulation) or actually feasible for the original problem. Erlenkotter [E1] has achieved excellent results on the uncapacitated plant location problem with a dual ascent algorithm that interacts with a dual adjustment algorithm. Wong [W2] has achieved similar results in both speed and solution quality with a dual ascent algorithm for the Steiner Tree Problem on directed graphs. Magnanti et al. [M2] have incorporated a dual ascent algorithm in their Benders Decomposition approach to network design. In many cases, their dual ascent routine was able to find an optimal solution, rendering the Benders Decomposition unnecessary. Other successful uses of dual ascent include database location in computer networks, by Fisher and Hochbaum [F3], and dynamic plant location, by Van Roy and Erlenkotter [V3].

As we have noted, a dual ascent solution is often nearly feasible and can induce a feasible primal solution of excellent quality. In fact, Wong's [W2] approach includes the generation of a primal feasible solution. Magnanti et al. [M2] also use such a technique. Importantly, if  $W_F$  is the value of a primal feasible solution for  $F$ , then  $W_F \geq v(F) \geq v(LPF) = v(DLPF) \geq W_D$ . Thus, both upper and lower bounds can often be generated as integral parts of dual ascent heuristic, increasing the effectiveness of the approach for branch-and-bound. Moreover, if  $W_F = W_D$ , then we have proved optimality of the generated solution. This was quite often the case for many of the previously mentioned results (see, for example, Wong [W2] and Erlenkotter [E1]). Indeed, as Fisher [F1] points out, it would be particularly beneficial to understand which properties of combinatorial problems give rise to good approximations in the LP relaxation.

Fisher [F1] also points out that the development of dual methods for solving such LP's by taking advantage of their special structure would be singularly constructive. These exact methods of solution are represented by



such methods as those developed in Schrage [S1] and Miliotis [M8]. In concluding this section, we must also mention that the recent discovery of a new polynomial-time algorithm for solving linear programs by Karmarkar [K1] could have a very positive influence on the dual ascent approach. This could be especially true for those problems where a dual ascent heuristic will not, in general, yield tight bounds.

## 2.7 Cutting Planes

Recent years have seen the method of cutting planes applied to TSP and facility location problems. Cutting planes that define facets for the polytope of the formulation have been used in conjunction with branch-and-bound by Grötschel and Padberg [G8] and Crowder and Padberg [C3] to solve very large symmetric TSP's. Guignard [G9] has applied a facet-generating cutting-planes approach to a simple facility location problem, showing how to identify some of the most important facets. Crowder and Padberg [C3] use their technique in conjunction with a branch-and-bound procedure, providing another example of an approach where a synthesis of methods yields a superior overall methodology.

## 2.8 Summary

In this chapter we have reviewed the literature with respect to problems that are similar to ours and with respect to optimization-based solution approaches that have been effective in solving related problems. We have not surveyed any of the well-known heuristic methods for obtaining good feasible solutions to vehicle routing problems. Such techniques should not be eliminated as viable approaches, however, and we shall consider these in our later development where appropriate. Methods that are representative of these heuristics are Clarke-Wright savings approaches, k-opt algorithms, and sweep algorithms. Descriptions of these techniques can be found in Golden and Magnanti [G7].

An important point that was mentioned in this chapter is that a synthesis of methodologies is proving to be a very powerful way of attacking

previously unassailable problems. Magnanti et al. [M2] make this point in their report on applying Benders Decomposition to network design. Indeed, their synthesis of approaches enabled them to solve some very large problems. Fisher et al. [F2] have used a similar tactic in addressing their problem in multiple vehicle routing. A central premise in this is that many problems, especially real-life ones, simply are not amenable to solution by generalized techniques. In order to profitably attempt a solution to these problems, the richness of the application at hand must be properly reflected in the design of the algorithms and overall procedures. This will be our approach in subsequent chapters, because the problems that we have formulated are not only rich in structure, but are not, to our knowledge, quite like any problem thus far appearing in the popular literature.

## Chapter 3

# THE SINGLE-HUB, SINGLE-TURN PROBLEM

Virtually any system for an express carrier will have one or more hubs that provide the sorting function and act as bases for aircraft that serve a specified region. In fact, no matter what the overall network structure is, simple single-hub designs will comprise natural elemental subsystems. For this reason, a proper assessment of the total system must include an evaluation of each single-hub component.

In this chapter, we propose and analyze different formulations for the Single-Hub System Design Problem, SHP. Although formulation (SDP) is very finely-grained and exhibits much of the problem's structure, because of its complexity we need another, more tractable, formulation. We investigate several models, not only as part of a search for a solution method, but also in an effort to expose the richness of the problem.

### 3.1 Additional Formulations and Benders Decomposition

Our first approach to remodeling SHP entails absorbing all time constraints and flow conservation constraints into a *route bundle* decision variable. A route bundle describes a route, a type of aircraft flying that route, and the quantity of that aircraft type on the route. We also employ a flow variable in our new formulation, thereby creating a path-flow model, which

we designate (PF).

The variable definitions are:

$m$  = a subscript that designates both the type and quantity of aircraft being employed:

$T_m$  = type,  $N_m$  = quantity

Example: Let type 1 = B727-200, and

type 2 = MD-80

Then a possible assignment of values for  $m$  is

$m$	$T_m$	$N_m$
1	1	1
2	1	2
3	2	1
4	2	2

If  $m = 3$  then the aircraft type is the MD-80, and the quantity is 1.

$k$  = a superscript that denotes the period, or function, of a route bundle, a flow variable, etc. If  $k = 1$ , the period (i.e., function) is "pickup". If  $k = 2$ , the period is "delivery".

$R_i^k$  = the set of all routes in period  $k$  that include node  $i$ .

$S_r$  = the set of stops on route  $r$ .

$F_i$  = the set of pickup routes whose starting point is node  $i$ .

$L_i$  = the set of delivery routes whose last stop is at node  $i$ .

$K_m$  = the total capacity of  $N_m$  aircraft of type  $T_m$ .

$c_{rm}$  = the cost of using route bundle  $\beta_{rm}^k$ .

$c_{ija}$  = the per-unit operating cost of ferrying aircraft type  $a$  from node  $i$  to node  $j$ .

$d_i^k$  = the demand at node  $i$  during period  $k$ .

$I$  = the set of node indices, excluding the hub.

$I^\circ =$  the set of node indices including the hub.

$A =$  the set of aircraft type indices.

$Q_a =$  the quantity of aircraft type  $a$  available.

The decision variables are:

$$\phi_{rm}^k = \begin{cases} 1 & \text{if route } r \text{ is flown by } N_m \text{ aircraft of type } T_m \text{ during period } k \\ 0 & \text{otherwise} \end{cases}$$

We let  $\beta_{rm}^k$  denote the actual route bundle for which  $\phi_{rm}^k$  is a decision variable.

$x_{ija} =$  an integer variable that denotes the number of placement flights of aircraft type  $a$  from node  $i$  to node  $j$ .

$\gamma_{irm}^k =$  amount of cargo picked up ( $k = 1$ ) or delivered ( $k = 2$ ) by route bundle  $\beta_{rm}^k$  at node  $i$ . Node  $i$  must be a stop on route  $r$ .

Using these definitions, the new formulation, (PF), is

$$\text{minimize } V = \sum_k \sum_m \sum_r c_{rm} \phi_{rm}^k + \sum_a \sum_i \sum_j c_{ija} x_{ija} \quad (\text{PF-1})$$

subject to

$$\sum_m \sum_{r \in R_i} \gamma_{irm}^k = d_i^k \quad i \in I, k = 1, 2 \quad (\text{PF-2})$$

$$K_m \phi_{rm}^k \geq \sum_{i \in S_r} \gamma_{irm}^k \quad \text{for appropriate } r - m, k = 1, 2 \quad (\text{PF-3})$$

$$\sum_{m: T_m = a} \sum_r N_m \phi_{rm}^k \leq Q_a \quad a \in A, k = 1, 2 \quad (\text{PF-4})$$

$$\sum_{m: T_m = a} \sum_{r \in L_i} N_m \phi_{rm}^2 - \sum_{j \in I^\circ} x_{ija} = 0 \quad i \in I, a \in A \quad (\text{PF-5a})$$

$$\sum_{m: T_m = a} N_m \sum_{r \in F_q} \phi_{rm}^1 - \sum_{p \in I^\circ} x_{pqa} = 0 \quad q \in I, a \in A \quad (\text{PF-5b})$$

$$\gamma_{irm}^k \geq 0, \phi_{rm}^k = 0 \text{ or } 1, x_{ija} \geq 0, \text{ integer} \quad (\text{PF-6})$$

By convention, the indices  $r$  and  $m$  appear only in combinations that are physically allowable, that adhere to operational rules, etc. For example, a carrier could have a restriction on wide-body aircraft making multiple-stop flights, and such an  $r - m$  combination would not be permitted. The technique of forming route bundles thus allows us to incorporate many peculiarities of a system into our model that might be difficult to formulate within the (SDP) framework. Constraints (PF-2) ensure that the cargo flow on all routes that include node  $i$  satisfies the demand at node  $i$ . Constraints (PF-3) are capacity constraints for each route bundle, both ensuring that no aircraft carries more than its capacity allows, and that no flow occurs unless the appropriate route bundle is operational (i.e.,  $\phi_{rm}^k = 1$ ). Thus, these are “forcing” constraints. We model aircraft quantity limitations with constraints (PF-4). The last set of constraints, (PF-5a) and (PF-5b), model placement flights. These appear much as they do in formulation (SDP), and we can infer the same property about them concerning their modeling a transportation problem for each aircraft type.

This reformulation achieves a reduction in complexity over (SDP) first because it subsumes all time constraints into the route bundle decision variables  $\phi_{rm}^k$ . Also, these same variables satisfy the flow conservation constraints for aircraft and the forcing constraints for aircraft facilities, (1-4) and (1-3b). Finally, the route bundle variables incorporate the flow conservation constraints for cargo (1-2b).

Although (PF) greatly simplifies modeling the single-hub problem, we still face an enormous mixed-integer linear programming problem, one with a column and a row for each route bundle decision variable. Also, in choosing to streamline our model in this fashion, we implicitly create a formidable subproblem in generating route bundles. Therefore, if we do address formulation (PF), we must be judicious about which route bundles we consider directly, and we must design an efficient procedure that constructs appropriate variables as we need them.

Although we might, for example, successfully use a column-generation approach tailored to (PF), we first apply Benders Decomposition. Our motivation is twofold: first, we still seek a more tractable formulation, and

second, the process may yield added insight into the problem's structure. To simplify our exposition, we use matrix notation. Let  $\phi$  be the vector of route bundle variables,  $\gamma$  the vector of flow variables,  $F$  the coefficient matrix for  $\phi$ ,  $A$  the coefficient matrix for  $\gamma$ ,  $b$  the right-hand side vector, and  $c$  the vector of route bundle costs. Also as a simplification for exposition, we temporarily disregard the placement constraints, which allows us to separate the pickup problem from the delivery problem. Therefore, suppose that we are solving the pickup problem only. Formulation (PF) is now

$$\begin{aligned} & \text{minimize } c\phi \\ & \text{subject to} \\ & \quad F\phi + A\gamma \geq b \end{aligned}$$

The Benders restricted master problem for this reformulation is

$$\begin{aligned} & \text{minimize } z \\ & \text{subject to} \\ & \quad z \geq c\phi + (b - F\phi)^T u_i, \quad i \in J_1 \\ & \quad (b - F\phi)^T u_j \leq 0, \quad j \in J_2 \\ & \quad \phi \text{ a feasible route} \end{aligned}$$

The  $u_i$  here are extreme points or extreme rays of the cone

$$R = \{u_i \mid u_i A \leq 0, u_i \geq 0\},$$

and the sets  $J_1$  and  $J_2$  are indices for subsets of  $R$ . The usual development of Benders Decomposition includes a cost term for  $\gamma$  in the objective, say  $g\gamma$ , and the  $u_i$  in the inequalities  $z \geq c\phi + (b - F\phi)^T u_i$  are extreme points of the polytope  $S = \{u_i \mid u_i A \leq g, u_i \geq 0\}$ . Since, in our case,  $g = 0$ ,  $S = R$ .

The Benders subproblem for (PF) also differs from its usual structure. The subproblem in its primal form is

minimize  $g\gamma$

subject to

$$A\gamma \geq b - F\hat{\phi}$$

$$\gamma \geq 0,$$

where  $\hat{\phi}$  is a given route bundle vector. Since  $g$  is 0, the primal reduces to finding a feasible solution to the constraints. If the primal is feasible, then by linear programming duality theory the dual maximum objective value is zero. The dual problem (D) is

$$\text{maximize } u(b - F\phi)$$

subject to

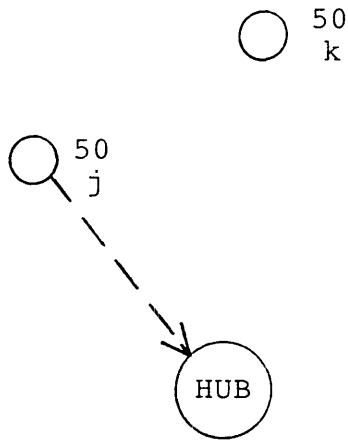
$$uA \leq 0$$

$$u \geq 0$$

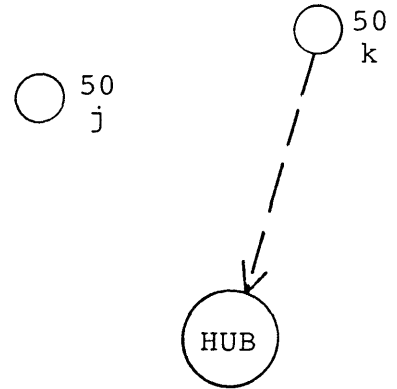
If we use the dual as the subproblem for the restricted master, we either verify optimality or generate extreme rays for the constraints, since dual boundedness implies primal feasibility and thus optimality.

An examination of the feasible solutions for (D) reveals that we can express the restricted master problem in a simpler, more intuitively appealing form. We illustrate our discussion with the example shown in Figure 3-1. This figure depicts a small system consisting of a hub and two airports. Two aircraft types are available, type 1 with a capacity of 50 units and type 2 with a capacity of 100 units. The demands are shown next to each node. (Recall that we are considering a pickup problem.) We will consider four route bundle variables, shown in Figures 3-1a - 3-1d. Formulation (PF) applied to this system yields the mixed-integer linear program (E) shown in Table 3-1, where we have simplified the subscripting. The route bundle decision variables have only a single subscript, and the cargo flow variables have two, one for the node and one for the route bundle.



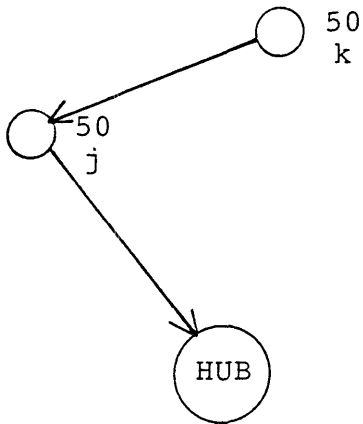


a). Route bundle 1

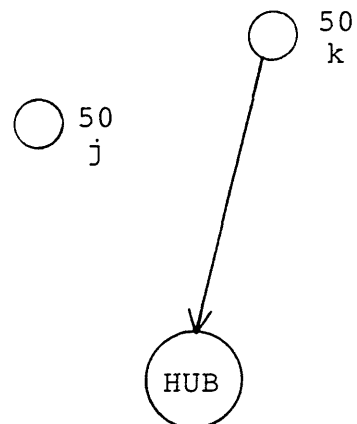


b). Route bundle 2

--- Type 1 Capacity = 50  
 ——— Type 2 Capacity = 100



c). Route bundle 3



d). Route bundle 4

**Figure 3-1. Four Route Bundles for the Example Problem**

		minimize	$\sum_{i=1}^4 c_i \phi_i$			Row	
			$\gamma_{j1}$	$+ \gamma_{j3}$	$= 50$	1	
				$\gamma_{k2}$	$+ \gamma_{k3}$	$+ \gamma_{k4} = 50$	2
(E)	$50\phi_1$		$- \gamma_{j1}$		$\geq 0$	3	
	$50\phi_2$			$- \gamma_{k2}$	$\geq 0$	4	
	$100\phi_3$		$- \gamma_{j3}$	$- \gamma_{k3}$	$\geq 0$	5	
	$100\phi_4$				$- \gamma_{k4} \geq 0$	6	
	$- \phi_1$				$\geq -2$	7	
	$- \phi_2$				$\geq -1$	8	
	$- \phi_3$		$- \phi_4$				
	$\underbrace{\hspace{10em}}_{F\phi}$			$\underbrace{\hspace{10em}}_{A\gamma}$		$\underbrace{\hspace{10em}}_{\bar{b}}$	

$$\phi_i \in \{0, 1\}, \quad \gamma_{zi} \geq 0$$

**Table 3-1. Example Pickup Problem (E)**

We want to examine the behavior of the function  $u(b - F\hat{\phi})$  for a given  $\hat{\phi}$ , where  $uA \leq 0$ . Because the term  $b - F\hat{\phi}$  has eight rows, we will consider  $A$  to have eight rows also, the last two of which are zero, in order to form the vector  $u$  with consistent dimensions. Thus, each  $u$  has dimensions  $1 \times 8$ . We now determine the form of all solutions to the system  $uA \leq 0$ ,  $u \geq 0$ . To do this, we build a set  $U$  of  $u$ -vectors that serves as a generator set for all solutions to this system. We begin the vector construction with  $u^1$ .

Suppose first that a "1" appears in position 1 of  $u^1$ . This means that at least a "1" must also appear in positions 3 and 5, because position 1 in the vector  $u^1$  corresponds to row 1 of (E) – there are two positive entries in this row,  $\gamma_{j1}$  and  $\gamma_{j5}$ , and these variables also appear, with a "-1" coefficient, in rows 3 and 5 respectively. Thus, to obtain  $u^1A \leq 0$ , there must also be at least a "1" in positions 3 and 5. Since  $u^1 = [1, 0, 1, 0, 1, 0, 0, 0]$  implies  $u^1A \leq 0$ , we set  $u^1$  to this value. By analogous reasoning, we let  $u^2 = [0, 1, 0, 1, 1, 1, 0, 0]$  and  $u^3 = [1, 1, 1, 1, 1, 1, 0, 0]$ . Note that for the general pickup problem, if a "1" appears in position  $i$  of  $u^k$ , and row  $i$  in formulation (PF) is a (PF-2) constraint, then at least a "1" must appear in each position of  $u^k$  that corresponds to a row of (PF) in which a variable from row  $i$  appears. We designate any 0-1 vector  $u^k$  that we form using this rule as type  $(i)$ . Thus, there are three type  $(i)$  vectors associated with the example problem (E).

Now suppose that  $u$  has no positive terms in any position that corresponds to a (PF-2) constraint. For our example problem (E) and the set  $U$ , we construct six vectors having this property. First, for each row  $i$  in (E) that is a (PF-3) constraint, we form a  $u$ -vector with a "1" in position  $i$  and zeros elsewhere. There are four of these  $u$ -vectors associated with (E), one for each of rows 3, 4, 5, and 6 (the first one being  $[0, 0, 1, 0, 0, 0, 0, 0]$ ). We label these vectors  $u^4$  through  $u^7$  respectively. For the pickup problem in general we classify such a  $u$ -vector as type  $(ii)$ . Finally, we construct two more vectors for  $U$  in exactly the same manner, one each for constraints 7 and 8 of (E), and label them  $u^8$  and  $u^9$  respectively. Because rows 7 and 8 in (E) are (PF-4) constraints from (PF), we classify these  $u$ -vectors as type

(iii). This completes our construction of the set  $U$ .

There are nine vectors in  $U$ : three type (i) vectors, four type (ii) vectors, and two type (iii) vectors. Any two of the type (i) vectors and all of the type (ii) and type (iii) vectors are linearly independent, so all solutions to the dual of the Benders subproblem for (E) can be generated from  $U$ , since  $A$  has eight rows. We now show that we can completely determine the behavior of the objective function  $u(b - F\hat{\phi})$  for any feasible  $u$  by observing only the elements of  $U$ .

We have observed that the special character of our application dictates determining the nature of extreme rays to the subproblem dual. This amounts to determining when  $u(b - F\phi)$  is positive. Our search examines three classes of feasible solutions to the dual; the first of these classes contains all feasible solutions formed from type (i) and type (ii)  $u$ -vectors. For our example problem, such a solution  $\hat{u}$  has the form  $\hat{u} = [\hat{u}_1, \hat{u}_2, \hat{u}_3, \hat{u}_4, \hat{u}_5, \hat{u}_6, 0, 0]$ . Since  $\hat{u}$  is feasible,  $\hat{u}A \leq 0$  and  $\hat{u} \geq 0$ , so the following conditions apply:

$$\hat{u}_1 \geq 0, \quad \hat{u}_3 \geq \hat{u}_1, \quad \hat{u}_5 \geq \max(\hat{u}_1, \hat{u}_2)$$

$$\hat{u}_2 \geq 0, \quad \hat{u}_4 \geq \hat{u}_2, \quad \hat{u}_6 \geq \max(\hat{u}_1, \hat{u}_2).$$

The validity of these inequalities is easily verified by inspection.

Let  $m = \max(\hat{u}_1, \hat{u}_2)$ , and let  $\hat{u}$  satisfy all of the above constraints with equality. Suppose that  $m = \hat{u}_1$ . Then  $\hat{u} = \hat{u}_2 \cdot u^3 + (\hat{u}_1 - \hat{u}_2)u^1 + (\hat{u}_1 - \hat{u}_2)u^7$ . Now if  $\hat{u}(b - F\hat{\phi}) > 0$  for some  $\hat{\phi}$ , then at least one of the terms  $\hat{u}_2 \cdot \hat{u}^3(b - F\hat{\phi})$ ,  $(\hat{u}_1 - \hat{u}_2) \cdot u^1(b - F\hat{\phi})$ , and  $(\hat{u}_1 - \hat{u}_2) \cdot u^7(b - F\hat{\phi})$  must be positive. Of these terms, only the first two can ever be positive since  $\hat{u}_1 - \hat{u}_2 \geq 0$  and  $u^7(b - F\hat{\phi}) \leq 0$  for any  $\hat{\phi}$ . This can be verified by direct calculation of each of these expressions, where  $b - F\hat{\phi} = [50, 50, -50\hat{\phi}_1, -50\hat{\phi}_2, -100\hat{\phi}_3, -100\hat{\phi}_4, -2 + \hat{\phi}_1 + \hat{\phi}_2, -1 + \hat{\phi}_3 + \hat{\phi}_4]$ . Thus,  $\hat{u}(b - F\hat{\phi}) > 0$  implies that either  $u^3(b - F\hat{\phi}) > 0$  or  $u^1(b - F\hat{\phi}) > 0$ , or both. We note that  $u^1$  and  $u^3$  are type (i)  $u$ -vectors and that  $u^7$  is a type (ii)  $u$ -vector. If we assume that  $m = \hat{u}_2$ , we obtain a similar result, namely that if  $\hat{u}$  is a linear combination of nonnegative multiples of type (i) and type (ii) vectors, only type (i) vectors can cause  $\hat{u}(b - F\hat{\phi})$  to be positive.

We have restricted the terms  $\hat{u}_3$  through  $\hat{u}_6$  of  $\hat{u}$ , but only for illustrative purposes. If  $u^0$  is a solution in class one such that  $u_1^0 = \hat{u}_1, u_2^0 = \hat{u}_2$ , and  $u^0 \geq \hat{u}$ , it follows from the form of  $b - F\hat{\phi}$  that  $u^0(b - F\hat{\phi}) \leq \hat{u}(b - F\hat{\phi})$ . Thus,  $u^0(b - F\hat{\phi}) > 0$  implies that  $\hat{u}(b - F\hat{\phi}) > 0$ , which in turn implies that  $u^k(b - F\hat{\phi}) > 0$  for some type (i) vector  $u^k$ . It therefore suffices to require that  $u^k(b - F\hat{\phi}) \leq 0$  for all type (i) vectors  $u^k$  in order that  $\hat{u}(b - F\hat{\phi}) \leq 0$  for any feasible solution  $\hat{u}$  in the first class. It is straightforward (but tedious) to show that the previous statement is true for the general problem (PF).

We now examine the nature of the constraints that are generated whenever  $u^k(b - F\hat{\phi}) > 0$  for a type (i) vector in the general case. Recall that a type (i)  $u$ -vector is a 0-1 vector with at least one "1" in a position that represents a (PF-2), or demand, constraint. For each "1" that appears in such a position  $i$ , a "1" also appears in every position that represents a (PF-3) constraint that contains a variable in row  $i$ . Thus, if a "1" appears in position  $i$  of  $u^k$ , and  $\gamma_j$  appears in rows  $i$  and  $m$  of (PF), then a "1" also appears in position  $m$  of  $u^k$ . Since every flow variable  $\gamma_n$  appears in exactly two constraints, one a (PF-2) constraint and the other a (PF-3), or capacity, constraint, construction of  $u^k$  is quite straightforward. There are as many (PF-2) constraints as there are nodes in the problem, so if there are  $M$  nodes, there are  $\sum_{n=1}^M \binom{M}{n}$  type (i)  $u$ -vectors, one for each subset  $S$  of  $I$ .

As we have noted, we require that  $u^k(b - F\hat{\phi}) \leq 0$  for all type (i)  $u$ -vectors  $u^k$ . The form of these constraints results from the fact that if  $u^k$  is constructed from a subset  $S$  of  $I$ , then  $u^k b = \sum_{i \in S} d_i$ , and  $u^k F\hat{\phi} = \sum_{rm: r \in R_S} K_m \hat{\phi}_{rm}$ , where  $R_S = \bigcup_{i \in S} R_i$  and  $R_i$  is the set of all routes that pass through node  $i$ . The constraints themselves are

$$\sum_{rm: r \in R_S} K_m \hat{\phi}_{rm} \geq \sum_{i \in S} d_i \quad \text{for all } S \subseteq I. \quad (\text{B-2})$$

We note that the number of (B-2) constraints is quite large for most realistic problems.

We now consider the second class of feasible solutions  $\hat{u}$  for our example problem, those having the form  $\hat{u} = [0, 0, 0, 0, 0, 0, \hat{u}_7, \hat{u}_8]$ , where  $\hat{u}_7$  and  $\hat{u}_8$

are arbitrary nonnegative numbers. Obviously,  $\hat{u} = \hat{u}_7 \cdot u^8 + \hat{u}_8 \cdot u^9$ , so for any solution  $\hat{\phi}$  to the restricted master problem we have  $\hat{u}(b - F\hat{\phi}) = \hat{u}_7 \cdot u^8(b - F\hat{\phi}) + \hat{u}_8 \cdot u^9(b - F\hat{\phi})$ . Thus,  $\hat{u}(b - F\hat{\phi}) > 0$  only if  $u^8(b - F\hat{\phi}) > 0$  or  $u^9(b - F\hat{\phi}) > 0$ , or both. The constraints that apply to the master problem are  $u^8(b - F\hat{\phi}) \leq 0$  and  $u^9(b - F\hat{\phi}) \leq 0$ . These two inequalities are quite easy to express – they are simply the last two constraints of (E).

It is straightforward to extend this discussion to the general case. We obtain the result that any feasible solution  $\phi$  to the restricted master must adhere to the fleet availability constraints. Thus, the next set of extreme rays generates the constraints

$$\sum_{m: T_m=a} \sum_r N_m \phi_{rm} \leq Q_a, \quad a \in A \quad (\text{B-3})$$

The third class of feasible solutions to the subproblem dual for our example allows positive entries in all positions of  $\hat{u}$ . However, these solutions are always linear combinations of types (i), (ii), and (iii)  $u$ -vectors, and we have already explored how these vectors behave in our example and in the general case. Thus, the third class generates no new constraints for the restricted master problem.

We are now in a position to formulate the full master problem that results from applying Benders decomposition to formulation (PF). Since the only extreme point in question is the zero vector, the Benders constraints of the form  $z \geq c\phi + u^i(b - F\phi)$  reduce to the single constraint  $z \geq c\phi$ . Summarizing, we have the following formulation of the Benders master problem.

$$\text{minimize } z = \sum_{r,m} c_{rm} \phi_{rm} \quad (\text{B-1})$$

subject to

$$\sum_{r,m:r \in R_s} K_m \phi_{rm} \geq \sum_{i \in S} d_i \quad \text{for all } S \subseteq I \quad (\text{B-2})$$

(B)

$$\sum_{m: T_m=a} \sum_r N_m \phi_{rm} \leq Q_a \quad \text{for all } a \in A \quad (\text{B-3})$$

$$\phi_{rm} = 0 \text{ or } 1 \quad (\text{B-4})$$

It is immediate that constraints (B-2) are a logical aggregate of constraints (PF-2) and (PF-3) from formulation (PF). They say that the capacity offered at every subset of nodes must be at least as great as the demand at those nodes. This property is obviously necessary for any feasible solution, and, in the absence of fleet quantity constraints, formulation (B) implies that it is sufficient as well. We shall refer to this property as the *demand-sum property*, and the constraints that express this requirement as the *demand-sum constraints*.

The combinatorial nature of the demand-sum constraints is reminiscent of Benders-derived constraints for other problem formulations in the field of combinatorial optimization. For example, Gavish and Graves [G2] derive the subtour-breaking constraints for the Traveling Salesman Problem formulation of Dantzig, Fulkerson, and Johnson [D1] by applying Benders decomposition to an assignment-based formulation of the problem. Magnanti [M1] applies Benders decomposition to a commodity-flow-based formulation of the capacitated multiple-vehicle routing problem and shows that the Benders subproblem generates constraints that enforce vehicle capacity restrictions and prohibit subtours. In fact, Magnanti's application can be viewed as a generalization of Gavish and Graves' work.

Although formulation (B) is simple and conceptually attractive, it is still a huge 0-1 integer programming problem. If we adopt a straightforward approach to Benders decomposition in this setting, we would likely restrict (B) to all of constraints (B-3) and several of constraints (B-2). The Benders subproblem is easily solved in its dual form, given an integer solution  $\hat{\phi}$  to the restricted master problem. We simply trace out each route bundle, adding the demands of nodes that are visited only by that route bundle. If the total capacity of the route bundle is greater than this sum, then we have satisfied (implicitly or explicitly) a constraint of the Benders master problem. If not, we can add the appropriate constraint to the restricted master. Once we have performed this test for each route bundle in the trial solution, we make another pass. This time we perform the test for each pair of route bundles that intersect, adding the demands at nodes that are visited by either or both of the route bundles in the pair. After

all intersecting pairs of route bundles have been tested, we make another pass, this time checking for sets of three route bundles that intersect, and so on. While the theoretical worst-case behavior of this approach might not be very good, the empirical performance would probably be quite nice, since route bundle intersections are likely to be relatively infrequent or uncomplicated.

The Benders approach nevertheless faces some significant obstacles in the restricted master problem. First, as the problem grows, we face an increasingly-large integer programming problem. Obviously, the number of route bundles could be quite large, and it would be desirable to handle implicitly as many of these as possible. Second, there is the problem of route bundle generation – we need to know what constitutes a good route bundle and how to construct one.

Desrosiers, et al. [D4] address a special case of these two difficulties in a set partitioning formulation of a routing problem with time-window constraints. Their approach uses a Dantzig-Wolfe type of decomposition, where the subproblem is a shortest-path problem with schedule constraints [D5]. They solve their restricted master problem as a linear program, adding cuts to encourage integer solutions. A major factor in the effectiveness of their approach was the high probability of obtaining an integer solution to the LP.

Although Desrosiers, et al. enjoyed considerable success with their techniques, they addressed a smaller, more-structured formulation than (B). We could conceivably tailor their general approach to our needs, but first we investigate our own set partitioning formulation for the single-hub problem.

The sequence (SDP) to (PF) to (B) exhibits a formulation pattern that is progressively less granular in nature. A set partitioning formulation to the single-hub problem could be viewed as a logical termination of this succession. As we did in applying Benders decomposition to (PF), we consider only one side of the problem and discard the placement constraints for the time being. Thus, suppose that we are addressing the pickup problem with a set partitioning model.

We essentially subsume all constraints from formulation (SDP), with



the exception of the fleet availability constraints, into the decision variable  $\sigma_j$ . Since the  $\phi_{rm}$  in formulations (PF) and (B) modeled route bundle decisions, it is logical that  $\sigma_j$  should represent an even more complex entity. Obviously,  $\sigma_j$  must denote a decision on a route or set of routes that completely serves any node that it serves, if we are to adopt a set-partition approach. We refer to such an aircraft-route structure  $\Omega$  (that  $\sigma_j$  denotes a decision about) as a *route complex*, and we let

$$a_{ij} = \begin{cases} 1 & \text{if route complex } j \text{ serves node } i \\ 0 & \text{otherwise} \end{cases}$$

Also,  $N_{aj}$  is the quantity of aircraft type  $a$  that route complex  $j$  uses,  $c_j$  is the cost of route complex  $j$ , and  $Q_a$  is the quantity of aircraft type  $a$  available. The formulation is

$$\text{minimize } \sum_{j \in J} c_j \sigma_j \quad (\text{SP-1})$$

subject to

$$\sum_{j \in J} a_{ij} \sigma_j = 1, \quad i \in I \quad (\text{SP-2})$$

$$\sum_{j \in J} N_{aj} \sigma_j \leq Q_a, \quad a \in A \quad (\text{SP-3})$$

(SP)

$$\sigma_j = 0 \text{ or } 1, \quad j \in J \quad (\text{SP-4})$$

Obviously, this is not a true set-partitioning formulation because of the side constraints (SP-3), but these constraints are few in number and will often not be binding in many practical situations.

We now develop the concept of a route complex and present a working definition. Obviously, multiple aircraft are required whenever the demand at a node is greater than the capacity of the largest available aircraft. However, such a technique is also desirable in other situations, as Figure 3-2 depicts. Consider the three nodes A, B, and C, each with a demand of 60 units, and one available aircraft type with a capacity of 90 units. Letting H be the hub, we see that the routings B-A-H and C-A-H allow only two aircraft to satisfy the pickup demand at the three nodes. It is impossible to represent these routings as two separate route complexes in (SP), since

alone neither completely serves the nodes it visits. Therefore, we use the entire structure to form a route complex. Should this complex be chosen for a solution, no other routes are necessary for nodes A and B.

It is interesting to note that, although the route complex in Figure 3-2 is potentially optimal, especially if the fixed costs of aircraft operation are high, such a structure is specifically prohibited in the classical capacitated-vehicle-routing problem. (See, for example, Magnanti, [M1].) Our models must allow for such an option, however, since an actual system could demand it.

A further desirable property of a route complex is that it be indivisible in some sense. For example, Figure 3-3 shows a situation where node D has been added to the previous system. We assign a demand of 90 to D and construct the routings B-A-H, C-A-H, and D-H. We might build a route complex from these three routes, but we could also build two route complexes, one consisting of D-H and the other consisting of B-A-H and C-A-H. Certainly, using two complexes for this example instead of one could produce a lower objective value for the overall, larger, problem. However, it is not possible to further divide these two route complexes.

The foregoing examples suggest a rule that combines with the demand-sum property to yield a working definition of a route complex. Let  $\Omega = \{\beta_{rm}\}$  be a set of route bundles, and let  $S$  consist of each node that at least one member of  $\Omega$  visits. Also, let  $S_{rm}$  be the set of nodes that route bundle  $\beta_{rm}$  visits, and for any subset  $\psi$  of  $\Omega$ , let  $S_\psi = \bigcup_{rm:\beta_{rm}\in\psi} S_{rm}$ . Thus,  $S_\psi$  consists of every node that some member of  $\psi$  visits. Finally, if  $\beta_{rm}\in\Omega$ , then set  $\phi_{rm}$  to 1; otherwise, let  $\phi_{rm} = 0$ . We define  $\Omega$  to be a route complex provided:

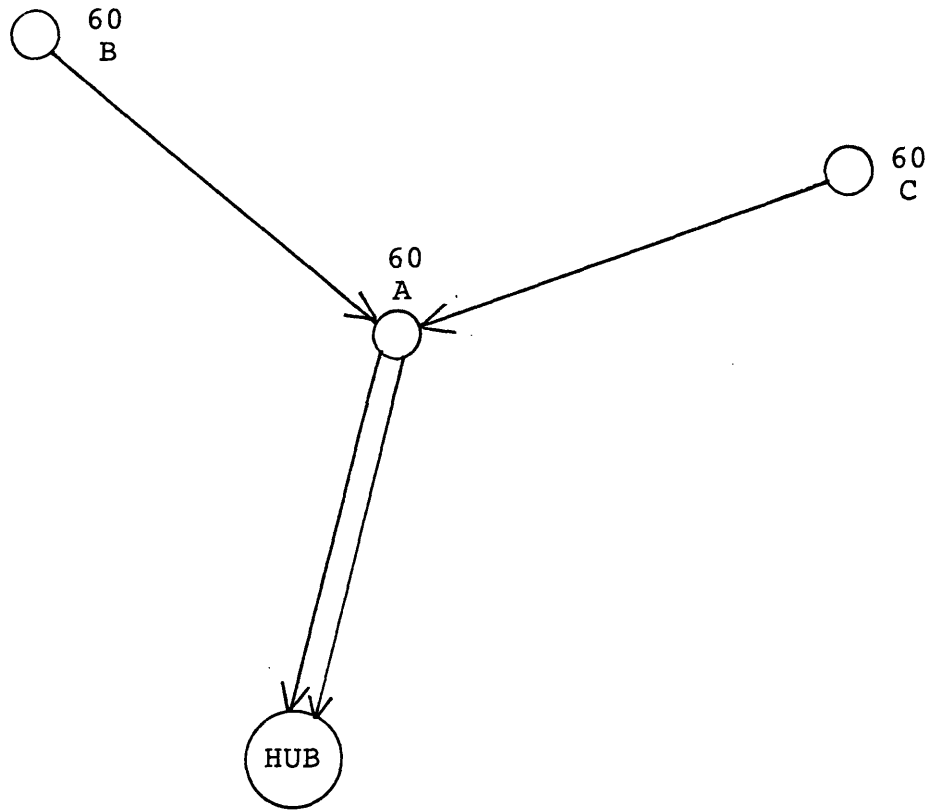
- a) If  $\Omega$  contains more than one route bundle,

$$\sum_{i\in S_\psi} d_i > \sum_{rm:\beta_{rm}\in\psi} K_m \phi_{rm}$$

for each nonempty proper subset  $\psi$  of  $\Omega$ .

- b) The demand-sum property holds for each subset  $T$  of  $S$ :

$$\sum_{rm:rc\in T} K_m \phi_{rm} \geq \sum_{i\in T} d_i$$



Aircraft capacity = 90

**Figure 3-2. Two Aircraft Serving Three Nodes**

We note that this includes the case  $T = S$ .  $R_T$  has the same definition here as in formulation (B).

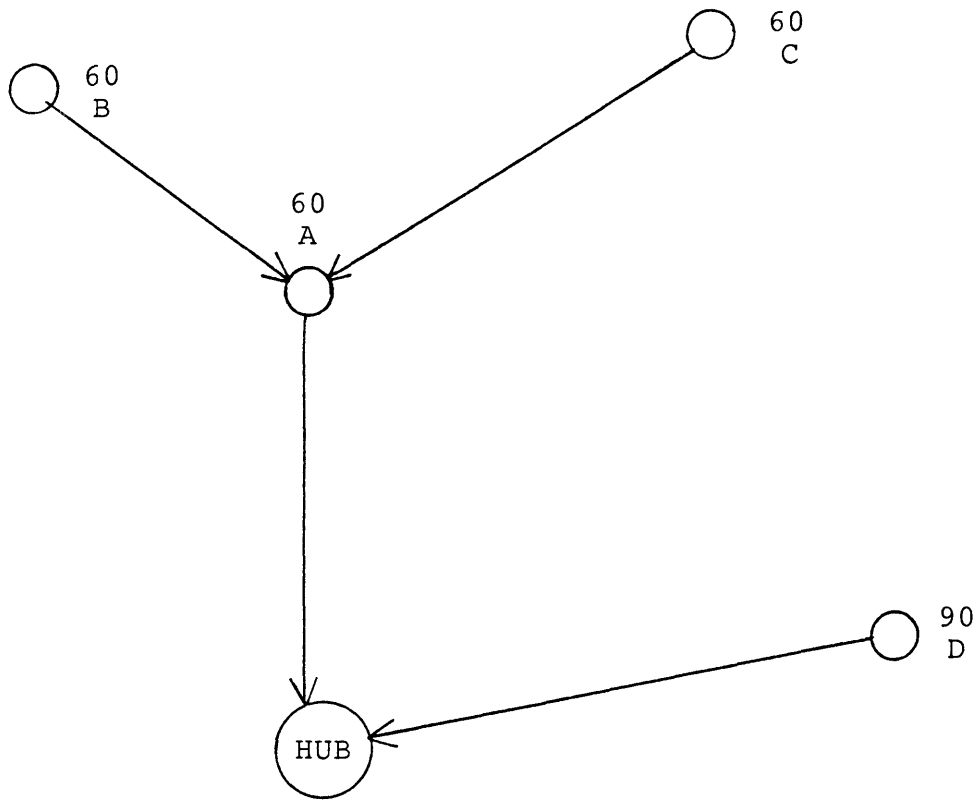
Part (a) of this definition guarantees that a route complex has the desired indivisibility property, as Figure 3-3 illustrated. Part (b) ensures that a route complex completely satisfies the demand at each node it visits. The two parts of the definition exhibit a complementarity that is analogous to necessity and sufficiency with respect to satisfying demand. Part (a) says that each subset of route bundles is necessary (in that the greater-than relation is true for all route subsets except  $\Omega$  itself), and part (b) says that the set of route bundles that visit any given subset of nodes is sufficient.

We can infer additional properties about the structure of a route complex. Suppose that we view the graph formed from a route complex when all arcs incident to the hub are deleted. Figure 3-4 depicts such a graph, where each route bundle is represented by a uniquely-drawn line,  $K_m$  represents the capacity of the route bundle, and the demand at each airport appears next to the corresponding node. In addition, for any route bundle that has only one stop, we delete its total capacity from the demand of the node it serves. Figure 3-5 shows this transformation. We term the structure that Figure 3-5b represents a *modified route complex*. We note that properties (a) and (b) in the definition of a route complex apply to a modified route complex. The following lemma demonstrates that many route complexes need not be considered.

To obtain this result, we assume the triangle inequality holds for any three airports with respect to the flight costs. This should be true in most practical situations for flight *costs*, even though it may not be true for flight *times*. For example, due to prevailing winds, it could take more time to fly from Boston to Denver nonstop than to fly from Denver to Hartford nonstop and then from Hartford to Boston. However, if cycle costs are sufficiently high, the one-stop Denver-Hartford-Boston trip will cost more than the nonstop Boston-Denver trip. We thus assume the triangle inequality and prove the following lemma.

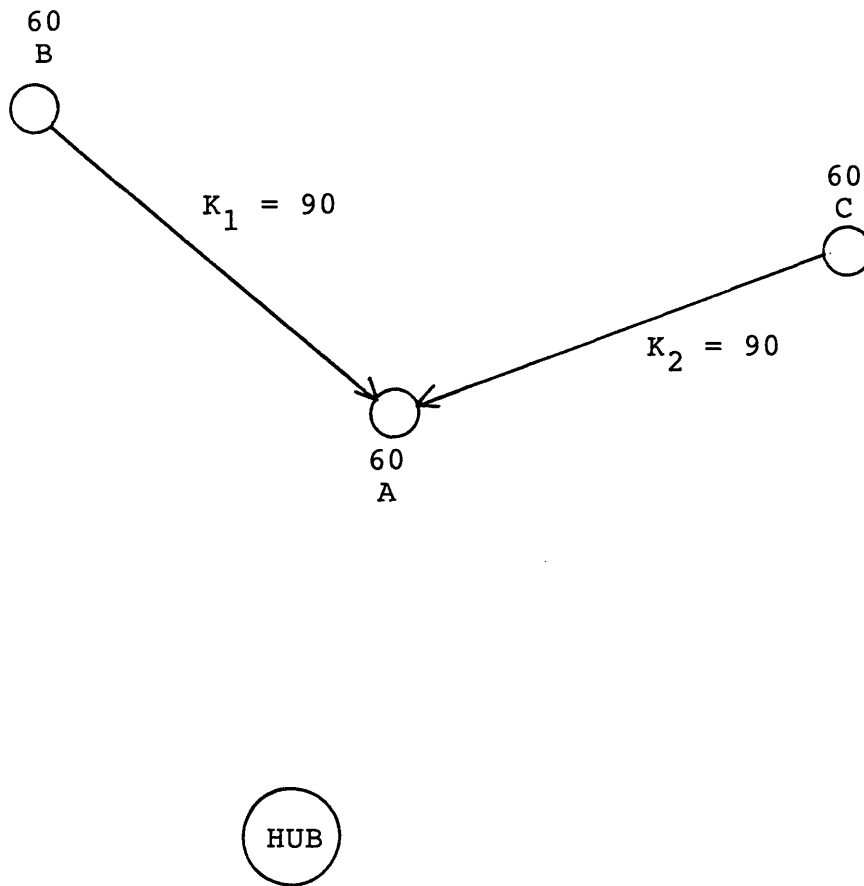
**Lemma 3.1:**

An optimal modified route complex contains no (undirected) cycles.



Aircraft capacity = 90

**Figure 3-3. A Possibility For Two Route Complexes**



**Figure 3-4. The Transformed Route Complex of Figure 3-2**

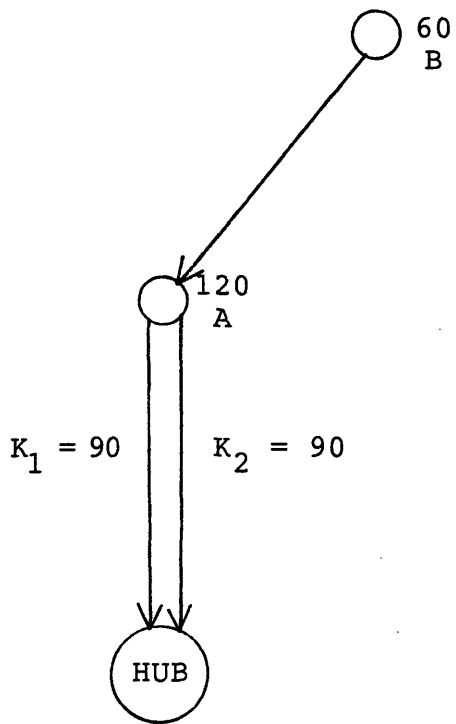
**Proof:**

Suppose that a modified route complex contains a cycle. Figure 3-6 shows an example of such a cycle, where route bundles, demands, and capacities are pictured using the same convention as in Figure 3-4. Since a route complex satisfies the demand-sum property, it is possible to make a feasible assignment of the demands to route bundles. Figure 3-7 illustrates such a feasible assignment.

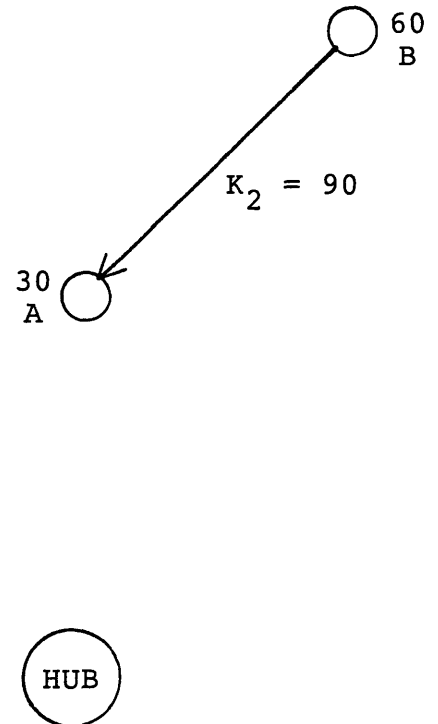
In Figure 3-8, we perform a similar assignment of node demands, but we do so only for nodes that are not in the cycle, and in a way such that an overall feasible assignment exists. Once the assignment is made, we subtract the total demand assigned to a route bundle from the capacity of that route bundle. We then remove any arc from the graph that is not in the cycle in question. As with a modified route complex, property (b) of the route complex definition applies to the remnants of the route bundles in the cycle.

Now, if any node in the cycle has only one route bundle through it, we erase the node and collapse the arcs on either side of it so that only one arc remains where there were two. In addition, we decrease the capacity of the corresponding route bundle by the weight of the node. We repeat this operation for all such nodes in the cycle. Figure 3-9 illustrates the procedure. What remains is a cycle in which each arc represents a different route bundle. Again, property (b) applies to the arcs of this cycle. Consequently, a feasible assignment of node demands to arcs (*route bundles*) exists for this cycle.

Let  $d_1, \dots, d_n$  represent the demands at nodes  $1, \dots, n$  of the cycle, respectively. Number the arcs of the cycle so that arc  $k$  is between nodes  $k$  and  $k+1$ , for  $k = 1, \dots, n-1$ ; arc  $n$  will then be incident to nodes  $1$  and  $n$ . Because a feasible demand assignment exists, we can partition  $d_k$  at each node  $k$  into two nonnegative quantities  $d_{k1}$  and  $d_{k2}$ , where  $d_{k1} + d_{k2} = d_k$ ,  $d_{k1}$  is the amount assigned to arc  $k_1$  from node  $k$  (arc  $n$  if  $k = 1$ ), and  $d_{k2}$  is the amount assigned to arc  $k_2$  from node  $k$ . The total demand assigned to arc  $k$ , for  $k = 1, \dots, n-1$ , is  $d_{k2} + d_{k+1,1} = D_k$ , and the total demand assigned to arc  $n$  is  $d_{11} + d_{n2} = D_n$ .



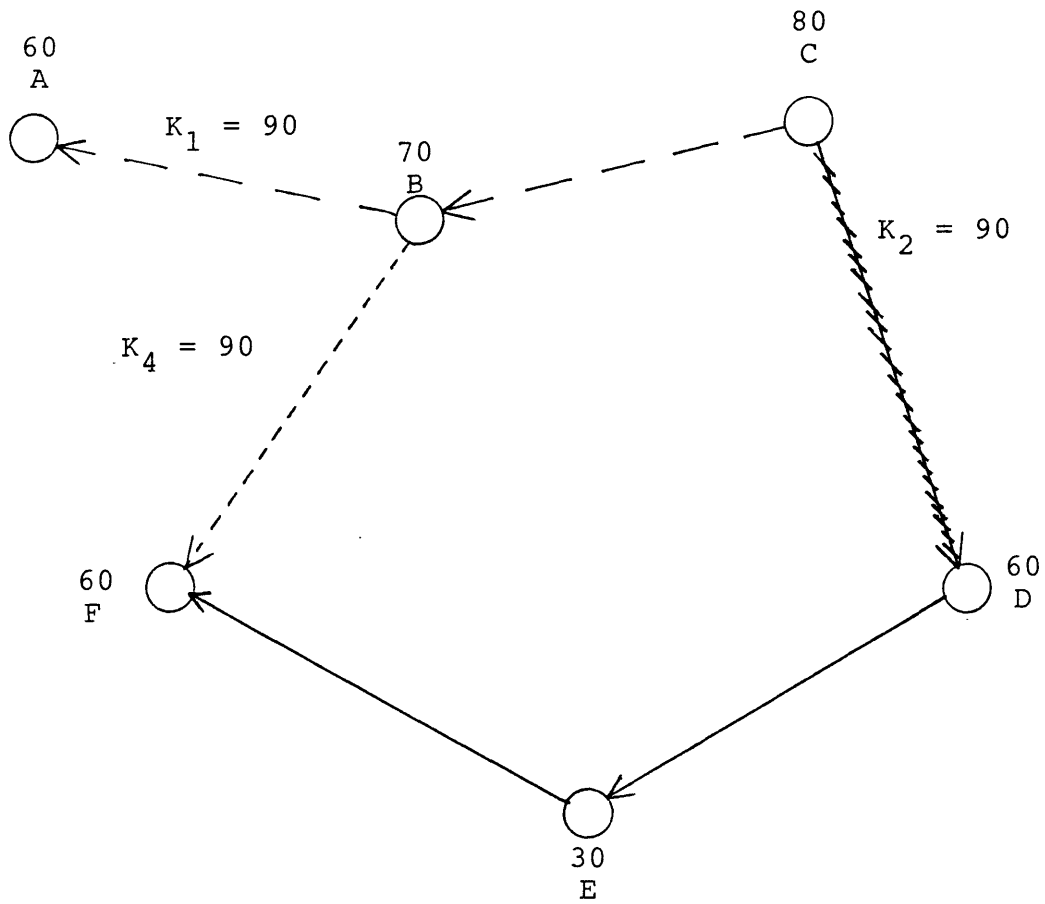
a). Original Route Complex



b). Modified Route Complex

**Figure 3-5. Complete Transformation of A Route Complex into a Modified Route Complex**





**Figure 3-6. A Cycle in a Modified Route Complex**

We now consider the set  $\{d_{11}, d_{12}, d_{21}, \dots, d_{n2}\}$ .

Let  $d^* = \min_{k,j} d_{kj}$ , and suppose that  $d_{k',j'} = d^*$ .

We then adjust all  $d_{kj}$  as follows:

$$d_{k1} \leftarrow \begin{cases} d_{k1} - d^*, & \text{if } j' = 1 \\ d_{k1} + d^*, & \text{if } j' = 2 \end{cases}$$

$$d_{k2} \leftarrow \begin{cases} d_{k2} + d^*, & \text{if } j' = 1 \\ d_{k2} - d^*, & \text{if } j' = 2 \end{cases}$$

After this adjustment, each  $d_{kj}$  is still nonnegative,  $d_{k1} + d_{k2} = d_k$  for all  $k$ ,  $d_{k2} + d_{k+1,1} = D_k$  for  $k = 1, \dots, n-1$  and  $d_1 + d_{n2} = D_n$ . Thus, the new values of  $d_{jk}$  represent a feasible demand assignment. However, at least  $d_{k',j'}$  is now zero, indicating that no demand from node  $k'$  is assigned to one of the arcs incident to node  $k$ . But this means that it is not necessary for the corresponding route bundle to visit node  $k'$ . There are two implications of this – first, it is unnecessary for the cycle to exist in order to satisfy the demand at all nodes in the route complex, and second, one route bundle has one less node to visit, thus implying a cheaper route bundle by the triangle inequality, and thus a cheaper route complex. We have shown that any modified route complex that contains a cycle can be improved, and this completes the proof.

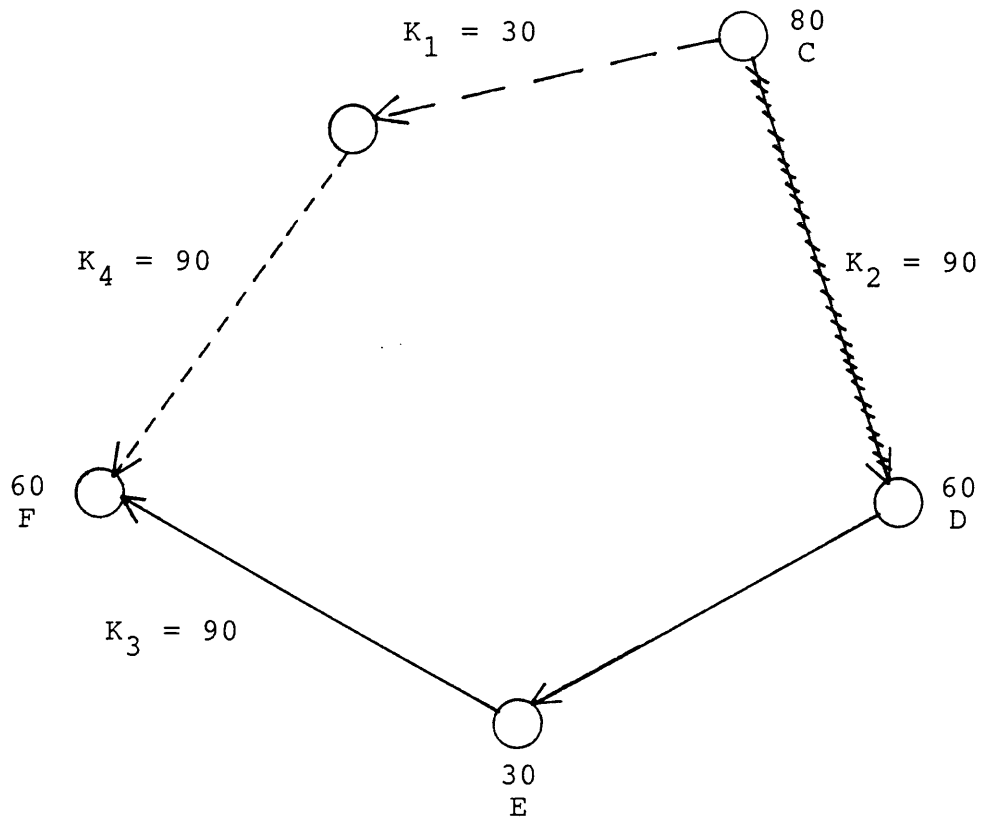
Lemma 3.1 has implications for a route complex approach to solving SHP. It says that many route complexes contain no cycles, and, if one does, two arcs of the cycle must be incident to the hub node. Equivalently, all cycles in any directed graph of a set of pickup (or demand) routes must pass through one node, namely the one representing the hub.

## 3.2 Building Route Complexes – Some Examples

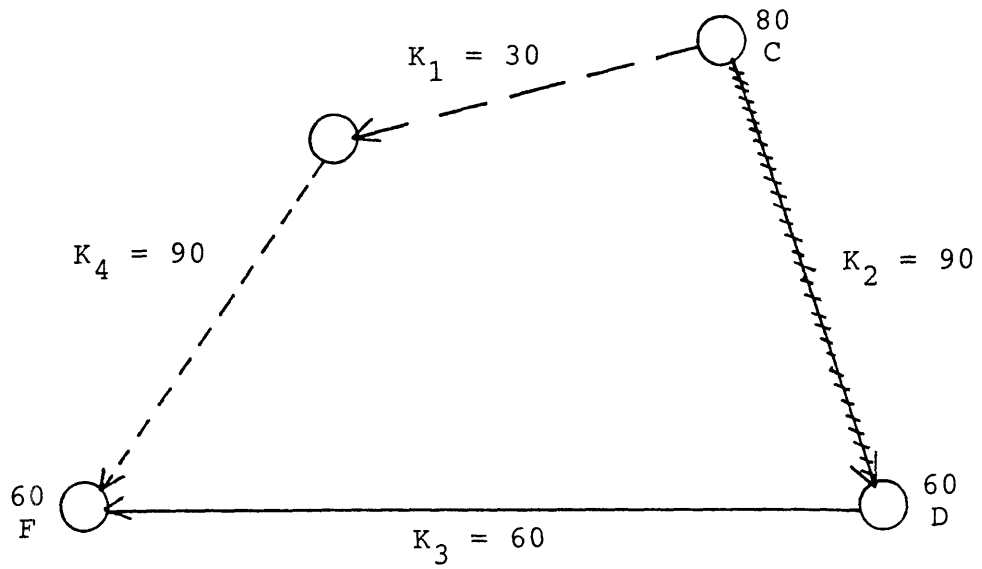
In this section we illustrate the construction of route complexes for one field node and for two field nodes, respectively. We begin with an example for one field node, and we will term each route complex that we construct

NODE	DEMAND	ASSIGNED DEMAND	ROUTE BUNDLE
A	60	60	1
B	70	10 60	1 4
C	80	20 60	1 2
D	60	30 30	2 3
E	30	30	3
F	60	30 30	3 4

**Figure 3-7. Assigning the Node Demands of Figure 3-6 to Route Bundles**



**Figure 3-8. Isolating the Cycle**



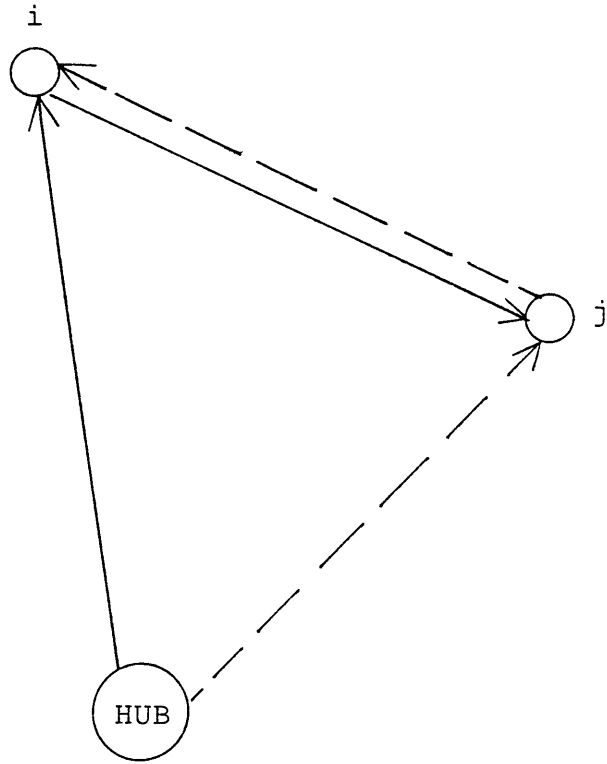
**Figure 3-9. Collapsing Arcs in the Cycle**

an *order-one* route complex, or simply a *1-plex*. In general, we will call a route complex that covers  $n$  nodes an *order- $n$*  route complex, or an  *$n$ -plex*. Suppose that we are constructing complexes for the delivery side of the problem at node  $i$ . We assume that we have two aircraft types,  $\alpha$  and  $\beta$ , with capacities of 30 and 90 units, respectively. Node  $i$  has a delivery demand of 200 units.

Constructing 1-plexes is quite simple. If we list the order-one route complexes for node  $i$  as ordered pairs, with the first element of the pair giving the number of  $\alpha$ -type aircraft and the second element the number of  $\beta$ -type aircraft, there are four such pairs: (7,0), (4,1), (1,2), and (0,3). Now add node  $j$  to the problem, with a delivery demand of 70 units. Constructing order-two route complexes is considerably more complicated than building 1-plexes. To begin, lemma 3.1 implies that at most one two-stage route exists in any 2-plex for  $i$  and  $j$ . This follows by considering that a two-stage route traverses either the sequence Hub- $i$ - $j$  or Hub- $j$ - $i$ ; two of either sequence or one of each produces the undirected cycle  $i$ - $j$ - $i$ , which lemma 3.1 shows can be improved. See Figure 3-10.

Since each of the above sequences is possible for each of the two aircraft types, there are four combinations of route and aircraft that can comprise the two-stage route of a 2-plex for this example. Because the total demand between the two nodes is 270 units, we must allocate some of this demand to one-stage routes at  $i$  and  $j$ . It is apparent that there are many ways of doing this. To facilitate this effort, we construct a type of 1-plex that *leaves* demand at its field node. The two-stage route will satisfy this remainder. For example, node  $j$  has two "natural" complexes of this type; writing these as ordered pairs using the above convention, these are (1,0) and (2,0). The *partial 1-plex* (1,0) leaves 40 units of demand unsatisfied at node  $j$ , and (2,0) leaves 10 units. In addition to these "natural" partial 1-plexes, we will use the empty 1-plex (0,0). Thus there are three partial 1-plexes associated with node  $j$ . Listing the partial 1-complexes for node  $i$  is similar, but more work, since node  $i$  has a large demand. The list is (0,0), (1,0), (2,0), (3,0), (4,0), (5,0), (6,0), (1,1), (2,1), (3,1), (0,1), and (0,2).

An example of a 2-plex for nodes  $i$  and  $j$  is the triple  $(\beta ij, (2, 0)i, (0, 0)j)$ ,



Sequences HUB-i-j and HUB-j-i  
produce the Cycle i-j-i

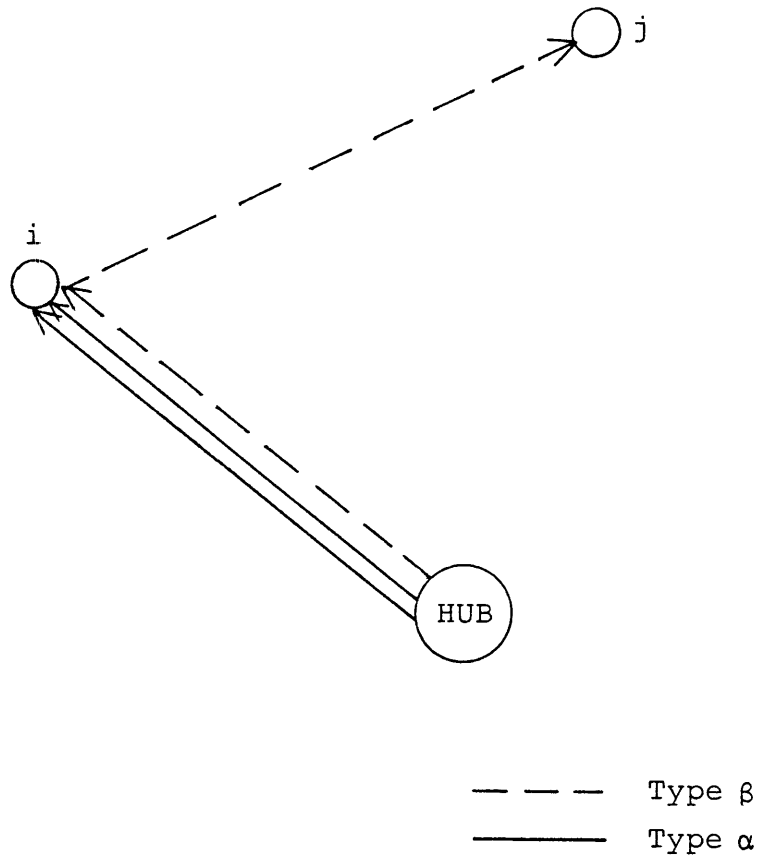
**Figure 3-10.**

where  $\beta ij$  represents an  $\beta$ -type aircraft flying from the hub to  $i$  and then to  $j$ , and  $(2,0)i$  and  $(0,0)j$ , represent the partial 1-plexes at nodes  $i$  and  $j$  respectively. The two-stage aircraft delivers 20 units to node  $i$  and 70 units to node  $j$ , which represent in turn the unsatisfied demands of  $(2,0)i$  and  $(0,0)j$ . See Figure 3-11.

If we count the possibilities for ordered triples that identify distinct 2-plexes in this example, there are four possibilities for the first element (the number of two-stage route possibilities), twelve for the second element (partial 1-plexes for node  $i$ ), and three for the third (partial 1-plexes for node  $j$ ). Thus, 144 possible 2-plexes exist for this simple example. Obviously, not every ordered triple represents a valid 2-plex. For example,  $\beta ji$  could possibly never represent a valid two-stage route due to time constraint violations in the two stage route. The empty 1-plex for node  $i$  will never be a component of a 2-plex since it leaves 200 units of demand unsatisfied, and the largest aircraft available for the two-stage route has a capacity of only 90 units. Also, certain elements in combination are invalid, as in replacing  $\beta ij$  with  $\alpha ij$  in the previous triple.

Although many two-stage routes and partial 1-plexes in combination will not form valid 2-plexes, there will still be an extremely large number of valid 2-plexes for many real-life problems. Thus, we must be judicious about handling them. Holding a place in storage for each one could be excessive. Conversely, constructing all 2-plexes from scratch every time we wish to examine them could be computationally excessive. A middle ground could be to store all combinations of city-pairs and aircraft types for which two-stage routes are time-feasible. Then, when a solution algorithm calls for an examination of 2-plexes, load feasibilities for combinations of the appropriate 1-plexes can be checked. Such an approach could be especially important when dealing with 3-plexes, for in this case explicit storage and explicit construction could very quickly become prohibitive.





**Figure 3-11.**

### 3.3 Route Complexes of Higher Order

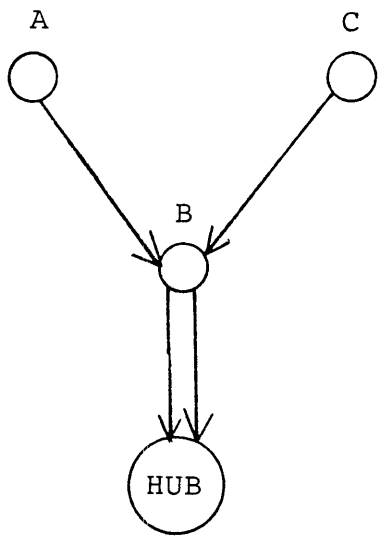
Up to this point we have considered route complexes of order two or less. In this section we discuss higher order route complexes. As we noted earlier, we are assuming that the time constraints hold route lengths down to three stages or less. However, a route complex can be of arbitrarily high order, even if no route is longer than two stages.

Nonetheless, we will limit our consideration of route complexes to those of order three or less, for two reasons. First, the cost structure could dictate using extra aircraft on very short routes (and thus smaller route complexes). Thus, we might expect only a small number of large route complexes in an optimal solution, with a small attached cost savings relative to the optimal solution without them. The second reason is operational. For example, an order-three route complex containing only two-stage routes introduces operational complications, since the load at one node must be split among extra aircraft. (See Figure 3-12.) Possibly more importantly, the additional multiple-stage routes required by higher order route complexes make recovery more difficult in the event of an aircraft failure, since the recovery aircraft would be more likely to have to fly a multiple-stage route.

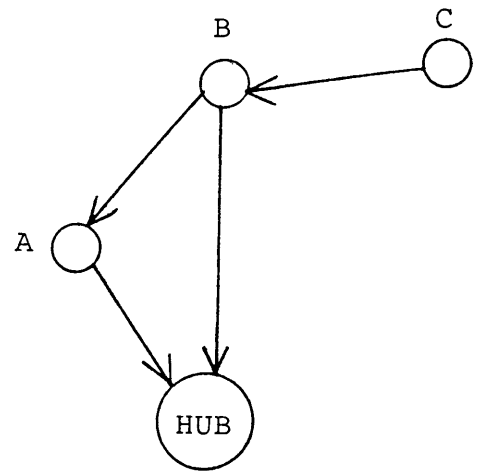
In addition, there is no uniformity of operational difficulty even among route complexes of the same order. Thus, we may wish to exclude some route complexes of order  $\rho$  while allowing others of order  $\rho$ . We briefly elaborate on this aspect of route complexes.

Figure 3-12 shows the three basic forms of a 3-plex formed from two-stage routes. We will sometimes refer to these as *star* 3-plexes to distinguish them from ones formed with a 3-leg route. There are three forms since there must be at least two two-stage routes to connect the complex and no more than two two-stage routes or a cycle results, which lemma 3.1 shows is unnecessary; For our discussion, suppose that the route complexes represent pickup routes.

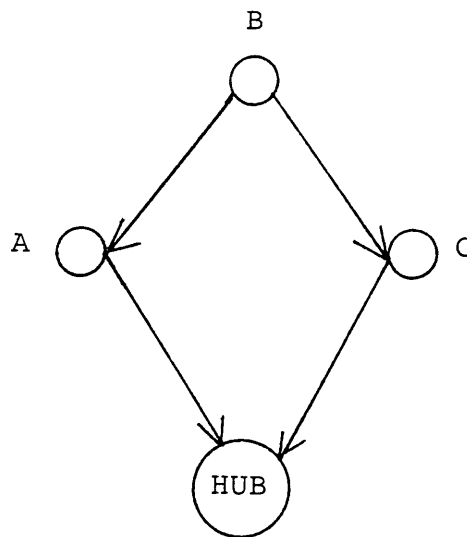
The most preferable of the three route complexes from an operational standpoint is 3-12a. The reason is that none of the three field nodes need any information about the amount of cargo at any other node in order to



(A.)



(B.)



(C.)

**Figure 3-12.**

load their aircraft; they each simply fill all one-stage aircraft first and load the remainder of the cargo on their respective two-stage aircraft. Least preferable of the three is 3-12c, since node B must have load information from *both* of nodes A and C on a nightly basis to insure that it does not overload one of its two-stage aircraft. Figure 3-10b depicts a route complex in which node B needs the load information from node A only, unless the two-stage route from node C leaves node B before the other two stage route leaves node B. (In that case, node B needs no load information.) However, in a tightly time-constrained system, two-stage routes will often have to start soon after the pickup cutoff times, so the route through B and A will likely depart before the flight from C arrives at B. Thus, 3-10b is a second choice operationally behind 3-12a.

Based on the above ordering, we may wish to bar route complexes of form 3-12c from consideration, depending on whether or not our system has an accurate and dependable information transfer network.

This is an important factor, since a high degree of reliability is necessary for the survival of any overnight system. One implication of our discussion is that a planner should carefully determine what is operationally acceptable and use this as a guide in structuring solutions to any aircraft selection and routing problem. Another implication is that the route complex approach represents a viable means of excluding undesirable route systems, since the programming required to enforce the restrictions just discussed should be relatively simple.

We now conclude Chapter 3. We have investigated several formulations and found that a set partitioning (with complicating constraints) approach appears attractive. In the next chapter we analyze the complicating constraints with respect to Lagrangian relaxation. Our focus will be on obtaining feasible solutions directly, without using branch-and bound.

## Chapter 4

# AN ANALYSIS OF THE COMPLICATING CONSTRAINTS – FORMING A SOLUTION APPROACH

In this chapter we discuss the complicating constraints for SHP when it is formulated using a route complex approach. We first generalize our model to discuss dualization of the placement constraints. In conjunction with this we outline how to extract a nonbipartite perfect matching problem from the set partitioning constraints when only 1-plexes and 2-plexes are included. After this we examine the aircraft availability constraints in some detail, especially for very simple cases. Finally, we briefly discuss the inclusion of order-three route complexes.

Rather than aggregating and analyzing all of the complicating constraints at the outset, we will deal with each in turn. Our reasons for this approach are fourfold. First, the problem with all three constraint types is quite complex, and getting even a good solution with a reasonably tight lower bound could be quite time-consuming. However, such an approach may not always be needed. For example, in a highly symmetric system

the placement constraints could be unnecessary – solving a “symmetric” problem could yield a solution that is excellent as it stands, or one that can be made viable with a few adjustments. Another possibility is that a system has no aircraft availability constraints, and system planners have discovered that route complexes of order three or greater do not offer any substantial improvement over solutions formed from 1-plexes and 2-plexes. In this case, the single-hub problem becomes two matching problems, with placement flights inducing the only complicating constraints. Each of the above situations could arise in either an operational or long-range planning scenario. Thus, examining the problem with each type of complicating constraint in isolation, and developing an efficient solution method for each, could have important implications for the airline system planner.

The second reason for approaching SHP as we propose is that we might expose more of the problem’s special structure, leading to a better solution technique for the overall problem. As an example, if the aircraft availability constraints in isolation can be dualized and the problem solved relatively quickly via subgradient optimization, we may wish to decompose the problem accordingly. That is, given a problem, we might wish to dualize all constraints except the availability constraints, leaving behind two matching problems with side constraints (the availability constraints). We would then solve each of these two problems completely, and recalculate Lagrange multipliers for the placement (and other) constraints. Such a decomposition strategy could be superior to one that dualizes all constraints initially, leaving behind two pure matching problems. We shall in fact examine just such a strategy.

A third reason for treating each constraint set separately is that we can compare the resultant systems in this way. Planners could quite possibly wish to compare solutions with and without the availability constraints in force to see what the system would use if it could. Another likely occurrence is that one would wish to compare systems with and without higher order route complexes. If order-3 complexes do not offer a significant improvement over lower order complexes, operational considerations of the type discussed in the last chapter and high computational overhead might

induce the planner to restrict the system accordingly. This would simplify the problem greatly, especially if the availability and placement constraints yield to an efficient solution technique.

Our fourth reason for studying the constraint types separately is that each resulting problem is interesting in its own right. Each has received attention in the past, in some form, although the present application does have its own uniqueness. This fact warrants a separate examination of each type of subproblem.

We wish to establish at this point that our intention is to rely directly on Lagrangian relaxation to produce feasible solutions, forgoing branch-and-bound. This is a departure from the traditional basic philosophy of Lagrangian relaxation (See Fisher [F1]), although it is often the case that the branch-and-bound is not needed, even if provided. We shall apply an optimization-based approach to a realistic *subset* of the overall problem for which a large amount of the structure remains after constraint dualization. We hope that this will empirically justify our Lagrangian relaxation strategy. In addition to this hope, we will focus on theoretical justification for attempting to obtain feasible solutions directly from the Lagrange multipliers.

## 4.1 Dualizing the Placement Constraints

We can model an instance of SHP by forming the set of all feasible route systems,  $S$ , where  $S$  contains no placement flights. For example, if  $y_1$  is a set of routes that solves a particular problem's delivery side, and if  $y_2$  solves the same problem's pickup side, then  $y = (y_1, y_2) \in S$ . If the vector  $x$  represents placement flights, then we can express SHP in the form of a mixed integer program in  $x$  and  $y$ , since we have established that we can drop the integrality requirement for the placement constraints when SHP is suitably formulated. Thus, by defining the vectors  $b$ ,  $c$ , and  $d$ , and the matrices  $A$  and  $B$  appropriately, SHP is expressible as

$$\min_{y \in S} cx + dy$$

subject to

$$(MIP) \quad Ax + By = b$$

$$x \geq 0, \quad S \text{ finite}$$

Obviously, with suitable vectors and matrices, (MIP) could model many systems. It is natural to try dualizing the placement constraints of (SDP) using Lagrange multipliers. We adopt the convention of denoting a Lagrangian relaxation of a problem (P) by (LRPu), where  $u$  is the Lagrange multiplier. The dualized constraints will either be stated explicitly or will be clear from context. If we consider two or more distinct relaxations formed by dualizing different constraints in the same problem, we will develop the appropriate notation at that time. Also,  $v(P)$  will denote the optimal value of any problem (P), and any other usage of  $v$  will be stated or clear from context. Finally,  $w(LRP)$  denotes the optimum value of the Lagrangian dual.

For our purposes, (MIP) is two set partitioning problems with side constraints. The two set partitioning problems are the pickup and delivery problems, respectively. The side constraints are the aircraft availability constraints (if any) and the end-node or placement constraints, whichever is used. For the present, we assume that the pickup and delivery problems with availability constraints are easy to solve, and we discuss dualizing the placement constraints.

We can rewrite (MIP) as

$$v = \min_{y \in S} \min_{Ax=b-By, x \geq 0} cx + dy$$

If  $S^c$  represents the convex hull of  $S$ , and  $v$  is the optimal value of (MIP), then

$$v = K_o + \min_{y \in S^c} \min_{Ax=b-By, x \geq 0} cx + dy, \text{ where } K_o \geq 0.$$

Taking the linear programming dual of the inner minimization, we obtain

$$\begin{aligned} v &= K_o + \min_{y \in S^c} \max_{uA \leq c} dy + u(b - By) \\ &= K_o + \max_{uA \leq c} \min_{y \in S^c} dy + u(b - By), \end{aligned}$$



where the reversal of the max and min operators utilizes linear programming duality theory. Because the inner minimization is over a polyhedron, for any  $u$  the optimum will be at an extreme point of  $S^c$ , which is always an element of  $S$ . Thus,  $S^c$  can be replaced with  $S$  to obtain

$$v = K_o + \max_{uA \leq c} \min_{y \in S} dy + u(b - By).$$

Consider now the effect of including the term  $(c - uA)x$ , where  $x \geq 0$ , in the inner minimization. Since  $c - uA \geq 0$ , the above product will always be zero at the optimum for any given  $u$ . Thus,

$$\begin{aligned} v &= K_o + \max_{uA \leq c} \min_{y \in S, x \geq 0} dy + u(b - By) + (c - uA)x \\ &= K_o + \max_{uA \leq c} \min_{y \in S, x \geq 0} cx + dy + u(b - By - Ax) \\ &= K_o + \max_{uA \leq c} v(\text{LRMIP}_u) \end{aligned}$$

Since the inner minimization above is valid only for  $u$  such that  $uA \leq c$ ,

$$v = K_o + w(\text{LRMIP})$$

where  $K_o$  is the duality gap.

This development shows that

$$\begin{aligned} K_o + w(\text{LRMIP}) &= K_o + \min_{y \in S^c} cx + dy \\ &\text{subject to} \\ Ax + By &= b \\ x &\geq 0 \end{aligned}$$

Since  $S^c$  is the convex hull of the finite set  $S$ , every  $y \in S^c$  can be expressed as a convex combination of the members of  $S$ . Suppose that the cardinality of  $S$  is  $M$ , and denote the members of  $S$  by  $y^t$ , where  $1 \leq t \leq M$ .

Then we can reformulate the above problem as

$$\text{minimize } cx + d \sum_{t=1}^M \lambda_t y^t$$

subject to

$$\begin{aligned} Ax + B \sum_{t=1}^M \lambda_t y^t &= b \\ \sum_{t=1}^M \lambda_t &= 1 \\ x \geq 0, \lambda_t &\geq 0 \end{aligned}$$

Rearranging terms, this system is seen to be a linear program with  $m+1$  rows (where  $b$  is  $m \times 1$ ) and many columns.

$$\text{minimize } cx + \sum_{t=1}^M \lambda_t (dy^t)$$

subject to

$$Ax + \sum_{t=1}^M \lambda_t (By^t) = b$$

(MIP<sup>c</sup>)

$$\begin{aligned} \sum_{t=1}^M \lambda_t &= 1 \\ x \geq 0, \lambda_t &\geq 0 \end{aligned}$$

We designate the above *LP* as (MIP<sup>c</sup>) because it represents the convexification of (MIP). To see this, we note that  $w(\text{LRMIP})$  is equal to  $v^c$ , where  $v^c$  is the optimal value to (MIP)'s convexification. (See, for example, Magnanti, Shapiro and Wagner [M3].) Since  $v^c = w(\text{LRMIP}) = v(\text{MIP}^c)$ , we may treat (MIP)<sup>c</sup> as the convexification of (MIP).

At this point, we wish to demonstrate that any optimal solution to (MIP) may be considered a basic feasible solution to (MIP)<sup>c</sup>.

**Lemma 4.1:**

Let (MIP) and (MIP)<sup>c</sup> be as stated, and suppose that  $x$  is  $n \times 1$ ,  $y$  is  $p \times 1$ ,  $b$  is  $m \times 1$ ,  $n > m$ , and all other matrices have conformable dimensions. Also, suppose that the rows of  $A$  are linearly independent. Then if  $y^o \in S$  and the system

$$\begin{aligned} Ax &= b - By^o \\ x &\geq 0 \end{aligned}$$

has a feasible basic solution  $x^o$ ,  $(x^o, y^o)$  yields a basic feasible solution in (MIP)<sup>c</sup>.

**Proof:**

Since  $A$  has  $m$  linearly independent rows,  $(MIP)^c$  has  $m + 1$  linearly independent rows. Thus, a basis for  $(MIP)^c$  has  $m + 1$  columns. Let  $y^r$  be an element of  $S$ , and set  $\lambda_r$  to 1. Then, for  $t \neq r$ , it follows that  $\lambda_t = 0$ . Now consider the system

$$(TP) \quad \begin{aligned} Ax &= b - By^r \\ x &\geq 0 \end{aligned}$$

If (TP) has a basic feasible solution, a basis  $A_B$  has  $m$  columns since  $A$  has  $m$  linearly independent rows. Thus, the columns containing  $A_B$  and the column containing  $y^r$  in  $(MIP)^c$  are all linearly independent and so form a basis for  $(MIP)^c$ . Since  $y^r$  was arbitrarily chosen, the lemma is proven.

The single-hub single-turn problem has the interesting property that a set of placement flights exists for any feasible set of pickup routes and delivery routes. Thus, when expressed in the form of (MIP), a feasible  $x$  exists for any  $y \in S$ . Moreover, the constraint matrix  $A$ , for a suitable formulation of SHP [see Chapter 1], is nonsingular, as can be seen by inspection. Thus, lemma 4.1 applies and any solution (to any formulation) of SHP is represented as a basic feasible solution to  $(MIP)^c$ . This includes any optimal solution to (MIP), and so a zero duality gap is possible.

We could use formulation  $(MIP)^c$  to solve the Lagrangian dual via column generation. First, we solve the relaxed problem

$$\min_{y \in S} dy - u^0 By$$

for some initial  $u^0$ , where  $u^0 A \leq c$ . We then solve a restricted version of  $(MIP)^c$  that contains only one column of the form  $By^t$ , namely the solution to the relaxation. The dual variables from the solution to the restricted form of  $(MIP)^c$  provide a new vector  $u$  to use in solving the relaxation. We continue iterating between the relaxation and  $(MIP)^c$ , using only columns of the form  $By^t$  that can be generated from known solutions to the relaxation. This process is essentially Dantzig-Wolfe decomposition, which usually converges slowly. Researchers have developed other techniques using the simplex method, including the dual simplex method (Fisher [F5]) and variants of the primal-dual simplex method (Fisher and Shapiro [F7], Fisher, Northup, and Shapiro [F6], and Marsten [M6]).

An attractive alternative with a number of successful applications is to employ dual ascent heuristics. In particular, Erlenkotter [E1] enjoyed marked success using this approach for the uncapacitated facility location problem. Fisher, et al. [F2] had excellent results with this method for real-life capacitated multiple-vehicle routing problem. Also, Wong [W2] was very successful with the Steiner tree problem on a graph, and many others. (See Magnanti, Mireault, and Wong [M2].)

One of the most popular methods for solving the Lagrangian dual is subgradient optimization. (See Shapiro [S3].) However, at each iteration this method requires that we project the Lagrange multiplier  $\hat{u}$  onto the set  $\{u : uA \leq c\}$  of dual feasible solutions in a nontrivial way.

Before we decide which method is appropriate for determining the Lagrange multiplier  $u$ , we first seek to ascertain how the structure of the problem itself can guide us. We rewrite (MIP) below as the mixed integer linear program (F).

$$\text{minimize } cx + dy$$

subject to

$$B_1y = b_1$$

(F)

$$Ax + B_2y = b_2$$

$$x \geq 0, y \in Y$$

Formulation (F) partitions all of the complicating constraints into the set

$$\{(x, y) : Ax + B_2y = b_2, x \geq 0, y \in Y\}$$

Thus, the relaxation (LRFu) below is easily solvable.

$$\text{minimize } cx + dy + u(b_2 - Ax - B_2y)$$

subject to

$$B_1y = b_1$$

$$x \geq 0, y \in Y$$

The associated Lagrangian dual is

$$\max_{u \geq 0} v(\text{LRFu})$$

A straightforward application to this problem of any of the methods discussed above does not take advantage of all the problem's underlying structure. As lemma 1.1 shows, given  $\hat{y}$ , formulation (F) reduces to  $|A|$  transportation problems, one for each aircraft type  $a$ . It would be potentially beneficial computationally if we could use information from the solutions of these transportation problems to determine new Lagrange multipliers  $u$ . This makes intuitive sense also, since the dual variables of these transportation problems represent the per-unit value of having a certain aircraft type at a given node. Thus, we might expect good feasible solutions to result. The theorem below formalizes this idea further for (MIP) in general. We first define (*SMIP* $\hat{y}$ ) as

$$\text{minimize } cx$$

subject to

$$Ax = b - B\hat{y}$$

$$x \geq 0.$$

(This is the set of transportation problems given  $\hat{y}$ .)

**Theorem 4.2**

An optimal solution  $(\hat{x}, \hat{y})$  for the Lagrangian relaxation relative to the constraints  $Ax + By = b$  that is feasible (and hence optimal) in (MIP) exists if and only if any associated optimal Lagrange multiplier  $\hat{u}$  is dual optimal in (*SMIP* $\hat{y}$ ).

**Proof:**

Suppose that  $(\hat{x}, \hat{y})$  is feasible in (MIP), and let  $\hat{u}$  be an associated optimal Lagrange multiplier; that is,  $(\hat{x}, \hat{y})$  solves

$$\min cx + dy + \hat{u}(b - By - Ax)$$

$$(LRMIP\hat{u}) \quad y \in S$$

$$x \geq 0$$

where  $v(LRMIP\hat{u}) = \max_{uA \leq c} v(LRMIPu)$ . Rearranging the objective of  $(LRMIP\hat{u})$ , we obtain  $\min dy + \hat{u}(b - By) + (c - \hat{u}A)x$ . Since  $\hat{u}A \leq c$  and  $x \geq 0$ ,  $(c - \hat{u}A)x = 0$  at the optimum, so  $(c - \hat{u}A)\hat{x} = 0$ . Because  $\hat{u}$  is feasible in  $uA \leq c$ ,  $\hat{x}$  is feasible in  $(SMIP\hat{y})$ , and  $(c - \hat{u}A)\hat{x} = 0$ ,  $\hat{u}$  and  $\hat{x}$  satisfy the complementary slackness conditions for linear programming optimality. Thus,  $\hat{u}$  is dual optimal in  $(SMIP\hat{y})$ . This proves sufficiency.

Now suppose that  $\hat{u}$  is an optimal Lagrange multiplier for  $\max_{uA \leq c} v(LRMIPu)$ , and that  $\hat{u}$  is dual optimal in  $(SMIP\hat{y})$ , where  $\hat{y}$  is optimal in  $(LRMIP\hat{u})$ . Since  $(c - \hat{u}A)x = 0$  for any optimal solution of  $(LRMIP\hat{u})$ , it follows that  $v(LRMIP\hat{u}) = d\hat{y} + \hat{u}(b - B\hat{y})$ . Moreover,  $\hat{u}$  solves  $\max_{uA \leq c} u(b - B\hat{y})$ , since it is dual optimal for  $(SMIP\hat{y})$ . Any set of optimal dual variables for  $\max_{uA \leq c} u(b - B\hat{y})$  is feasible and optimal in  $\min_{Ax=b-B\hat{y}, x \geq 0} cx$ , which is  $(SMIP\hat{y})$ . If we designate such an optimal solution  $\hat{x}$ , then  $(\hat{x}, \hat{y})$  is optimal for the Lagrangian dual, since  $(c - \hat{u}A)\hat{x} = 0$  by complementary slackness. Moreover,  $(\hat{x}, \hat{y})$  is feasible in (MIP). This establishes necessity and completes the proof.

Although it is not true that *all* dual optimal  $u$  for  $(SMIP\hat{y})$  are optimal for  $\max_{uA \leq c} v(LRMIPu)$ , it is true that any optimal  $\hat{u}$  satisfies  $uA \leq c$  and this is dual feasible, even if  $K_0 > 0$ . This fact along with Theorem 4.2 provides incentive for using dual optimal variables from  $(SMIP\hat{y})$  to generate the next Lagrange multiplier. The recently developed technique of cross decomposition (see Van Roy [V1] and [V2]) provides a framework to do this. We will use cross decomposition to address the placement constraints, and develop the particulars of our approach later in the chapter. For a description of cross decomposition, see Appendix C.

As an alternative to cross decomposition, we could use subgradient optimization to perform the Lagrangian relaxation. We choose cross decomposition because of the great success Van Roy had with the method and because of the natural way in which the transportation subproblems arise in our application. We will also test the end-node constraints as a substitute for the placement constraints, and we will use subgradient optimization for the relaxation. Feasible solutions arise easily in this setting as well, simply by solving the transportation subproblems implied by the pickup and delivery solutions.

## 4.2 Setting Up the Matching Problem

We consider once again formulation (F), where  $B_1y = b_1$  are the set partitioning constraints,  $Ax + B_2y = b_2$  are all other constraints, and  $y$  is the set of all other route complex vectors. Thus,  $b_1$  is a column of 1's, and if the constraints  $Ax + B_2y = b_2$  are dualized, a set partitioning problem remains. This relaxation is a problem whose solution is in fact a pickup solution and a delivery solution. Without the complicating constraints there are thus *two* set partitioning problems, and so we can solve each separately.

Initially, we assume that each of the set partitioning problems is polynomially solvable. Thus,  $B_1^1$  and  $B_1^2$  below contain no column with more than two 1's. We can formulate the relaxation as

$$\min cx + dy + u(b_2 - B_2y - Ax)$$

subject to

$$B_1^1y_1 = \hat{1}$$

$$(LRFu) \quad B_1^2y_2 = \hat{1}$$

$$(y_1, y_2) \in Y_1 \times Y_2 = Y$$

where  $B_1 = \begin{pmatrix} B_1^1 & 0 \\ 0 & B_1^2 \end{pmatrix}$ ,  $y = (y_1, y_2)$ ,  $\hat{1}$  is a 1-vector, and (LRFu) is the Lagrangian relaxation of  $F$  relative to  $Ax + B_2y = b_2$ . We can transform

(LRFu) into two distinct minimum weight nonbipartite perfect matching problems using a technique suggested by Magnanti [M1]. For a discussion of current matching algorithms see Ball and Derigs [B2]. For examples of other applications of matching algorithms see Ball, et al. [B1] and Bertossi, Carreresi, and Gallo [B3] for other applications of matchings to vehicle routing and scheduling. We now outline the problem transformation.

Our restrictions on  $B_1^1$  and  $B_1^2$  imply that any column from either matrix contains either one 1 or two 1's, and that the other entries are zeros. Some of the columns will be duplicated; this occurs whenever more than one route complex exists for the same set of cities. As an example, suppose that one route complex serves cities  $i$  and  $j$  on the pickup side with one DC10-10, and another route complex serves the same two cities with one B727-200 and one B727-100. Within formulation (F), the columns representing these route complexes would be distinct, due to the aircraft availability constraints and the placement constraints. However, when these constraints are dualized, the remainders of these two columns are identical, with a 1 in the rows for  $i$  and  $j$  and zeros elsewhere. Because of this, only the cheaper of the two columns need be included when solving the relaxed problem. Thus, dualizing constraints  $Ax + B_2y = b_2$  results in an immediate column compression. Once we have compressed the columns of the relaxation in the manner just described, we can transform the problem into two matching problems. We consider the pickup side, and assume that  $B_1^1$  is the appropriate coefficient matrix. Thus, we wish to solve

$$\min(d - uB_2)_1 y_1$$

subject to

$$(PM) \quad B_1^1 y_1 = \hat{1}$$

where  $(d - uB_2)_1$  is the part of  $d - uB_2$  containing only coefficients of  $y_1$ .



$$\begin{array}{c}
 \begin{array}{c} \downarrow \qquad \downarrow \\ \left[ \begin{array}{cccccc} 1 & 0 & & 1 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \cdot y = b_1 \\
 \\
 Ax + \begin{array}{c} \left[ \begin{array}{cccccc} 2 & 0 & \dots & 2 & 1 & 1 & 2 & 1 & 0 & 1 \\ 1 & 2 & \dots & 1 & 0 & 1 & 0 & 1 & 3 & 1 \\ & & & 0 & & & 1 & 2 & 0 & 2 \\ & & & & & & 1 & 0 & 2 & 1 \end{array} \right] \cdot y = b_2 \\
 \\
 \hline
 \\
 \begin{array}{c} \downarrow \\ \left[ \begin{array}{cccccc} 1 & 0 & & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & & 0 & 1 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & \dots & 0 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 0 & & 1 & 0 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{array} \right] \cdot y = b_1
 \end{array}
 \end{array}$$

Column compression when the constraints  $Ax + B_2y = b_2$  are removed (dualized). The marked columns in the top figure become identical (and thus redundant) in the lower figure.

Figure 4-1.

If there are  $n$  cities in the system including the hub, then there are  $n - 1$  rows in (PM), each row representing a field node. To set (PM) up as a matching problem, we first create a graph  $G$  of  $n - 1$  nodes. Each column in  $B_1^1$  containing two 1's corresponds to an arc in  $G$  and vice versa. Thus, if a column has a 1 in positions  $k_1$  and  $k_2$ , then nodes  $k_1$  and  $k_2$  have an undirected arc between them. We assign a weight to this arc equal to the cost of the column. Next, we create a *reflection* graph  $G'$  that is an exact duplicate of  $G$ . For reference, we number the nodes of  $G'$   $n$  through  $2n - 2$ . The reflection of node  $k \in G$  in  $G'$  is node  $k + n - 1$ . Thus, if nodes  $k_1$  and  $k_2$  in  $G$  have an arc between them, nodes  $k_1 + n - 1$  and  $k_2 + n - 1$  have a *reflection* arc between them in  $G'$ . All arcs in  $G'$  have zero weight.

We now link  $G$  and  $G'$  together using the columns in  $B_1^1$  that contain exactly one 1. There must be  $n - 1$  of these columns, for otherwise some field node is not being served. Consider any one of these columns, and suppose that it has a 1 in row  $k$ . We then construct an undirected arc between node  $k$  in  $G$  and node  $k + n - 1$  in  $G'$ , and we assign a weight to this arc equal to the cost of the column. Having carried out this operation for each such column of  $B_1^1$ , we complete our construction of the graph for which we wish to find a minimum weight perfect matching (MWPM). We denote this graph as  $H$ . Figure 4-2 shows the construction of  $H$  from a given matrix  $B_1^1$  and the associated cost vector.

We now show that solving the matching problem for  $H$  also solves (PM).

**Lemma 4.3:**

Solving the MWPM problem for  $H$  produces an optimal solution for (PM) and vice versa; the columns from  $B_1^1$  that are in the (PM) solution correspond to those arcs in the MPWM solution for  $H$  that have at least one end in  $G$ .

**Proof:**

Let  $\hat{y}_1$  be any feasible solution to (PM). A corresponding perfect matching of equal cost exists in  $H$  by the following mapping. If a column containing two 1's is in the solution  $\hat{y}_1$ , choose the corresponding arc in  $G$  and its reflection arc in  $G'$  to be in the perfect matching. Similarly, if a column with exactly one 1 is in the solution  $\hat{y}_1$ , then choose the corresponding arc

in  $H$ . Note that every time a node in  $G$  is covered, its reflection node in  $G'$  is covered. Since the solution  $\hat{y}_1$  of (PM) results in exactly one 1 for each row, each node of  $G$  covered exactly once, and thus a perfect matching of equal cost results for  $H$ . A similar argument constructs a solution  $\hat{y}_1$  for (PM) that has the same cost as any perfect matching for  $H$ . This completes the proof.

The integer programming formulation that we solve for the transformed relaxation has  $2n - 2$  rows and  $2N - n + 1$  columns, where  $N$  is the number of columns in (PM). There are presently matching codes that run in  $O(|V| |E| \log E)$  and  $O(|V|^3)$  time, where  $|V|$  is the number of vertices and  $|E|$  is the number of edges. For example, see Ball and Derigs [B2]. These codes not only have excellent worst-case bounds, but empirically run very fast. Since a large practical single-hub problem typically contains no more than 100 nodes, the corresponding matching problem will usually contain no more than 200 nodes, which is moderately sized. We would thus expect run times on this phase of our decomposition procedure to be quite fast.

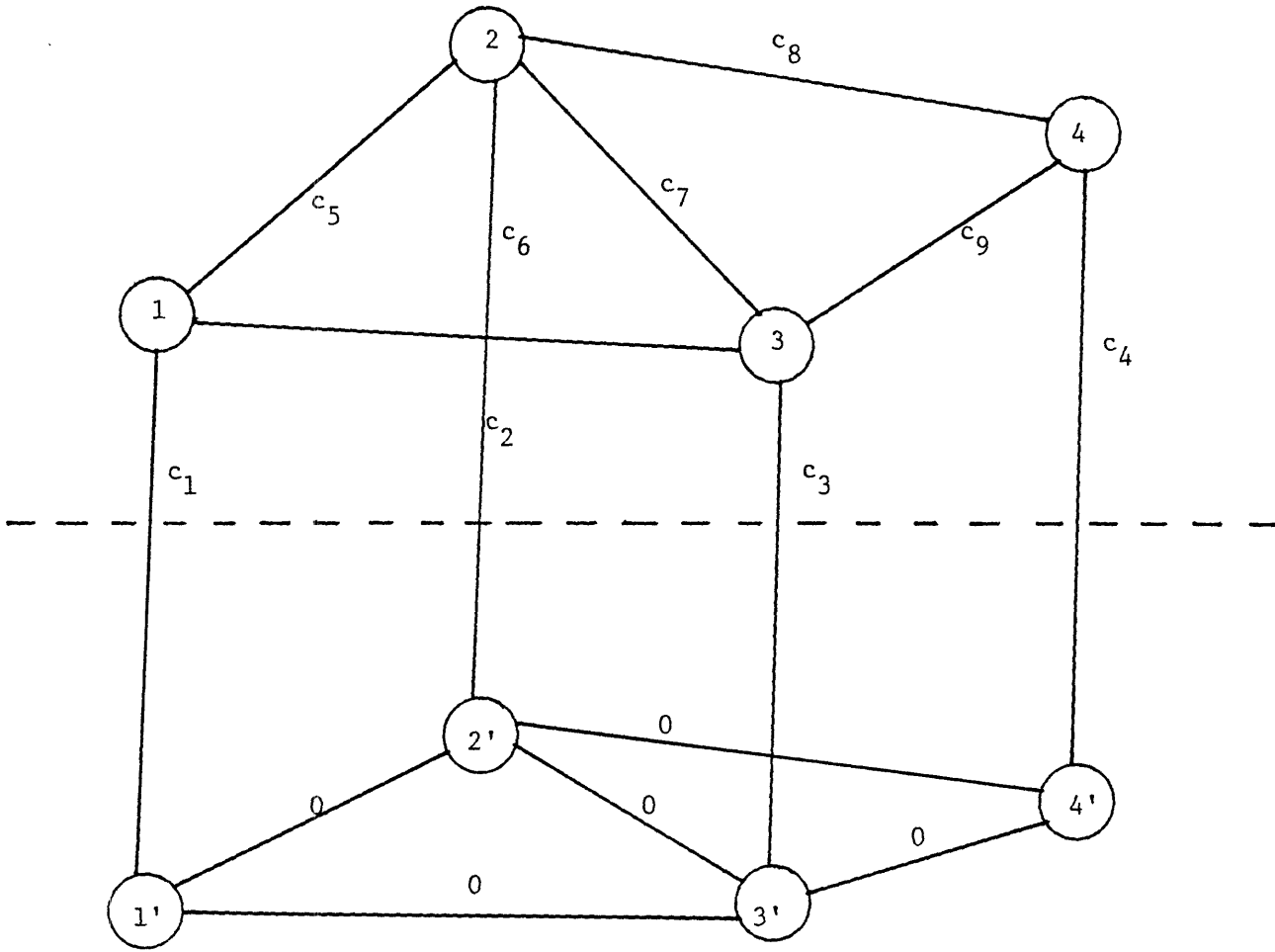
An alternative method for transforming (PM) is to add a single row to  $B_1^1$  and  $B_2^1$  that contains a 1 in each column where exactly one 1 already appears, and zeros elsewhere. The resultant coefficient matrices each have columns all of which contain exactly two 1's. The right-hand side for the extra row is 0, and the relation is *greater-than-or-equal-to*. This corresponds to a degree-constrained subgraph problem, which has been investigated by Urquhart [U1].

We have seen that the Lagrangian subproblem, where each route complex covers at most two nodes, is polynomially solvable as two nonbipartite MWPM problems. We now examine the Benders subproblem

$$\begin{aligned}
 & \text{minimize} && cx + d\hat{y} \\
 & \text{subject to} \\
 & (TP\hat{y}) && Ax = b_2 - B_2\hat{y} \\
 & && x \geq 0,
 \end{aligned}$$

where  $\hat{y}$  is given. The constraints of  $(TP\hat{y})$  consist of the fleet availabil-

		ARC AND COST								
		$c_1$	$c_2$	$c_3$	$c_4$	$c_5$	$c_6$	$c_7$	$c_8$	$c_9$
NODE	1	1	0	0	0	1	1	0	0	0
	2	0	1	0	0	1	0	1	1	0
	3	0	0	1	0	0	1	1	0	1
	4	0	0	0	1	0	0	0	1	1



Construction of the symmetric graph from the node-arc incidence matrix and the associated cost vector.

Figure 4-2.

ity constraints and the placement constraints. For now we will consider only the placement constraints, deferring discussion of the fleet availability constraints until later. Therefore, we assume that we have essentially unlimited numbers of all aircraft types under consideration. The second type of constraint in the Benders subproblem is the placement constraint. We take these to have the form

$$\sum_{j \in J^0} x_{ija} = \sum_{r \in L_i} N_{\alpha 2}^{ri} y_2^r \equiv s_{ia}, \quad i \in I^0$$

$$\sum_{i \in I^0} x_{ija} = \sum_{r \in F_j} N_{\alpha 1}^{rj} y_1^r \equiv r_{ja}, \quad j \in J^0,$$

where  $L_i$  is the set of delivery side route complexes that have at least one stop at node  $i$ ,  $N_{\alpha 2}^{ri}$  is the number of type  $\alpha$  aircraft in delivery route complex  $r$  whose last stop is at node  $i$ , and  $F_j$  and  $N_{\alpha 1}^{rj}$  have similar definitions for the pickup side. Lemma 1.1 applies to this formulation of the placement constraints. Thus, the subproblem  $(TP\hat{y})$  decomposes as  $|A|$  transportation problems, efficiently solvable by a large number of methods.

The resultant matrix  $A$  in the above formulations is totally unimodular. Thus, extreme point solutions to  $(TP\hat{y})$  are integral, and we can maintain the model's integrality (i.e., integral numbers of aircraft and flights) and justify using cross decomposition, by treating all  $x$  variables as linear.

Even though the transportation problem in general is easy to solve, we can simplify the solution in our own application. It is intuitively apparent that an aircraft should remain at an airport after finishing its delivery flight if there is a demand for that aircraft's type on a pickup flight originating at the same airport. The lemma below justifies this assumption and potentially greatly reduces the transportation problem *a priori*. We again assume that the triangle inequality holds with respect to flight costs.

**Lemma 4.4:**

Let the pickup and delivery flights be given for some instance of SHP, such that all demands and cutoff times are met. In the resulting placement flight problem, suppose that aircraft type  $a$  has a supply of  $p$  at node  $j$ , and a demand of  $q$  at node  $j'$ , where  $j$  and  $j'$  represent the same airport. Then arc  $j - j'$  has a flow of  $r = \min(p, q)$  in the optimal placement flight

solution.

**Proof:**

Consider any other proposed flow assignment out of  $j$  and into  $j'$ . In order to effect such a proposal, there must be flow on arcs of the form  $n_1 - j'$  and  $j - n'_2$ . Figure 4-3 depicts this situation.

For any arc  $n_1 - n'_b$  let  $c(n_a - n'_b)$  be the cost per unit flow for that arc. By the triangle inequality assumption,  $c(n_1 - j') + c(j - n'_2) > c(n_1 - n'_2)$ . Since  $c(j - j') = 0$ , it is also true that  $c(n_1 - j') + (j - n'_2) > c(n_1 - n'_2) + c(j - j')$ . Let  $f_1$  be the flow on  $n_1 - j'$ , and let  $f_2$  be the flow on  $j - n'_2$ . Then the above inequality shows that reassigning a flow of  $f = \min(f_1, f_2)$  from  $n_1 - j'$  and  $j - n'_2$  to  $j - j'$  and  $n_1 - n'_2$  improves the proposed solution. But this shows that flow should not exist both on  $j - n'_2$  and  $n_1 - j'$ , implying that the optimum flow across  $j - j'$  is  $r = \min(p, q)$ . This proves the lemma.

The implication of lemma 4.4 is that we can solve a (potentially significant) portion of the placement flight problem trivially before calling the transportation problem algorithm. Thus, the transportation algorithm need only work on that part of the problem that actually requires flying aircraft. If little flying is called for, we can obtain a significant computational savings.

### 4.3 Constructing the Lagrangian Dual

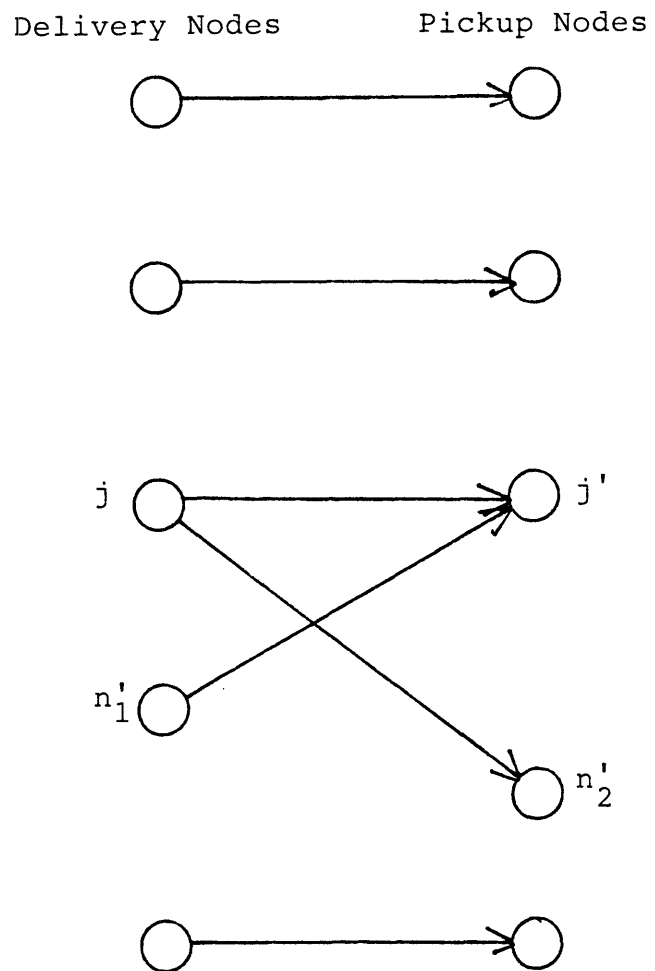
In the cross decomposition algorithm we have the option of using an *efficient cutset* of  $(TP\hat{y})$ , (see Appendix C) where  $\hat{x}$  is the current solution of  $(TP\hat{y})$ . We now examine the method of Van Roy [V2] relative to our own problem.

We reformulate the Lagrangian dual to better illustrate the logic of the dual cut procedure. Using (F), the relaxed problem (LRFu) is

$$\min_{(x,y) \in (X,Y)} cx + dy + u(b_2 - B_2y - Ax)$$

subject to

$$B_1y = b_1,$$



**Figure 4-3.      A Proposed Placement Flight Solution**

where  $X$  and  $Y$  are finite sets of integer vecors. The Lagrangian dual is then

$$\max_u \min_{(x,y) \in (X,Y)} (c - uA)x + (d - uB_2)y + ub_2$$

subject to

$$B_1y = b_1$$

Since  $(X, Y)$  is finite, this is equivalent to

$$\max w$$

subject to

$$(MD) \quad w \leq (c - uA)x^t + (d - uB_2)y^t + ub_2 \quad t \in T_D,$$

where  $T_D$  is the index set  $\{t : (x^t, y^t) \in (X, Y), B_1y^t = b_1\}$ . The dual of (MD) is

$$\text{minimize} \quad \sum_{t \in T_D} \lambda_t (cx^t + dy^t)$$

subject to

$$\sum_{t \in T_D} \lambda_t (Ax^t + B_2y^t) = b_2$$

(DMD)

$$\sum_{t \in T_D} \lambda_t = 1$$

$$\lambda_t \geq 0$$

From earlier we know that  $(MIP^c)$  (where  $B \equiv B_2$ ) and  $(DMD)$  have the same optimal value.

The dual cut generation procedure finds a basis  $TP^0\hat{y}$  for the cuts of  $TP\hat{y}$  by using  $\hat{x}$  to generate a feasible solution  $\hat{\lambda}$  to  $(DMD\hat{y})$  with the same objective value as  $(TP\hat{y})$ . We know that  $v(MD\hat{y}) \leq v(DMD\hat{y})$ , since  $(DMD)$  is the dual of  $(MD)$ . Further,  $v(TP\hat{y}) = v(MD\hat{y})$  by cross decomposition theory, so  $v(MD\hat{y}) = v(TP\hat{y}) = v(DMD\hat{y} : \lambda = \hat{\lambda}) \geq$



$v(DMD\hat{y}) \geq v(MD\hat{y})$ . Hence  $\hat{\lambda}$  is dual optimal for  $(MD\hat{y})$ , and thus the basic indices of  $\hat{\lambda}$  define an efficient cutset for  $(TP\hat{y})$ . That is,  $\{(x^t, y^t) : t \text{ is basic in } \hat{\lambda}\}$ , where  $y^t \equiv \hat{y}$ , generates an efficient cutset for  $(TP\hat{y})$ .

Van Roy shows that his dual cut generation algorithm is quite efficient and generates  $O(m)$  cuts, where  $m$  is the number of basic  $x$  variables. This is done each time the Benders subproblem  $(TP\hat{y})$  is solved, so that the number of constraints in the restricted Lagrangian master problem (or columns in the dual) grows as  $O(m)$ .

We propose to adapt Van Roy's dual cut algorithm to our own problem, as well as his dual improvement heuristic, which we discuss next.

## 4.4 Determining Lagrange Multipliers from the Benders Subproblem

The Benders subproblem  $(TP\hat{y})$  is a set of transportation problems and is consequently highly degenerate. Thus, multiple dual optimal solutions are inevitable, and we must choose the "best", or at least a good, Lagrange multiplier from among these. The theory of cross decomposition provides a guide for doing this. (See Appendix C.)

Cross decomposition theory states that the Benders restricted master problem for  $(F)$ , where the only constraints are those formed by  $\hat{u}$ , is equivalent to  $(MD)$  restricted to  $\hat{u}$ . Thus, a good Lagrange multiplier would be one that produces a "high" value of the restricted master relative to other dual optimal solutions of  $(TP\hat{y})$ . Formalizing this idea, we say that a Benders cut  $u^t b + (d - u^t)By \leq v$  "dominates" the cut  $u^r b + (d - u^r)By \leq v$  if  $u^r b + (d - u^r)By \leq u^t b + (d - u^t)By$  for all  $y$  with at least one strict inequality for some  $\hat{y}$ . A cut is said to be "strong" or *pareto-optimal* if no other cut dominates it. Thus, a strong cut corresponds to a good Lagrange multiplier.

Magnanti and Wong [M4] show how to obtain a pareto-optimal cut for any mixed integer LP, and present an efficient algorithm for finding such a cut for the uncapacitated facility location problem. Van Roy [V2] exhibits a procedure for strengthening a cut for the capacitated facility location

problem. Each uses essentially the same dual ascent method to optimize their respective objective functions. We will devise a cut-strengthening algorithm based on the same dual ascent principle. In addition, we will rely on the fact that  $(TP\hat{y})$  is block-diagonal to decompose the problem and combine the individual dual solutions to create a single Lagrange multiplier vector. Magnanti and Wong [M4] justify this technique in constructing pareto-optimal cuts for Benders decomposition.

Our algorithm first decomposes  $(TP\hat{y})$  by aircraft type and repeats the procedure for each type in the fleet. We consider  $(TP\hat{y}\alpha)$ , the restriction of  $(TP\hat{y})$  to aircraft type  $\alpha$ .

$$\text{minimize } \sum_{i \in I^0} \sum_{j \in J^0} c_{ij\alpha} x_{ij\alpha}$$

subject to

$$\sum_{i \in I^0} x_{ij\alpha} = r_{j\alpha} \quad j \in J$$

$(TP\hat{y}\alpha)$

$$\sum_{j \in J^0} x_{ij\alpha} = s_{i\alpha} \quad i \in I$$

It is well established that the above formulation is “weak”, in that the number of required Benders cuts using this formulation strategy is, in general, greater than many other less compact formulations. (See Magnanti and Wong [M4].) By recasting  $(TP\hat{y}\alpha)$  as a “stronger” formulation, we decrease the number of Benders cuts needed for convergence. Not only should such a reformulation provide us with a means for finding a strong cut, but we also improve the tightness of the Lagrangian relaxation. (See Appendix C, Theorem C.)

The Benders subproblem of the capacitated facility location problem is a transportation problem, as is  $(TP\hat{y}\alpha)$  without the two fleet availability constraints. We will concentrate on strengthening the transportation problem formulation, so for the moment we will ignore the availability constraints. The principle difference between the facility location transportation subproblem and our transportation subproblem is that the set of customers

in the former is fixed, which translates into the demand side of the transportation subproblem being fixed. In our problem *both* the source and the demand sides of the subproblem may vary from iteration to iteration. We will address this peculiarity by using a highly redundant primal formulation that allows us to treat the dual ascent in two parts, first with the demand side assumed fixed and then with the source side assumed fixed.

We introduce the new transportation problem formulation with some notation. We let  $L_{im}$  represent the set of indices of route complexes that have  $m$  aircraft of type  $\alpha$  ending their delivery routes at node  $i$ . The constant  $M_i^2$  will denote the set of all such  $m$  for each  $i \in I$ . Thus for aircraft type *alpha* in the example of 1-plex construction at the end of chapter 3,  $M_i^2 = \{7, 4, 1, 0\}$ . We define  $F_{jm}$  and define  $M_j^1$  similarly for pickup routes. Finally, we let

$$z_{jm}^1 = \sum_{p \in F_{jm}} y_1^p \quad \text{for each } j \in J \text{ and}$$

$$z_{im}^2 = \sum_{p \in L_{im}} y_2^p \quad \text{for each } i \in I.$$

Note that since  $\sum_{p \in F_{jm}} y_1^p \leq 1$ ,  $z_{jm}^1$  is a 0–1 variable for every  $j-m$  combination. The same applies to the  $z_{im}^2$  variables. We now state a stronger formulation for the transportation subproblem for aircraft  $\alpha$ ,  $(STP\alpha)$ , where each  $z$ -variable is given by summing the proper components of  $\hat{y}$ .

$$\text{minimize } \frac{1}{2} \sum_{i,j} c_{ij} \sum_{m,k} (x_{imjk}^1 + x_{imjk}^2)$$

subject to

$$(I) \left\{ \begin{array}{l} \sum_{j \in J^0} \sum_{m \in M_j^1} x_{imk}^2 = \hat{z}_{im}^2 \quad i \in I, m \in M_i^2 \\ \sum_{i \in I^0} \sum_{m \in M_i^2} m x_{imjk}^2 = k \hat{z}_{jk}^1 \quad j \in J, k \in M_j^1 \\ x_{imjk}^2 \leq \hat{z}_{jk}^1 \quad \begin{array}{l} i \in I^0, m \in M_i^2 \\ j \in J^0, k \in M_j^1 \end{array} \end{array} \right. \quad (STP\alpha)$$

$$(II) \left\{ \begin{array}{l} \sum_{i \in I^0} \sum_{m \in M_i^2} x_{imjk}^1 = \hat{z}_{jk}^1 \quad j \in J, k \in M_j^1 \\ \sum_{j \in J^0} \sum_{m \in M_j} k x_{imjk}^1 = m \hat{z}_{im}^2 \quad i \in I, m \in M_i^2 \\ x_{imjk}^1 \leq \hat{z}_{im}^2 \quad \begin{array}{l} i \in I^0, m \in M_i^2 \\ j \in J^0, k \in M_j^1 \end{array} \end{array} \right.$$

$$x_{imjk}^1, x_{imjk}^2 \geq 0 \text{ for all } i, j, k, m$$

The decision variable  $x_{imjk}^l$  denotes the portion of  $m$  aircraft of type  $\alpha$  that ferry from node  $i$  to node  $j$ , for  $l = 1$  or  $2$ . As we have noted, the  $z$  variables are either 0 or 1. Moreover, for  $l = 1$  or  $2$ , we have  $\sum_{m=1}^{M_i^l} \hat{z}_{im}^l \leq 1$  for every node  $i$  due to the way in which  $\hat{z}_{im}^l$  is defined. From this it follows that  $m = s_{i\alpha}$  if and only if  $m \hat{z}_{im}^2 = s_{i\alpha}$ , and  $m = r_{j\alpha}$  if and only if  $m \hat{z}_{im}^1 = r_{j\alpha}$ . Otherwise,  $m \hat{z}_{im}^k = 0$  for  $k = 1$  or  $2$  and any node  $i$ . From this we can conclude that constraint set (I), given  $\hat{z}$ , is equivalent to

$$\sum_{i \in I^0} x_{ij\alpha} = r_{j\alpha} \quad j \in J$$

$$\sum_{j \in J^0} x_{ij\alpha} = s_{i\alpha} \quad i \in I$$

$$x_{ij\alpha} \leq r_{j\alpha} \quad j \in J, i \in I$$

An analogous property holds for constraint set (II). It is evident from this reasoning that each of the constraint sets (I) and (II) define the same set of transportation problem solutions as (TP $\hat{y}$  $\alpha$ ). We thus halve the objective function value to obtain the true cost.

The dual of (STP $\alpha$ ) is

$$\text{maximize } \sum_{j,k} \hat{z}_{jk}^1 v_{jk}^1 + \sum_{i,m} (m \hat{z}_{im}^2) u_{im}^1 - \sum_{i,m,j,k} \hat{z}_{im}^2 w_{imjk}^1$$

$$+ \sum_{i,m} \hat{z}_{i,m}^2 v_{im}^2 + \sum_{j,k} (k \hat{z}_{jk}^1) u_{jk}^2 - \sum_{i,m,j,k} \hat{z}_{jk}^1 w_{im,jk}^2$$

subject to

$$v_{jk}^1 + k u_{im}^1 - w_{im,jk}^1 \leq \frac{1}{2} k c_{ij} \quad i, m, j, k$$

$$v_{jk}^1 \leq \frac{1}{2} k c_{oj} \quad j, k$$

$$k u_{im}^1 - w_{imok}^1 \leq \frac{1}{2} k c_{io} \quad i, m, k$$

$$v_{im}^2 + m u_{jk}^2 - w_{im,jk}^2 \leq \frac{1}{2} m c_{ij} \quad i, m, j, k$$

$$v_{im}^2 \leq \frac{1}{2} m c_{io} \quad i, m$$

$$m u_{jk}^2 - w_{om,jk}^2 \leq \frac{1}{2} m c_{oj} \quad j, m, k$$

$$w_{im,jk}^1, w_{im,jk}^2 \geq 0 \text{ for all } i, m, j, \text{ and } k.$$

We can easily obtain an initial feasible solution for (DSTP $\alpha$ ) by solving (TP $\hat{y}\alpha$ ) and making an assignment of the resultant dual variables. The dual of (TP $\hat{y}\alpha$ ) is

$$\text{maximize } \sum_{i \in I^0} s_i v_i + \sum_{j \in J^0} r_j u_j$$

subject to

$$v_i + u_j \leq c_{ij} \quad i \in I, j \in J$$

$$v_i \leq c_{io} \quad i \in I$$

$$u_j \leq c_{oj} \quad j \in J$$

An assignment of any optimal solution ( $u^*, v^*$ ) to the variables of (DSTP $\alpha$ ) that optimizes its objective is

$$\begin{aligned}
v_{jk}^1 &\leftarrow \frac{k}{2}u_j^* \\
u_{im}^1 &\leftarrow \frac{1}{2}v_i^* \\
v_{im}^2 &\leftarrow \frac{m}{2}v_i^* \\
u_{jk}^2 &\leftarrow \frac{1}{2}u_j^* \\
w_{im,jk}^1, w_{im,kj}^2 &\leftarrow 0
\end{aligned}$$

To strengthen the Benders cuts associated with (DSTP $\alpha$ ) and thereby improve the current Lagrange multipliers we will attempt to increase the coefficients of all  $\hat{z}_{jk}^1$  and  $\hat{z}_{im}^2$  that are currently zero. Thus, if  $\hat{z}_{im}^2 = 0$ , we have formulation (SC) below, in which we

$$\text{maximize } mu_{im}^1 - \sum_j \sum_k w_{im,jk}^1$$

subject to

$$v_{jk}^1 + ku_{im}^1 - w_{im,jk}^1 \leq \frac{1}{2}kc_{ij} \quad j, k$$

$$ku_{im}^1 - w_{im,jk}^1 \leq \frac{1}{2}kc_{io}$$

where we hold  $v_{jk}^1$  constant. We will adapt Van Roy's algorithm for strengthening Benders cuts to our problem. We omit the details for this and the generation of efficient cuts, but the adaptations are straightforward. The only significant difference is that for Van Roy's application the demand nodes are given at the beginning of the problem. For our problem both source and demand for the transportation subproblems can change from iteration to iteration. Thus, when creating the efficient set of cuts, we first treat the source nodes as given and create the cuts, then treat the demand nodes as given and create another set of cuts. Both sets together comprise an efficient cutset. When strengthening the Benders cuts to improve the Lagrange multipliers, we increase coefficients for both  $\hat{z}_{jk}^1$  and  $\hat{z}_{im}^2$  that are zero. For further details, see Van Roy [V2].

## 4.5 The Aircraft Availability Constraints and the Symmetric Problem

In this section we focus on obtaining feasible solutions to the single hub problem relative to the aircraft availability constraints. To this end we can implicitly guarantee a solution to the placement constraints by creating a symmetric problem. That is, we solve a single problem that solves the delivery problem by flying the indicated routes outward from the hub and solves the pickup problem by flying the indicated routes into the hub. Alternatively, we could ignore the placement constraints and solve the pickup and delivery sides separately, to assess our handling of the availability constraints. Eventually we must solve each side separately relative to the availability constraints in order to address the overall problem, but we introduce the symmetric solution strategy at this point because it is computationally useful. Our technique for satisfying the availability constraints will apply in either case. As in previous sections, we limit the candidate route complexes to those that cover at most two cities.

To create a symmetric solution, we first adjust the demands at each field node  $i$ . We define new demands  $\hat{d}_i^1$  and  $\hat{d}_i^2$  for every  $i$ , where  $\hat{d}_i^1 = \hat{d}_i^2 = \max\{d_i^1, d_i^2\}$ . Second, we require that any route adhere to all cutoff times both outbound from the hub and inbound to the hub. Thus, all solutions will be symmetric. There are two potential drawbacks to this technique as a practical method of obtaining solutions. First, we may induce a significant degree of suboptimality in any solution obtained, especially if the actual demands and time constraints skew the true problem. Second, and potentially more serious, we may induce infeasibility in the problem if the availability constraints are tight. However, availability constraints are often loose enough to permit feasibility, and thus a symmetric solution potentially represents a first cut at a good or even optimal solution.

Regardless of whether we solve the symmetric problem or the pickup and delivery sides separately, let  $y^m$  represent the  $m^{\text{th}}$  route complex, and let  $N_\alpha^m$  be the number of type  $\alpha$  aircraft that route complex  $m$  uses. Also, let  $q_\alpha$  be the available number of type  $\alpha$  aircraft. We can then write the

aircraft availability constraints as

$$\sum_m N_\alpha^m y^m \leq q_\alpha \quad \alpha \in A$$

We begin our analysis of dualizing these constraints by considering the simplest case, where  $|A| = 1$ . We could create a greater degree of freedom in multiplier adjustment by adopting a finer-grained formulation strategy, as we did in adding the extra constraints to the placement flight problem. However, in this instance such an approach may not be necessary. Lemma 3.1 provides a means of studying the structure of route complexes that will aid us in our determination. The following lemma applies the principle of acyclic modified route complexes to our current problem.

**Lemma 4.5:**

Suppose that the single-leg (order-one) route complexes for nodes  $i$  and  $j$  that use only aircraft type  $\alpha$  require  $m^1$  and  $m^2$  aircraft, respectively. Then any order-two route complex containing  $i$  and  $j$  (using only type  $\alpha$  aircraft) requires  $m^1 + m^2 - 1$  aircraft, provided such a route complex exists.

**Proof:**

We first suppose that an order-two route complex for  $i$  and  $j$  exists (e.g. time constraints allow it). Then by lemma 3.1 we may assume that the 2-plex has exactly one two-stage route, flown by a single aircraft that visits  $i$  and  $j$ , in the route complex. If there is more than one aircraft on a two-stage route, we obtain a cycle and can improve the route complex using the same number of aircraft. Figure 4-4 illustrates this. (There must be at least one aircraft on a two stage route in order to “hold”  $i$  and  $j$  together.) We now ascertain how many aircraft are on single-stage routes in the route complex.

Let  $\hat{m}^1$  and  $\hat{m}^2$  be the number of single-stage routes to  $i$  and  $j$ , respectively, in the 2-plex. Clearly,  $m^1 > \hat{m}^1$  and  $m^2 > \hat{m}^2$ , for otherwise the two-stage route would be unnecessary and the 2-plex would violate part (a) of the route complex definition. However, because only one aircraft flies a two-stage route, there must be less than an aircraft load for it to carry at both  $i$  and  $j$ . Thus,  $\hat{m}^1 \geq m^1 - 1$  and  $\hat{m}^2 \geq m^2 - 1$ . These two inequalities combined with the previous two show that  $\hat{m}^1 = m^1 - 1$  and  $\hat{m}^2 = m^2 - 1$ .



Therefore, the total number of aircraft in the order-two route complex containing  $i$  and  $j$  is  $\hat{m}^1 + \hat{m}^2 + 1 = m^1 - 1 + m^2 - 1 + 1 = m^1 + m^2 - 1$ , and the proof is complete.

Applying lemma 4.5 to a special case, we can guarantee a zero duality gap when the single aircraft availability constraint is dualized. Let (RC2) denote either a symmetric or one-sided (pickup or delivery) single-aircraft-type route complex formulation of the single-hub problem in which all route complexes are order-two or less. We then have the following result .

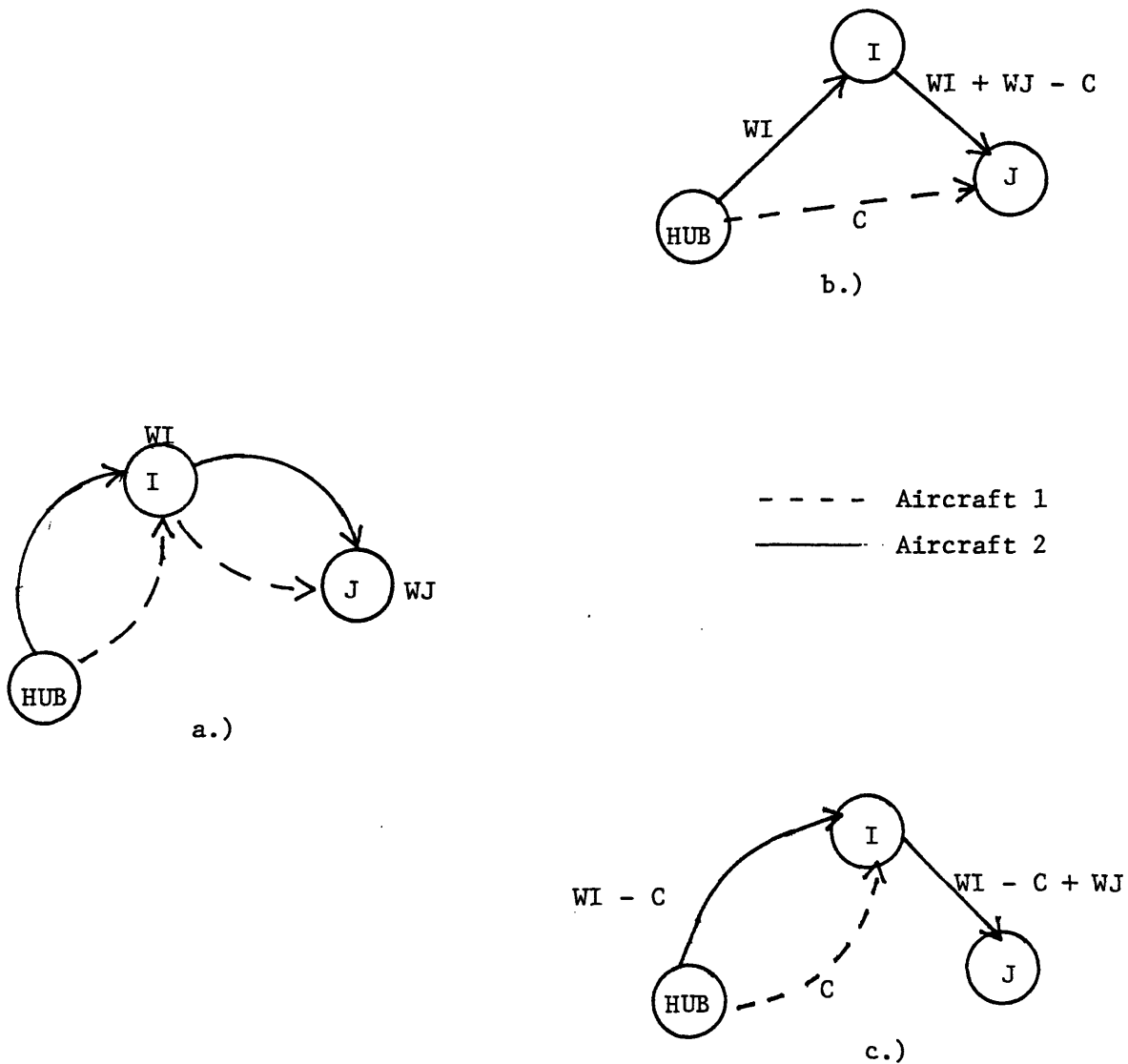
**Proposition 4.6:**

Suppose that an instance of (RC2) is a pure set partitioning problem when the aircraft availability constraint is dualized. Then the Lagrangian relaxation (LRRC2u) with respect to this constraint has a zero duality gap, with an associated optimal aircraft availability-feasible solution, provided a feasible solution exists for (RC2).

**Proof:**

The proof is trivial if removing the constraint produces a feasible solution. Suppose then that an infeasible solution results when the availability constraint is ignored. The penalty term in the objective function of (LRRC2u) is  $u \left( \sum_m N_m y^m - q \right)$ , where  $u \geq 0$ . We consider how the solution of (LRRC2u) changes as  $u$  increases linearly.

A count of the aircraft in any solution is expressible as the sum of aircraft in order-one route complexes plus the sum of aircraft in order-two route complexes. To this end let  $M_i$  be the number of aircraft in the 1-plex that covers node  $i$ , and let  $M_{j,k}$  be the number of aircraft in the 2-plex that covers nodes  $j$  and  $k$ . Let  $O$  be the set of nodes covered by 1-plexes in some solution  $y$ , and let  $T$  be the set of node pairs covered by 2-plexes in the same solution. The total number of aircraft in  $y$  is  $m = \sum_{i \in O} M_i + \sum_{(j,k) \in T} M_{j,k}$ . By lemma 4.5,  $\sum_{(j,k) \in T} M_{j,k} = \sum_{(j,k) \in T} (M_j + M_k - 1)$ , so  $m = \sum_{i \in I} M_i - |T|$ , where  $I$  is the index set of all field nodes. Thus, the number of aircraft in any solution is dependent only on the *number* of 2-plexes as opposed to *which* 2-plexes. We now claim that for  $u$  large enough, we maximize the possible number of 2-plexes in any solution to (RC2). Let  $c_i$  and  $c_{k,j}$  be the costs of the 1-plex



For the two aircraft to be able to pick up the loads  $W_I$  and  $W_J$  at  $I$  and  $J$  respectively, it must be the case that  $W_I + W_J \leq 2 * C$ , where  $C$  is the capacity of each aircraft. But then either  $W_I \leq C$  or  $W_J \leq C$ ; in the first case, figure b) is an alternate, cheaper solution than a), and in the second case figure c) is a cheaper solution. The total load on each aircraft after the pickup at the node at the head of each arc is shown next to the arc.

**Figure 4-4. Improving a Route Complex With a Cycle**

covering node  $i$  and the 2-plex covering nodes  $j$  and  $k$ , respectively. We can then express the cost of any solution as

$$\begin{aligned}
v &= \sum_{i \in O} c_i + M_i u & + \sum_{(j,k) \in T} c_{jk} + M_{jk} u - qu \\
&= \sum_{i \in O} c_i + M_i u & + \sum_{(j,k) \in T} c_{jk} + (M_j + M_k - 1)u - qu & \text{by lemma 4.5} \\
&= \sum_{i \in O} c_i + \sum_{(j,k) \in T} c_{jk} & + u \sum_{i \in I} M_i - |T|u - qu.
\end{aligned}$$

Since  $u \sum_{i \in I} M_i$  and  $-qu$  are part of the cost for any solution, it follows that the optimal solution minimizes

$$c' = \sum_{i \in O} c_i + \sum_{(j,k) \in T} c_{jk} - |T|u$$

Thus, for  $u$  large enough, we maximize the size of  $T$  in an optimal solution to a relaxed problem. Since the number of aircraft in any solution is  $\sum_{i \in I} M_i - |T|$ , choosing  $u$  this large minimizes the number of aircraft in any possible solution to (RC2). Such a solution must be therefore feasible relative to the availability constraint, since we have assumed that a feasible solution to (RC2) exists.

We have shown that increasing  $u$  from zero in a continuous fashion results in optimal solutions to the relaxation that use decreasing numbers of aircraft, and that a feasible solution eventually results. To complete the proof, we need only show that there exists  $\hat{u} > 0$  such that  $\hat{u} \left( \sum_{y^m \in y^*} N_m Y^m - q \right) = 0$ , where  $y^*$  is the set of route complexes in the optimal solution to (LRRC2 $\hat{u}$ ). To do this, we show that we can increase  $|T|$  in increments of 1 as  $u$  increases from zero.

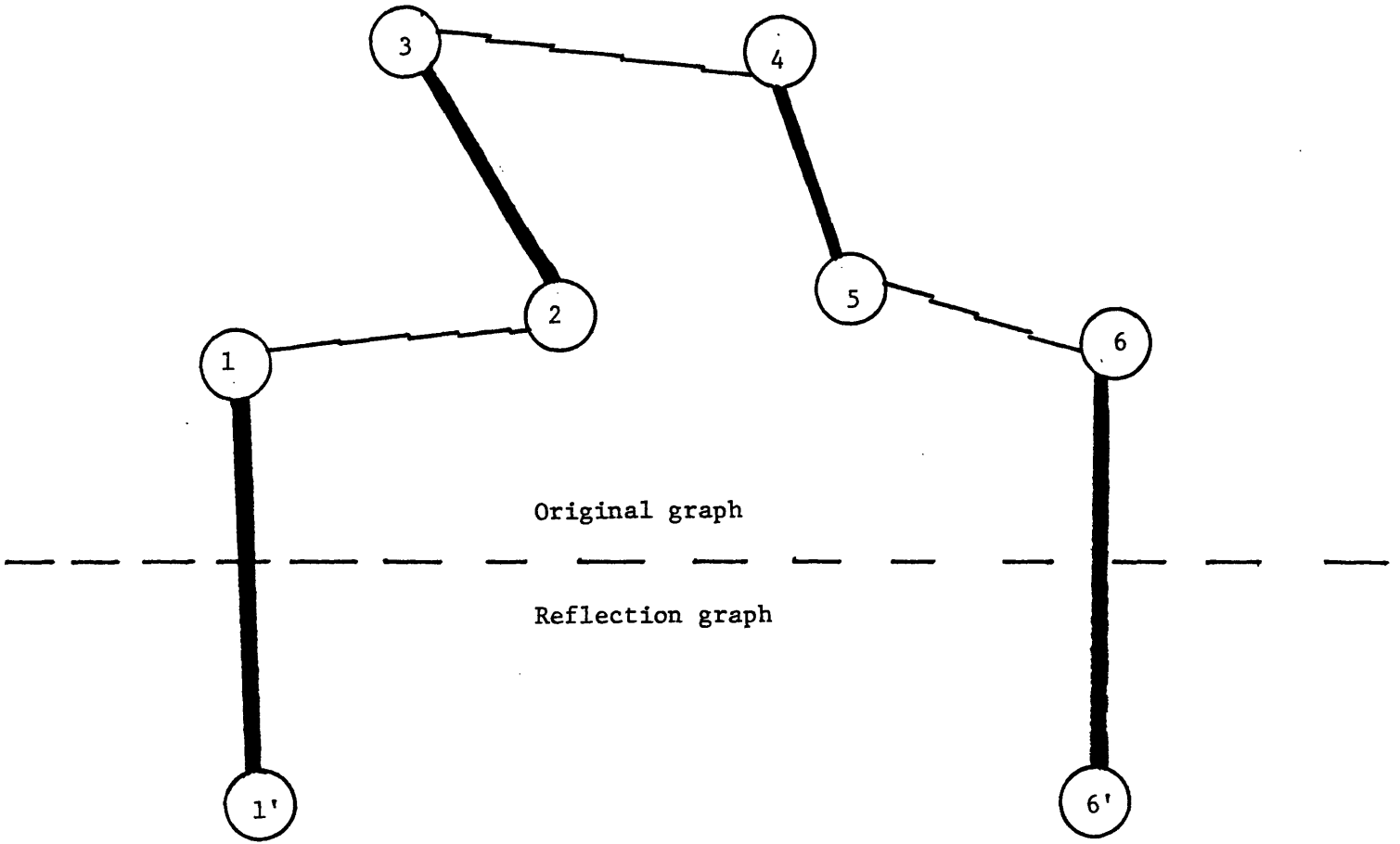
If, at every solution change in performing the algorithm, the number of 2-plexes increases by one, there is nothing to prove. For, at some point, the Lagrangian penalty term in the objective function will then equal zero, since we have shown that we eventually arrive at a feasible solution. Thus, suppose that the number of 2-plexes increases by more than one at some change in the solution. Let  $\hat{u}$  be the largest  $u$  that does not actually *force* this change, and let  $T'$  be the set of 2-plexes in the new solution as  $u$  increases from  $\hat{u}$ .

We consider a matching problem to demonstrate our result. The graph that we will use is the symmetry graph that we constructed at the beginning of this chapter. Since the MWPM solution for this graph represents an optimal solution to  $(LRRC2u)$ , we will observe what happens to the solution when  $u$  becomes larger than  $\hat{u}$ . Figure 4-8 shows a graph of this type where the bold edges are in the old optimal matching, and the jagged edges are in the new optimal matching.

We have seen that, in the new solution, the number of edges representing order-two complexes (we call these edges *2-edges* or *2-arcs*) increases. Thus, the number of order-one complexes (represented respectively by *1-arcs*) must decrease. The implication for node 1 in Figure 4-5 is that some jagged 2-arc must be incident to another node that is already covered by a bold 2-arc (node 2 in the example). If we traverse this arc to its other incident node (node 3), we see that a jagged 2-arc must cover it in the new matching. Continuing in this way, we can trace a path of 2-arcs that is alternating in bold and jagged edges. The path begins with node 1' (node 1's reflection node), goes to node 1, then to node 2, etc. in Figure 4-5.

Because our constructed alternating path begins at a reflection node, it must end at this same node. This must occur, since otherwise the path would turn back in on itself at some intermediate node. However, this would imply that two edges from the same matching are incident to each other, which cannot occur. Therefore, the path must at some point turn back to node 1'. The only way for this to happen is that the path must traverse another 1-arc at some node (node 6 in Figure 4-5). As we have seen previously, in the graph that we have constructed it is always possible to consider any matching to be symmetric in 2-arcs; that is, reflection arcs have zero cost. Consequently, when our alternating path next traverses a 1-arc, we may duplicate the preceding path of 2-arcs with a path of reflection 2-arcs. This creates an alternating cycle, as Figure 4-8 shows. Thus, every solution change as  $u$  increases involves alternating cycles of the type just described, each one traversing exactly two 1-arcs, and containing two distinct paths of 2-arcs.

Consideration of the paths of 2-arcs shows that each one contains exactly



**Figure 4-5. Finding Alternating Cycles in the Symmetry Graph**

one more edge from the new matching than from the old matching. This corresponds to one more order-two route complex in the new solution than in the old solution for each alternating cycle of this kind. For  $u = \hat{u}$ , the new and old matchings have equal value, so for any of these alternating cycles the sum of the bold edge costs equals the sum of the jagged edge costs. Thus, we can create a sequence of optimal solutions for (LRRC2 $\hat{u}$ ) in which the number of order-two route complexes in the solution increases by one each time, until an optimal solution is reached that is also optimal for  $u$  slightly greater than  $\hat{u}$ . This shows that we can create the desired sequence of solutions and completes the proof of Theorem 4.4.

Derigs [D3] refers to the alternating cycle that we constructed in the proof above as a *negative alternating cycle*. It is so named because of the relative costs of the edges in the cycle. Let  $\Gamma_o$  and  $\Gamma_n$  be the edges of alternating cycle  $\Gamma$ ; let  $\Gamma_o$  be those that are in the incumbent matching and  $\Gamma_n$  those that are *not* in the matching. Also, let  $c(\Gamma_o)$  and  $c(\Gamma_n)$  be the sums of the edge costs in  $\Gamma_o$  and  $\Gamma_n$ . We define the cost of the cycle to be  $c(\Gamma) = c(\Gamma_n) - c(\Gamma_o)$ . An alternating cycle  $\Gamma$  is *negative* if  $c(\Gamma) < 0$ . The following result applies to any perfect matching.

**Theorem 4.7 (Derigs)**

A perfect matching is minimal if and only if it admits no negative alternating cycle.

Using Theorem 4.7, we can narrow the search for alternating cycles as  $u$  increases from its value at a solution change. Consider any alternating cycle consisting only of 2-arcs (with nonzero costs  $c_{ij}$ ). In calculating the cost of such a cycle, each term  $M_i u$  appears once with a positive sign and once with a negative sign. Also,  $u$  itself appears equally often with positive and negative signs. We see this in the cost of an edge, which is  $c_{ij} + M_{ij} u = c_{ij} + (M_i + M_j - 1)u$ . Thus, the cost of any cycle of this kind depends on the original costs  $c_{ij}$  and not on the value of  $u$ . It follows that no alternating cycle of 2-arcs ever becomes negative as  $u$  increases. We express the important implication of this for our problem in the lemma below.

#### Lemma 4.8

As  $u$  increases from zero in (LRRC2 $u$ ), the only alternating cycles relative to the incumbent optimal solution that become negative contain 1-arcs.

Derigs [D3] uses the negative alternating cycle concept as a foundation to devise a *shortest augmenting path* algorithm for MWPM problems. (An augmenting path for a matching is an alternating path that begins and ends with arcs that are not in the matching.) We calculate the cost of an augmenting path exactly as we do the cost of an alternating cycle. We will adapt the notion of a shortest augmenting path to (RC2) to determine exactly when an alternating cycle becomes negative and which arcs in the incumbent optimal solution of (LRRC2 $u$ ) change. Consider an optimal solution to (LRRC2 $u$ ) when  $u = 0$  and the corresponding optimal matching in the appropriate symmetry graph. We now remove all reflection nodes from the graph, leaving only original nodes and 2-arcs between original nodes. Any original node that was covered by a 1-arc is now left exposed. Our aim is to find the shortest alternating (i.e., augmenting) path between every pair of exposed nodes.

Consider any alternating path  $\rho$  between two arbitrary exposed nodes 1 and  $\lambda + 1$  and let  $c_{kl}$  be the cost of any 2-arc between nodes  $k$  and  $l$ . Let  $M_{kl}$  be the number of aircraft in the corresponding route complex. As before, let  $M_i$  be the number of aircraft in the order-one route complex that covers node  $i$ ; also, let  $c_i$  be the cost of that route complex and the corresponding 1-arc. Finally, let  $\lambda$  be the number of arcs in an augmenting path  $\rho$  between nodes 1 and  $\lambda + 1$ . Then with Lagrange multiplier  $u$ , the cost of  $\rho$  is

$$\begin{aligned}
c(\rho) &= \sum_{i=1}^{\frac{1}{2}(\lambda+1)} (c_{2i-1,2i} + M_{2i-1,2i}u) \\
&\quad - \sum_{i=1}^{\frac{1}{2}(\lambda-1)} (c_{2i,2i+1} + M_{2i,2i+1}u) \\
&= \sum_{i=1}^{\frac{1}{2}(\lambda+1)} c_{2i-1,2i} - \sum_{i=1}^{\frac{1}{2}(\lambda-1)} c_{2i,2i+1} + (M_1 + M_{\lambda+1} - 1)u \\
&= c'_\rho + (M_1 + M_{\lambda+1} - 1)u.
\end{aligned}$$

The term  $(M_1 + M_{\lambda+1} - 1)u$  results from applying lemma 4.5 and noting that the  $M_i$  “telescope”. Thus the coefficient of  $u$  in  $c(\rho)$  is independent of the actual path and depends only on the first and last nodes.

The path  $\rho$  is part of an alternating cycle  $\Gamma$  in the symmetry graph that consists of the reflection arcs of  $\rho$  and 1-arcs between the ends of the two paths. Since the arcs of the reflection path have zero cost, the cost of the cycle  $\Gamma$  is given by

$$\begin{aligned}
c(\Gamma) &= c(\rho) - (c_1 + M_1u + c_{\lambda+1} + M_{\lambda+1}u) \\
&= c'_\rho - c_1 - c_{\lambda+1} - u
\end{aligned}$$

We note that the coefficient of  $u$  is entirely independent of the number of aircraft involved, and that the costs of any two alternating cycles of this type differ only by the arc costs of the respective paths. Therefore, among all alternating cycles  $\Gamma$  that contain an augmenting path  $\rho$  between nodes 1 and  $\lambda + 1$ , the one whose cost becomes negative first is the one with the smallest  $c'_\rho$ . It follows that finding the appropriate path for each exposed node pair (when all 1-arcs are removed from the symmetry graph) will allow us to determine exactly how and when to alter the initial optimal matching as  $u$  increases from zero. The steps below outline a suitable overall algorithm for (RC2).



## (RC2) Solution Procedure

1. Obtain an optimal matching for (LRRC2 $u$ ) when  $u = 0$ . If the availability constraint is satisfied, stop; this solution is optimal and feasible. Otherwise, do step 2.
2. Remove all reflection nodes (if any remain) from the symmetry graph, and find the shortest augmenting path between all pairs of exposed original nodes. Do step 3.
3. Calculate  $c(\Gamma)$  for each path, and find the smallest of these. Do step 4.
4. Augment the matching for the reduced symmetry graph using the shortest augmenting path chosen in step 3. Do step 5.
5. If the availability constraint is satisfied with equality, stop. The present matching and 1-arcs that cover the still-exposed nodes comprise the optimal solution. If the availability constraint is not satisfied and more than one exposed node exists, repeat step 2. Otherwise, stop; no feasible solution exists.

It is clear that the above procedure is quite efficient, with the possible exception of finding the shortest paths. First, constructing the route complexes is  $O(n^2)$ , and solving (LRRC2 $_{u=0}$ ) is  $O(n^3)$ . Finding the smallest of all shortest paths (once all  $c(\Gamma)$  have been found) is  $O(n^2)$ , and augmenting the matching each time is  $O(n)$ . Steps 2 through 5 are repeated at most  $\frac{n}{2}$  times. It only remains to show that we can find the shortest alternating paths efficiently to conclude that the entire procedure is polynomially solvable. In fact, a modification to the Floyd-Warshall all-pairs shortest path algorithm accomplishes this.

Briefly recalling the algorithm, it proceeds at the  $k^{\text{th}}$  step by finding the shortest path between nodes  $i \neq k$  and  $j \neq k$  using only nodes 1 through  $k$ . To do this it uses the formula

$$d_{ij}^k = \min \{ d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1} \},$$

where  $d_{ij}^k$  is the length of arc  $i - j$  if it exists and  $\infty$  otherwise. The algorithm maintains a path matrix  $[r_{ij}^k]$  that is updated by the formula

$$r_{ij}^k = \begin{cases} k & \text{if } d_{ij}^{k-1} > d_{ik}^{k-1} + d_{kj}^{k-1} \\ r_{ij}^{k-1} & \text{otherwise} \end{cases}$$

For a graph with  $n$  nodes, the final shortest path from  $i$  to  $j$  is given by the sequence  $(r_{ik_1}^n, r_{k_1k_2}^n, \dots, r_{k_pj}^n)$ .

Our proposed modification is as follows. Since we wish to have an alternating path, we will allow an intermediate node  $k$  to be considered in the path between  $i$  and  $j$  only if the arcs of the path incident to  $k$  are alternately in and out of the current matching. More formally, let  $g = r_{ki}^{k-1}$  and  $h = r_{kj}^{k-1}$ . Then if exactly one of the arcs  $k-g$  and  $k-h$  is in the current matching, we allow node  $k$  to be considered as a possible intermediate node in the shortest alternating path between  $i$  and  $j$ . Otherwise we skip over  $k$  as a possibility.

The modification guarantees that any path constructed will be alternating, and it does not prevent any alternating path from being constructed. Thus, the modified Floyd-Warshall algorithm will accomplish the desired task, in  $O(n^3)$  time, provided all shortest alternating paths are well-defined. Lemma 4.8 guarantees that this is the the case, by showing that successive optimal solutions of (LRRC2u) involve no negative alternating cycles consisting only of 2-arcs. Since a shortest alternating path between two nodes in the reduced symmetry graph is well-defined provided no negative alternating cycles exist, we conclude that the modified Floyd-Warshall algorithm will work properly. We can now state the computational complexity result.

**Theorem 4.9:**

Problem (RC2) is polynomially solvable, in  $O(n^4)$  time, where  $n$  is the number of field nodes.

**Proof:**

The most time-consuming steps of the (RC2) solution algorithm are 1 and 2, and these are  $O(n^3)$  each. Since step 2 is repeated at most  $\frac{n}{2}$  times, the entire process is  $O(n^4)$ , and this completes the proof.

It is of interest to note that any method used for obtaining an optimal Lagrange multiplier will result in an optimal feasible solution to (RC2). We

assume for the purpose of discussion that  $|T|$  as defined earlier increases in increments of 1 as  $u$  increases continuously and that the availability constraint cannot simply be ignored, but that a feasible solution to (RC2) does exist. Let  $u^*$  be the smallest optimal value for  $u$ . We have already seen that  $u^*$  produces a feasible solution to (RC2). By lemma 4.6, as  $u$  increases from  $u^*$  the next forced change in solution occurs when an alternating cycle with a 1-arc in it goes negative, if any change occurs at all. If such a change does occur, say at  $u = \hat{u}$ , the term  $u \left( \sum_m N_m y^m - q \right)$  becomes negative, and continues to decrease until another solution change occurs as  $u$  increases. Since the Lagrangian dual is concave, we can infer that  $v(LRRC2u) < v(LRRC2\hat{u})$  for all  $u > \hat{u}$ . A similar argument shows that  $v(LRRC2u) < v(LRRC2u^*)$  for all  $u < u^*$ . Thus, the optimal set of Lagrange multipliers is  $\{u : u^* \leq u \leq \hat{u}\}$ . Moreover, any  $u$  in this set will produce a *feasible* solution to (RC2).

The implication of the above discussion is that we could use a method as simple to program as subgradient optimization or an ascent algorithm with a binary search on  $u$  to solve the Lagrangian dual and obtain a *feasible* solution to (RC2) at optimality. This makes subgradient optimization an attractive method to try for a problem with more than one aircraft type.

We cannot directly extend our development for a single aircraft type to the same problem with multiple aircraft types, since we are then dealing with a symmetry graph that has multiple edges. Although we only choose the cheapest edge out of all edges between two nodes  $i$  and  $j$  as part of the matching problem, varying the Lagrange multipliers could cause a solution change other than by rotating all the edges of a negative alternating cycle. The solution could change from a simple switching of two of the multiple edges between a single pair of nodes. This switch could involve a 1-arc or a 2-arc. Consequently, a straightforward application of the concepts of negative alternating cycles and shortest augmenting paths will not address the problem adequately.

Because the subgradient optimization method is attractive for (RC2), we will use it for problems involving multiple aircraft types. We will use a single availability constraint for each aircraft type, since this technique is

quite sufficient for solving (RC2) optimally and efficiently. Moreover, the subgradient optimization method is easily extendable to problems involving route complexes of order greater than two. Obviously, dualizing only the availability constraints for a problem of this complexity leaves behind a relaxation that is itself  $Np$ -complete. Thus, we must devise a strategy for dealing with set partitioning problems in which a column can have more than two 1's. This is the topic of the next section.

## 4.6 The Column-Joining Constraints

In adding 3-plexes to SHP we destroy the special matching structure that we have been able to address. We propose to remedy this by introducing a new set of constraints to the problem, first researched by Nemhauser and Weber [N1]. Any column of  $B_1$  containing three 1's will now appear as two nonzero columns, the sum of which equals the original column. In addition, a new constraint will "join" the two columns together. For example, if  $y^m$  represents a 3-plex, we replace it in the set partitioning constraints with  $y^{m1}$  and  $y^{m2}$ ; that is, if  $b^m$  is the column coefficient of  $y^m$  in  $B_1$ , we split  $b^m$  into columns  $b^{m1}$  and  $b^{m2}$ , where  $b^m = b^{m1} + b^{m2}$ ,  $b^{m1}$  and  $b^{m2}$  are both nonzero columns of 0's and 1's, and  $b^{m1}y^{m1} + b^{m2}y^{m2}$  substitutes for  $b^m y^m$  in the set partitioning constraints. Furthermore, we add the *column-joining* constraint  $y^{m1} = y^{m2}$  to the problem. Finally, we replace  $y^m$  in all other problem constraints with  $\frac{1}{2}(y^{m1} + y^{m2})$ . If we dualize all constraints of the form  $y^{m1} = y^{m2}$ , we can retain the underlying matching structure.

We will investigate the effect of adding 3-plexes to the problem by solving a number of symmetric problems with and without 3-plexes and with no availability constraints. We will use subgradient optimization for multiplier determination. Hopefully, we will obtain feasible solutions directly, with convergence of the Lagrangian dual. In the absence of convergence after a reasonable number of iterations, we will retain all 3-plexes for which the column-joining constraint is satisfied, and solve the remaining problem using only orders one and two route complexes. It is our hope that the incremental improvement afforded by 3-plexes will be small. This will give

us the opportunity to much more easily create good solutions, since we may then include only the two lowest order types of route complexes in our formulation. In such an event, only the placement (or end-node) and the aircraft availability constraints would remain. Moreover, we could save a significant amount of computer workspace by not having to store 3-plexes.

Although we have discussed dualizing complicating constraints only by individual type, any real problem is likely to contain more than one (or all) of the three types of constraints just discussed. Thus, we must be prepared to devise a strategy that can transcend these type boundaries. In the next chapter, we design a solution approach for combining the aircraft availability and placement/end-node constraints. While subgradient optimization is naturally suited to addressing all complicating constraints simultaneously, such is not the case with cross decomposition, which is tailored for mixed integer constraints. We will investigate and compare strategies based on these two techniques.

## Chapter 5

# COMPUTATIONAL RESULTS, CONCLUSIONS, AND SUGGESTIONS FOR FUTURE RESEARCH

In this chapter we detail our computational tests and report on the results. In our development to this point, the minimum weight nonbipartite perfect matching algorithm has evolved as the principal system design tool, applicable to every facet of the single-hub, single-turn problem that we discussed in the last chapter. We will organize our test design around this fact, at each step treating SHP as a matching problem with side constraints. We have identified three distinct types of complicating constraints. They are

- a. Aircraft availability
- b. Column-joining
- c. Placement and end-node

We performed our computational tests using a PL/1 program on an IBM 3090R processor under CMS at Federal Express Corporation. For the matching subproblems we used a FORTRAN 77 routine written by Professor Ulrich Derigs of the University of Bayreuth, Bayreuth, West Germany. His code runs in  $O(|V|^3)$  time, where  $|V|$  is the number of vertices in the graph. (See Ball and Derigs, [B2].) When using cross decomposition we

solved the restricted Lagrangian dual using a revised simplex code written by Professor James Ho of the University of Tennessee, Knoxville. All data was obtained from the Federal Express Corporation. For many of the larger test problems, run times were slow due to multiple calls to an inefficient transportation algorithm, which we coded in PL/1. An efficient transportation algorithm should produce much shorter run times for these problems.

We now review our computational results, drawing conclusions based on the empirical evidence and suggesting avenues for further research. We first discuss the dualization of the aircraft availability constraints using subgradient optimization.

Tables 5-1 through 5-4 contain our results for varying degrees of constraining aircraft numbers for each of four systems. The systems contain 14, 27, 41, and 81 nodes, respectively. Each system is symmetric and is allowed four aircraft types. We used the subgradient formula  $u_{\alpha}^{k+1} = \max \left( u_{\alpha}^k + t_k \left( \sum_m N_{\alpha}^m y^{mk} - q_{\alpha} \right), 0 \right)$ , where  $u_{\alpha}^k$  is the  $\alpha^{\text{th}}$  component for the  $k^{\text{th}}$  subgradient,  $N_{\alpha}^m$  is the number (quantity) of aircraft type  $\alpha$  used by route complex  $m$ ,  $y^{mk}$  is the  $k^{\text{th}}$  0-1 decision variable for route complex  $m$ , and  $q_{\alpha}$  is the number of aircraft type  $\alpha$  available.

The scalar  $t_k$  is calculated from the formula

$$t_k = \frac{\lambda_k (z^* - z(u^k))}{\left\| \sum_{\alpha} \left( \sum_m N_{\alpha}^m y^{mk} - q_{\alpha} \right) \right\|^2},$$

where  $z^*$  is an estimate of the optimal value, and  $z(u^k)$  is the optimal value of the relaxed problem using the subgradient  $u^k$ . The scalar  $\lambda_k$  is reduced by a factor  $r$  if the value of  $z(u^k)$  has not improved after a set number of iterations. (See Fisher, [F1].) Through experimentation we found that the algorithm seemed to perform best when  $r$  was in the range 0.65 to 0.78, multiplying  $\lambda_k$  by  $r$  after three iterations without an improvement of at least 0.2 percent. We set  $\lambda_0$  to 2.

In each of these four tables, the column marked "1ST" contains the optimal fleet numbers for unconstrained availabilities. The other columns

# NODES:	14	14	14	14					
<u>AIRCRAFT:</u>	<u>1st</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>
B727-100	4		10		7	7	--	10	10
B727-200	1	3	3	3	3	3	--	3	3
DC10-10	8	3	3	4	4	4	--	3	3
DC10-30	2	3	3	3	3	3	--	3	3
UNCONST.:	340,468			340,468		340,468		340,468	
BEST BD.:	352,936			345,746		345,090		352,926	
FINAL FEAS.:	352,936			345,746		none		352,926	
LAMBDA DEC.:	.65			.72		.72		.72	
% OF OPT.	100			100		0		100	
ITER.:	51			23		100		126	
TIME:	:04			:03		:07		:10	
						OSCILLATION			

**Table 5-1. Constrained Fleet Case A**



record the availabilities and the best solution obtained for different constrained systems. Throughout this chapter, if no specific limit appears for an aircraft type when running aircraft-constrained cases, then that aircraft type is unconstrained in availability. We recorded the costs for the unconstrained run and, for each constrained system, the best lower bound obtained and the best feasible solution obtained. The row labeled "LAMBDA DEC" records the  $r$  value for that system, and "% of OPT" contains the value (BEST BD./FINAL FEAS.) \* 100.

Our general methodology for constraining the fleet was to first allow the system to have as many aircraft of each type as the algorithm wished to give it. Based on the resultant quantities of aircraft used, we then constrained the fleet to force different quantities to be used. For example, in the 14-node system shown in Table 5-1, the unconstrained fleet contains 4,1,8 and 2 units of aircraft types 1 through 4 respectively. In the next two runs, we constrained types 2 through 4 to be 3, 3, and 3 respectively, and 3, 4, and 3 respectively. In each case, not only was the duality gap zero, but the optimal Lagrange multipliers also produced an optimal *feasible* solution.

We then constrained the following two runs to be exactly the optimal fleet numbers of the previous two runs. In one case we achieved convergence, although  $2\frac{1}{2}$  times the number of runs was required to obtain the same fleet numbers. The other run oscillated wildly with respect to the aircraft used in successive iterations, and no feasible solution resulted after 100 iterations. We discontinued the run because of the oscillation present, even though we allowed the previous, completely constrained, run to go further. Also, we tried only one initial value of  $\lambda$ .

The runs for the other three systems, shown in Tables 5-2 through 5-4, proceeded in much the same manner, with similar results. In general, we can draw the following conclusions. The Lagrangian relaxation for the single-hub single-turn problem with respect to the aircraft availability constraints is very tight, almost always having a duality gap of zero and producing optimal feasible solutions directly from the optimal Lagrange multipliers. Subgradient optimization proved to be quite adequate for most of the problems tested, but showed some instability with respect to  $r$ -values

# NODES: 27 27 27

<u>AIRCRAFT</u>	<u>1st</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>
B727-100	12		25	25	25	25	25
B727-200	2		6		6	6	6
DC10-10	11	4	4	4	4	4	4
DC10-30	3	3	3	3	3	3	3

UNCONST.: 635,255 635,255 635,255  
BEST BD.: 685,906 685,906 685,906  
FINAL FEAS.: 685,906 685,906 685,906

LAMBDA DEC.: .65 .72 .72  
% OF OPT.: 100 100 100  
ITER.: 44 42 44  
TIME: :09 :07 :08

NOTE: FOR LAST CASE, NOT A SINGLE FEASIBLE SOLUTION WAS OBTAINED AFTER 100 ITERATIONS FOR LAMBDA DEC = .65 OR .78.

**Table 5-2. Constrained Fleet Case B**

# NODES:	41	41	41	41	41						
<u>AIRCRAFT:</u>	<u>1st</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>
B727-100	16		20		25	25	--		25	25	25
B727-200	5		6		11	11	--		11	11	11
DC10-10	10	4	4	4	3	4	--	3	3	3	3
DC10-30	2		5	3	2	3	--	2	2	2	2
UNCONST.:	621,613		621,613	621,613	621,613	621,613		621,613		621,613	
BEST BD.:	630,167		644,460	644,460	644,463	662,972		662,972		662,972	
FINAL FEAS.:	630,167		662,972	662,972	none	662,972		662,972		662,972	
LAMBDA DEC:	.72		.65	.65	.65	.72		.72		.72	
% OF OPT.:	100		97.2	97.2	--	100		100		100	
ITER.:	7		100	100	100	31		31		8	
TIME:	:05		:39	:39	:39	:14		:14		:06	

**Table 5-3. Constrained Fleet Case C**

# NODES:	81		81		81		81		81		
<u>AIRCRAFT</u>	<u>1st</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>	<u>AVAIL</u>	<u>FINAL</u>
B727-100	33		36		36	33	31	33	33	31	--
B727-200	8		15	15	12	15	15	15	14	15	--
DC10-10	11	6	6	6	6	6	6	6	6	6	--
DC10-30	4	5	5	5	5	5	5	5	5	5	--
UNCONST.:	1,164,772			1,164,772		1,164,772		1,164,772		1,164,772	
BEST BD.:	1,173,250			1,173,250		1,173,053		1,174,353		1,175,167	
FINAL FEAS.:	1,173,250			1,173,250		1,176,797		1,174,353		none	
LAMBDA DEC:	.72			.72		.72		.65		.65, .72, .78	
% OF OPT.:	100			100		99.7		100			
ITER.:	27			20		100		79		100	
TIME:	:33			:26		1:36		1:24		MUCH OSCILLATION RELATIVE TO OTHER CASES.	

**Table 5-4. Constrained Fleet Case D**

for certain problems. In a practical setting, one might have to adjust these values as well as the fleet constraint numbers to achieve optimal solutions. A possible difficulty is that some desired fleet constraints could induce too much instability to obtain a good feasible solution. However, our results do indicate that judiciously setting the critical parameters will circumvent this snag.

We now turn to a discussion of the results of dualizing the column-joining constraints for 3-plexes. All problems studied were symmetric with no availability constraints. We used subgradient optimization just as before. Route complexes of orders one, two, and three formed the columns. Thus, the problems were essentially set partitioning problems that we transformed into matching problems with side constraints. Tables 5-5 and 5-6 display the results.

The fleet compositions for an initial feasible solution and the final solution appear for each case, along with the initial feasible bound, the unconstrained solution, the best lower bound obtained, and the cost of the best solution. The number and type of 3-plexes in the best solution are recorded in the row marked "3-PLEXES". Also, the number of candidate 3-plexes, the number of initial column-joining infeasibilities, and the final number of column-joining infeasibilities appear in the following three rows, respectively.

In the first table we show the results for four different two-aircraft systems. In each case we obtained feasible solutions that were extremely close to optimal, thus showing empirically that the duality gap relative to these constraints is quite small. When optimality was achieved, a feasible solution resulted, as happened with the aircraft availability constraints. When we stopped a run short of optimality, we kept all 3-plexes whose column-joining constraints had both been chosen and resolved the remainder of the system using only 1-plexes and 2-plexes. As can be seen, this produced excellent feasible solutions, the largest duality gap being no more than 0.5 percent.

It is of special interest to note that our initial feasible bound, which we obtained by solving the system with no 3-plexes, was always reasonably

# NODES:	14	27	41	81
<u>AIRCRAFT</u>	<u>1st</u> <u>FINAL</u>	<u>1st</u> <u>FINAL</u>	<u>1st</u> <u>FINAL</u>	<u>1st</u> <u>FINAL</u>
B727-100	5 4	14 12	26 24	50 45
DC10-10	11 11	16 16	13 13	16 17
ITER.:	100	73	150	100
FEAS. BD.:	341,115	658,723	655,309	1,221,254
UNCONST.:	308,111	609,150	556,059	1,066,243
BEST BD.:	332,984	643,977	643,234	1,185,456
FINAL FEAS.:	333,288	643,977	645,188	1,191,763
3-PLEXES:	1 STAR	2 STARS	1 3-LEG, 3 STARS	3 3-LEGS, 6 STARS
MATCHING COLS.:	40	92	305	761
INITIAL INFEAS.:	6	9	20	38
FINAL INFEAS.:	1	0	3	4
FINAL % OPT.:	99.9	100.0	99.7	99.5
IMPROVEMENT OVER 1st FEAS.:	2.3	2.2	1.5	2.4
TIME:	:04	:07	:30	1:29

**Table 5-5. Inclusion of 3-Plexes, Two Aircraft Types**

# NODES:	14	27	41	81
<u>AIRCRAFT:</u>	<u>1st</u> <u>FINAL</u>	<u>1st</u> <u>FINAL</u>	<u>1st</u> <u>FINAL</u>	<u>1st</u> <u>FINAL</u>
B727-100	4 3	12 12	16 14	33 33
B727-200	1 0	2 2	5 6	8 9
DC10-10	8 10	11 8	10 10	11 10
DC10-30	2 1	3 5	2 2	4 4
ITER.:	43	25	154	285
FEAS. BD.:	340,468	635,255	621,613	1,164,763
UNCONST.:	297,365	567,152	512,079	986,399
BEST BD.:	332,655	629,168	615,822	1,155,177
FINAL FEAS.:	332,655	629,168	619,054	1,160,854
3-PLEXES:	2 STARS	3 STARS	1 3-LEG, 1 STAR	2 3-LEGS, 2 STARS
MATCHING COLS.:	49	119	338	761
INITIAL INFEAS.:	9	14	40	74
INFAL INFEAS.:	0	0	0	6
FINAL % OPT.:	100	100	100	99.5
IMPROVEMENT OVER 1st FEAS.:	2.3	0.9	0.4	0.3
TIME:	:08	:12	1:20	6:51

**Table 5-6 Inclusion of 3-Plexes, Four Aircraft Types**

close to the best possible 3-plex solution. The best improvement was in the 81-node case, where the maximum possible improvement was only 2.9 percent. Nonetheless, as Table 5-5 shows, a reduction in the total number of aircraft accompanied the inclusion of 3-plexes in the final solution, for each case. In the 81 node system, a total of nine 3-plexes were present in the final feasible solution, saving four aircraft. Thus, even though the improvement over the initial feasible solution was only 2.4 percent, this represents a daily savings of almost \$30,000.

Table 5-6 shows the results for the same four systems with four available aircraft types. As with the previous cases, we observed excellent best feasible solutions. However, the cases for 41 nodes and 81 nodes show very little improvement from 3-plexes in the solution. This small change is accompanied by almost no decrease in the total number of aircraft used. It thus appears that the larger fleet mix has some implied advantages. Because it "fits" a given system better than a smaller fleet mix, it allows a very good solution to result from a relatively simple route network. Not only is this computationally attractive, since we can potentially eliminate the burden of including 3-plexes among the candidate columns, it is also operationally attractive for the reasons that we discussed at the end of chapter 3.

To further test the validity of our results for the 81-node system, we extended our series of runs to include ten more symmetric problems, five pickup side problems, and five delivery side problems. All data for these runs was identical to the data for the original 81-node system, except that we used different cargo demands for each run. We ran each problem for 400 iterations. For the symmetric problems, the average 2-plex system was within 0.8% of the best possible 3-plex system, and the average best feasible 3-plex system *obtained* was 0.5% better than the 2-plex system. For the pickup and demand problems, the averages were 3.7% and 1.3%, and 1.6% and 1.1% respectively.

We can argue, then, that leaving 3-plexes out of the problem formulation is not only computationally beneficial, but can often result in excellent feasible solutions, although the quality of the solutions is largely data dependent. One might contend that 1.3% of a million dollars (approximately)



daily expenditure is certainly worth saving. However, these solutions were obtained only after 400 iterations at significant computational cost. As we have discussed, we can obtain very good solutions without including 3-plexes. An alternative to applying an optimization-based approach for including 3-plexes might be to optimize over all lower-order route complexes and then heuristically swap in any 3-plexes that improve the solution. We might in this way capture some of the possible incremental improvement at a low computational cost.

It is intuitive that an increasingly varied fleet mix results in an increasingly simple optimal system. In the extreme case we have an aircraft tailored to each node, with only single-aircraft 1-plexes in the optimal route network. However, this notion is essentially counterintuitive to the expectation that increasing the number of aircraft types increases the combinatorial difficulty of obtaining an excellent feasible solution.

For the remainder of this chapter we discuss the results for asymmetric solutions. As we have indicated, we tested two methods for tying the delivery and pickup sides of a system together. We first discuss the placement constraints. In Table 5-7 we show the results for runs on four different systems for one, two, and four aircraft types in the fleet. Each aircraft type is unconstrained in availability in all cases.

In the two "FIRST" columns we recorded the costs of the first feasible solution and the first relaxation, which is just the cost of the first feasible solution minus the placement flight costs. The first "%" column is a measure of how close these two numbers are. For the largest system of 81 nodes this percentage grows larger as the number of aircraft types grows smaller. This would be expected for any large system, principally for the same reason that 3-plexes offer a minimal improvement in the presence of many aircraft types. That is, with more aircraft types, a better "fit" is possible for each node; thus, a load imbalance, consisting of a large difference between the supply and the demand at a node, will often result in different aircraft types being assigned to the supply and demand sides for the same node, forcing placement flights in the network.

# NODES / # AC TYPES	ITER.	TIME	FIRST FEAS.	FIRST BD.	%	BEST FEAS.	BEST BD.	%	TRANS\$
14,1	50	:25	434,967	404,655	93.0	432,830	419,090	96.8	22702
27,1	30	:45	840,404	776,597	92.4	833,216	805,041	96.6	45876
41,1	75	4:25	744,446	699,062	93.9	725,401	711,209	98.0	20515
81,1	25	8:15	1,346,136	1,272,018	93.2	1,336,518	1,301,325	97.4	49485
14,2	75	:37	427,844	315,946	73.8	356,928	326,580	91.5	32148
27,2	75	1:34	714,308	614,850	86.1	648,004	626,819	96.7	22215
41,2	150	6:44	730,615	582,325	79.7	609,927*	595,768	97.7	12785
81,2	75	14:30	1,276,495	1,096,247	85.9	1,170,653*	1,115,859	95.3	61288
14,4	100	1:10	398,179	305,497	76.7	350,598	313,700	89.5	22384
27,4	75	1:50	733,842	587,035	80.0	661,824	598,750	90.5	58034
41,4	135	8:34	673,742	563,961	83.7	609,675*	575,903	94.5	13053
81,4	45	11:13	1,285,086	1,060,722	82.5	1,165,498	1,071,243	91.9	78788

**Table 5-7 Asymmetric Solutions, Unconstrained Fleets**

The two "BEST" columns contain the costs of the best feasible solution and the best Lagrangian lower bound obtained during the run. The second "%" column measures the closeness of these two numbers, and the "TRAN\$" column is the placement flight costs for the best feasible solution.

The dual improvement heuristic did very little to improve the optimal transportation subproblem dual solutions, and the full values of the dual variables caused very large fluctuations in the aircraft numbers for resultant successive feasible solutions. We can see how this happens by considering formulation (SC) in chapter 4. Solving for  $w_{im,jk}^1$  in this formulation, we obtain an objective function that is piecewise linear in the only decision variable,  $u_{im}^1$ . That is, we wish to maximize  $mu_{im}^1 - \sum_j \sum_k \max[0, \hat{v}_{jk}^1 + ku_{im}^1 - \frac{k}{2}c_{ij}]$ , where  $\hat{v}_{jk}^1$  is fixed. The slope of this function for any segment is  $\Delta = m - \sum_{(j,k) \in (J,K)^+} k$ , where  $(J \times K)^+ = \{(j, k) : \hat{v}_{jk}^1 + ku_{im}^1 - \frac{k}{2}c_{ij} > 0\}$ . Thus, if  $\Delta > 0$  we increase  $u_{im}^1$  until  $\Delta$  becomes nonpositive, and if  $\Delta < 0$  we decrease  $u_{im}^1$  until  $\Delta$  becomes nonnegative.

Because of our initial assignment of values for  $v_{jk}^1$  and  $u_{im}^2$ , we start with  $(J \times K)^+ = \{\}$ . Thus, we increase  $u_{im}^1$  until  $\Delta$  becomes nonpositive. For any aircraft type,  $c_{ii} = 0$ , since this is the cost of simply remaining on the ground. Our transportation algorithm, which solved  $(TP\hat{y}\alpha)$ , always returned optimal values for  $v_i$  and  $u_i$  such that the constraint  $v_i + u_i \leq c_{ii}$  was tight for all  $i$ , and such that both  $v_i$  and  $u_i$  were fairly large in absolute value (several hundred up to several thousand).

The effect of constraint tightness for all  $c_{ii}$  is that whenever  $\hat{v}_{ik}^1 + ku_{im}^1 - \frac{k}{2}c_{ii}$  becomes positive for one  $k \in M_j^1$  as  $u_{im}^1$  increases, it becomes positive for all  $k \in M_j^1$ . Thus,  $\Delta$  becomes negative as soon as  $u_{im}^1$  undergoes any increase whatsoever, in most cases. This prevents any meaningful improvement of the initial optimal dual solution. The effect of the large absolute values of  $v_i$  and  $u_i$  is that, usually one side of (i.e., delivery or pickup) of a solution has very positive optimal dual variables, and the other side has very negative optimal dual variables. Therefore, the succeeding run with these (mostly unimproved) dual values as Lagrange multipliers results in very different fleets for the pickup and delivery sides of the problem, when multiple air-

craft types are used, and different end-nodes for the pickup and delivery sides of the problem. These solutions have very large placement flight costs and are expensive overall. The next solution using *these* dual variables will then tend to oscillate back in the other direction with respect to fleet composition and route end-nodes.

Experimenting with scaling the dual variables down, we found that using the following formula for generating the  $k + 1^{\text{st}}$  Lagrange multipliers seemed to work well. We set

$$v_{\alpha pi}^{k+1} = (1 - s) v_{\alpha pi}^k + s d_{\alpha pi}^k,$$

where  $v_{\alpha pi}^{k+1}$  is the  $k + 1^{\text{st}}$  Lagrange multiplier for node  $i$ , aircraft type  $\alpha$ , and side  $p$  (i.e.. pickup (demand) or delivery (supply)),  $d_{\alpha pi}^k$  is the appropriate dual variable in the corresponding transportation problem, and  $s$  is the scaling vector. We initialized with  $v_{\alpha pi}^0 = 0$ .

For one aircraft type we found  $s = .5$  to be a good scaling factor, and for the other fleet mixes we found  $s = 0.015$  to work well. There was still a certain amount of fluctuation, but we were able to take advantage of this by observing that the pickup side of a given solution would cause the generated Lagrange multipliers to adjust the *following* solution's delivery side accordingly. Thus, combining the pickup side of solution  $k$  with the delivery side of solution  $k + 1$  would often result in superior solutions. The same held true for the delivery side of solution  $k$  and the pickup side of solution  $k + 1$ . Therefore, we checked these pairings at every iteration. An asterisk appears by the cost of any best feasible solution that was produced in this way. The same convention is used in the remainder of the tables for this chapter.

In solving the restricted Lagrangian master problem after cycling occurred, infeasible Lagrangian multipliers often resulted. That is, we obtained values of  $u$  for which  $uA \leq c$  was not true. We corrected this situation by adding the appropriate constraints to the restricted master problem. Thus, if  $\hat{u}_i$  and  $\hat{v}_j$  were such that  $\hat{u}_i + \hat{v}_j > c_{ij}$ , we added the constraint  $u_i + v_j \leq c_{ij}$  to the master problem and then resolved it.

Table 5-7 shows that, although the proven bounds are all (with one exception) within 10% of optimum, our results for dualizing these constraints

using cross decomposition are not as tight as for the other two constraint types. Moreover, the placement flight costs are high in many cases. For example, the cost of \$78,788 for the four-aircraft-type, 81-node system is a substantial daily expenditure for a company. This amount could represent about 10 to 15 placement flights, and this number of zero-cargo flights could be quite difficult to sell to management or operations personnel such as pilots. Nonetheless, if the numbers represent good solutions, management should be aware of this fact.

Table 5-8 displays the results for some asymmetric systems with constrained fleets. The best fleet composition for each case is shown, and any limits on aircraft types are shown in parentheses. No number in parentheses indicates no limit. The "GAP" numbers represent how close the first feasible solution is to the initial bound and how close the best feasible solution is to the best bound, respectively. To obtain solutions, we ran each side (pickup and delivery) of the problem separately using Lagrangian relaxation until an aircraft-feasible solution occurred, and then allowed five more iterations to improve on this unless the feasible solution occurred on the first iteration. We formed the bounds from the best solution values on each side. Where two or more aircraft types were constrained, we used subgradient optimization to achieve feasible aircraft numbers. When only one type was constrained, we established an interval  $[0, \hat{\lambda}]$  for the single availability constraint such that  $\lambda^0 = \hat{\lambda}$  would yield a feasible solution, and used a simple binary search to find a good value for  $\lambda^*$ . We allowed four aircraft types in all cases.

As might be expected, the final solution values for two constrained aircraft types are not as low as for the unconstrained cases, except for the 27-node case. The 27-node problem proved to be very difficult to obtain a good asymmetric solution for, in almost all instances, because of large load imbalances. Apparently constraining the fleet as we did directed the algorithm toward a better portion of the feasible region.

The anomalous behavior of the 27-node case was repeated for *all* systems examined when one aircraft type was constrained. That is, for every case, the lowest feasible costs occurred when a limit was put on one aircraft

AIRCRAFT #

B727-100:	30	(32)	13	(15)	10	(10)	2	(6)	31	18	10	6	
B727-200:	9		7		3		3		8	4	3	1	
DC10-10:	5	(6)	3	(6)	6	(6)	1	(3)	6	5	6	3	(3)
DC10-30:	7		6		6		7		6	4	6	4	
1st GAP, 2nd GAP:	80.0, 91.0	83.6, 92.3	83.3, 92.2	78.6, 86.1	85.3, 92.9	80.6, 95.1	80.1, 92.4	83.5, 93.0					
FIRST BD.:	1,052,139	564,631	589,541	306,246	1,061,575	564,264	588,018	309,727					
FIRST FEAS.:	1,314,082	675,678	707,458	389,531	1,245,086	700,413	733,924	371,015					
FINAL BD.:	1,070,368	572,018	598,985	313,449	1,075,906	575,861	600,032	314,925					
FINAL FEAS.:	1,176,116*	620,065	649,971*	363,937	1,158,309*	605,779*	649,329*	338,697					
TRANS \$	102465	19999	38556	15323	88285	5703	39365	8542					
# NODES:	81	41	27	14	81	41	27	14					
TIME:	14:10	4:32	2:46	1:22	17:43	8:29	3:46	:42					
ITER:	37	50	40	100	40	100	100	50					

only. The bounds obtained were generally good, with all four cases being within 7.6% of optimum. However, the lower feasible costs indicate that the placement dual variables alone are not enough to properly drive the algorithm toward the best feasible solution. In a practical setting, some kind of additional search based on restricting the aircraft quantities may be necessary if placement dual variables are used. The apparent fact that the placement dual variables by themselves do not direct feasible solutions across the boundaries of aircraft types is evidenced by the results of Table 5-7, where we see that the best feasible solutions obtained have sharper lower bounds for fewer types in the fleet.

In Tables 5-9 through 5-12 we present the results of asymmetric runs based on the end-node constraints. Because these constraints do not truly model the problem, in that placement flights were not modeled, the resultant values of the Lagrangian do not represent true lower bounds on the optimum value. For these bounds we use the values from Tables 5-7 and 5-8 where appropriate. We include the best Lagrangian bounds from the end-node constraints to show how tight this particular relaxation is.

In all cases we used subgradient optimization on the end-node constraints. Tables 5-9 and 5-10 show the results for systems with two aircraft types and four aircraft types, respectively. In each table we used two different lambda-decrementing  $r$ -values, 0.5 and 0.85. When  $r = 0.5$  we allowed seven iterations for improvement and when  $r = 0.85$  we allowed four iterations for improvement. We defined "improvement" to be an increase of 0.2 percent in the Lagrangian.

For both values of  $r$  the resultant best feasible solutions showed significant improvement over cross decomposition with the placement constraints. The algorithm actually converged (relative to the end-node constraints) for a few of the cases. The results were somewhat better overall for  $r = 0.85$ , but the general tightness of the bounds for both  $r$ -values indicates robustness relative to this parameter. Not only were the best feasible solutions better than in Table 5-7 for these cases, but the costs for placement flights were far lower as well. As we mentioned earlier, this could make the solutions much easier to sell to management and operations personnel. We did

<u>#</u>	<u>#AC</u>	<u>ITER.</u>	<u>TIME</u>	<u>LAMBDA</u>	<u>BEST</u>	<u>BEST BD.</u>	<u>%</u>	<u>BEST BD.</u>	<u>%</u>	<u>TRANS\$</u>
<u>NODES</u>	<u>TYPES</u>			<u>DEC.</u>	<u>FEAS.</u>	<u>(END-NODE)</u>		<u>(PLACEMENT)</u>		
14	2	41	:16	0.5	332,704	332,704	100	326,580	98.2	0
27	2	100	1:18	0.5	645,570	642,012	99.4	626,819	97.1	8,289
41	2	100	4:01	0.5	611,282	603,161	98.7	595,768	97.5	5,185
81	2	75	20:54	0.5	1,170,759	1,118,868	95.6	1,115,859	95.3	29,791
14	2	37	:14	0.85	332,794	332,704	100	326,580	98.2	0
27	2	100	1:14	0.85	645,570	641,691	99.4	626,819	97.1	8,289
41	2	90	3:23	0.85	606,032*	603,593	99.6	595,768	98.1	5,486
81	2	83	18:22	0.85	1,142,279	1,134,435	99.3	1,115,859	97.7	9,156

TABLE 5-9 ASYMMETRIC SOLUTIONS (END-NODE CONSTRAINTS)



<u>#</u> <u>NODES</u>	<u>#AC</u> <u>TYPES</u>	<u>ITER.</u> <u>TIME</u>	<u>LAMBDA</u> <u>DEC.</u>	<u>BEST</u> <u>FEAS.</u>	<u>BEST BD.</u> <u>(END-NODE)</u>	<u>%</u>	<u>BEST BD.</u> <u>(PLACEMENT)</u>	<u>%</u>	<u>TRANS\$</u>
14	4	50 :46	0.5	326,774	323,047	98.9	313,700	96.0	7,829
27	4	110 1:23	0.5	627,733	615,580	98.1	598,750	95.4	16,127
41	4	75 13:37	0.5	588,620	583,815	99.2	575,903	97.8	4,966
81	4	82 18:22	0.5	1,104,397*	1,095,063	99.2	1,071,243	97.0	16,623
14	4	75 :45	0.85	326,774	322,958	98.8	313,700	96.0	7,829
27	4	100 2:08	0.85	627,733	616,504	98.2	598,750	95.4	16,127
41	4	72 14:15	0.85	583,888	583,888	100	575,903	98.6	0
81	4	141 35:40	0.85	1,109,729	1,096,300	98.8	1,071,243	96.5	12,801

TABLE 5-10 ASYMMETRIC SOLUTIONS (END-NODE CONSTRAINTS)

not complete runs for one-aircraft type systems using the end-node constraints because none of these had feasible solutions with zero placement flights. The Lagrangians were thus highly unstable.

Tables 5-11 and 5-12 display the results for constrained fleets. The format for fleet composition and constrained availabilities is the same as in Table 5-8. We initially attempted to handle the availability constraints as we did when using cross decomposition, but this proved to be inferior to simply dualizing both the availability and end-node constraints simultaneously. We used an  $r$ -value of 0.85, applied every four iterations with no improvement, defined as we did earlier. These constrained runs were rather sensitive to these values. As with the unconstrained availability runs, we obtained generally excellent bounds. Interestingly, we encountered some of the same anomalous behavior for one constrained aircraft type as we previously described for the placement constraints and one constrained type. Thus, it appears that, although the end-node constraint approach produced much better feasible solutions than the placement constraint approach, the Lagrange multipliers produced from these constraints alone were not sufficient to direct the algorithm to the best feasible solutions in all cases. Thus, some sort of search based on constraining aircraft availabilities might be called for. Nonetheless, the difference in the 81-node system, about 0.9 percent for an  $r$ -value of 0.85, could be quite tolerable without any type of search, especially in long-range planning exercises.

Table 5-12 shows the results for runs of two or more constrained aircraft types. These also are significantly superior to their counterparts using placement constraints. An interesting pair of runs is the 81-node system constrained 32,  $\infty$ , 6, 5, and  $\infty, \infty$ , 6, 5 respectively. Though the best solution in each case contained 32, 10, 6, and 5 units respectively, the latter run was clearly better. Thus, although Table 5-11 shows that some constraining can benefit the search for a good feasible solution, this comparison shows that too much constraining, even with the "correct" limits, can be detrimental, given the same lambda-decrementing scheme. Likewise, the 81-node system constrained 32  $\infty$ , 6,  $\infty$  did not fare quite as well as the run constrained  $\infty, \infty$ , 6,  $\infty$  (Tables 5-12 and 5-11 respectively), even

<u>AIRCRAFT #</u>				
B727-100:	9	8 (10)	17	30
B727-200:	0	3	4	9
DC10-10:	3 (3)	7	6 (6)	6 (6)
DC10-30:	4	6	4	6
BEST BD.:	325,215	615,052	584,744	1,096,898
BEST FEAS.:	329,494	619,979	587,587	1,099,010*
GAPS (%):	98.7, 95.6	99.2	99.5, 98.0	99.8, 97.9
TRANS\$:	7,829	11,646	3,310	6,492
# NODES:	14	27	41	81
TIME:	:41	3:01	14:03	44:08
ITER:	54	110	75	140

**Table 5-11. Asymmetric Solutions One Constrained Aircraft Type**

<u>AIRCRAFT #</u>						
B727-100:	5 (6)	9 (10)	31 (32)	14 (15)	32	32 (32)
B727-200	3	2	8	6	10	10
DC10-10	3 (3)	6 (6)	6 (6)	6 (6)	6 (6)	6 (6)
DC10-30	4	7	6	4	5 (5)	5 (5)
BEST BD.:	326,796	614,002	1,095,602	583,858	1,099,571	1,097,295
BEST FEAS.:	339,425	643,279	1,101,941	588,611	1,119,612	1,129,891
GAPS (%):	96.3, 92.3	95.4, 93.1	99.4, 97.1	99.2, 97.2	98.2	97.1
TRANS\$:	13,181	32,538	11,484	3,154	19,815	32,582
# NODES:	14	27	81	41	81	81
TIME:	1:20	4:10	32:15	8:20	36:27	43:15
ITER:	100	125	120	100	155	175

**Table 5-12. Asymmetric Solutions, Constrained Fleets**

though the latter run produced a result feasible with respect to the former availabilities. Still, the difference here is only about 0.3 percent, quite likely acceptable for many applications.

## 5.1 Conclusions and Suggestions for Further Research

We conclude overall that the route complex approach is a viable method of solution for the single-hub, single-turn problem SHP. Not only does it allow us to exploit the efficiently solvable structure of the problem – in the form of the minimum weight nonbipartite perfect matching problem – it also allows us to easily screen any operationally undesirable routes from consideration. The latter point is especially important in view of the fact that many operational constraints could be quite cumbersome to formulate mathematically, but simple to enforce programmatically.

The aircraft availability constraints and the end-node constraints responded well to dualization using subgradient optimization, both separately and together. Our results using cross decomposition on the placement constraints were inferior to subgradient optimization on the end-node constraints. It is possible that applying subgradient optimization to the placement constraints would produce better results. However, this particular type of constraint may not have as tight a relaxation as the end-node constraints.

The oscillation that occurred when we used the unscaled dual variables from the transportation subproblem solutions as Lagrange multipliers may well indicate the presence of a nonzero duality gap. Consider any optimal Lagrange multiplier,  $u^*$ . If there is no duality gap, we know from Theorem C.3 (Appendix C) that there exists an optimal solution  $(x^*, y^*)$  to the Lagrangian relaxation for  $u^*$  that is feasible in the unrelaxed problem. Further, Theorem 4.2 shows that  $u^*$  must be a dual optimal solution to the transportation subproblems associated with  $(x^*, y^*)$ . However, if all unscaled dual optimal solutions result in a significant oscillation, greatly differing end-nodes and fleets for the pickup and delivery sides result. The

cost of the transportation subproblems would then be very high, given our cost structure (outlined in Chapter 1) and the fact that many placement flights would have to occur. It is very unlikely that such a solution, even if feasible, is optimal. If the solution is *not* optimal, then it can only mean that the duality gap is positive.

We wish to emphasize that we tried but one heuristic to improve the optimal transportation subproblem dual variables. It could well be that some other technique would improve these dual variables and thereby avoid the oscillation. If this is the case, the duality gap could indeed be zero. Another possibility is that a different formulation of the placement constraints could produce a zero duality gap. (One conclusion is almost certainly true, nevertheless – formulation  $(TP\hat{y}\alpha)$  of chapter 4 has a positive duality gap, since *any* dual optimal solution is likely to cause oscillation from one relaxation to the next. This is because of constraint tightness for  $v_i + u_i \leq c_{ii} = 0$  and large absolute values for  $v_i$  and  $u_i$ , which we discussed earlier in this chapter.) Also, we should not rule out the possibility that the placement constraints, dualized with subgradient optimization or another technique, could produce quite excellent feasible solutions. Moreover, as we have noted, it is the placement constraints' dualization that provides a lower bound for our solutions. Thus, even if we abandon these constraints as a means of finding feasible solutions, we may wish to retain them to benchmark whatever alternative we choose.

Dualizing the column-joining constraints via subgradient optimization was successful, first because the duality gap was extremely small, and second because we found that in many cases the incremental improvement offered by 3-plexes in a solution is very low. Thus, if there is a good mix of aircraft in a fleet and the time constraints are tight enough to hold route lengths down, 1-plexes and 2-plexes are enough to provide a very good solution. This would be especially true if the demands at the individual nodes were high enough to prevent aircraft from taking on more than two nodes' worth of cargo, so that route lengths were held down by this factor as well. The particulars of the problem at hand are very important in making such determinations, and must be considered or even investigated

in detail before rendering these judgments.

We found the aircraft availability constraints to be sensitive in some instances to the levels of availability and the  $r$ -values chosen, even when a known optimal feasible solution existed. This sensitivity carried over when asymmetric problems were availability-constrained; we found that a much smaller range of  $r$ -values would produce good solutions for the end-node constraints *with* the availability constraints than for the end-node constraints alone. Thus, some care should be exercised when constraining a fleet – in general, if the algorithm automatically holds an aircraft’s usage to an acceptable level, then we should probably not constrain that type. Again, however, the individual problem must be considered. It is possible (and we have seen examples to support this) that *some* constraining of aircraft types actually improves the feasible solution obtained or the speed with which a solution is obtained. Further study into this property could prove enlightening.

As we stated early on, our intention throughout our employment of Lagrangian relaxation has been to obtain feasible solutions directly, without using branch-and-bound. This departure from what has usually been the ultimate intent of Lagrangian relaxation – to provide bounds for using in a branch-and-bound – has proven largely successful. Our effort was made possible largely by the fact that most of the problem’s structure remained after we dualized the complicating constraints.

We have taken the approach of designing an optimization-based algorithm to solve a subset, albeit a significant one, of the original problem. We have shown empirically by expanding the subset problem to include larger route complexes (i.e., 3-plexes) that the subset approach appears to be justified. Also, operational considerations such as those discussed at the end of Chapter 3 further lead us to accept such a strategy.

The richness and newness of the problems that we have discussed and addressed suggest a myriad of research opportunities. Within the confines of the single-hub single-turn problem SHP, several areas are open. One is the area of route complexes, their generation and storage, and the profitability of higher order complexes. In our application, for example, it was

impossible to explicitly store all 3-plexes. We were forced to store a list of triples for which some 3-plex existed, and regenerate the route complexes whenever we wished to scan all of them. It could be productive to find out beforehand if some route complexes would never be needed, or to develop an efficient means of constructing them dynamically, as in column or facet generation. With respect to the profitability of higher order route complexes, it would be beneficial to have some sort of an estimate of the incremental improvement these  $n$ -plexes represent. A probability study in this area might be quite useful.

It could also prove fruitful to investigate other avenues of route generation. Although the route complex approach worked well for our application, its success was heavily dependent on factors that we mentioned earlier – time constraints, demand distributions, etc. If, for example, SHP had much looser time constraints and much smaller demands at the nodes, a completely different strategy could be appropriate.

Another area of SHP that would be useful to research is alternatives for formulating and handling the placement constraints. As we have mentioned, applying subgradient optimization is one obvious possibility. Also, dealing with multiple constraint types is an eventuality for many real problems, and we did not address any combination involving the column-joining constraints in this thesis. Although we showed that 3-plexes and higher order route complexes were probably often not needed for excellent solutions, such may not always be the case. Also, it is quite possible that other types of constraints than those presented in this thesis could arise. One example is that a carrier may wish to force a certain number of some aircraft type into a solution. This type of constraint would be identical to the aircraft availability constraint, except that the relation would be equality or greater-than-or-equal-to instead of less-than-or-equal-to.

In addition to further research within the single-hub single-turn problem area, little research has been done on any of the other problems that we have described, as far as we are aware. Thus, each of the areas of regional hubs, bleeder systems, trunk hubs, feeder systems, double-turn systems, and all variations on these systems comprise totally new research opportunities. In



**general, express system design is not only an untouched research prospect,  
but one that is addressable with current techniques.**

# APPENDIX A

To illustrate the method of Lagrangian relaxation, we shall apply it to a formulation of the Traveling Salesman Problem (TSP). This formulation is due to Gavish and Graves [G2]. We wish to

$$\text{minimize } z = \sum_{i=1}^n \sum_{j=1}^n c_{ij} y_{ij} \quad (\text{i})$$

subject to

$$\sum_{i=1}^n y_{ij} = 1 \quad j = 1, \dots, n \quad (\text{ii})$$

$$\sum_{j=1}^n y_{ij} = 1 \quad i = 1, \dots, n \quad (\text{iii})$$

$$\sum_{\substack{j=1 \\ j \neq i}}^n f_{ji} - \sum_{\substack{j=2 \\ j \neq i}}^n f_{ij} = 1 \quad i = 2, \dots, n \quad (\text{iv})$$

$$f_{ij} \leq (n-1)y_{ij} \quad i = 1, \dots, n \quad (\text{v})$$

$$y_{ij} = 0 \text{ or } 1, f_{ij} \geq 0 \quad j = 2, \dots, n, i \neq j$$

The idea of Lagrangian relaxation is to express one or more of the problem constraints as part of the objective function. This is called *dualization* of the constraints. Ideally, dualization removes constraints that complicate the problem (*complicating constraints*), leaving behind a problem that is relatively easy to solve, and for which the optimal solution is close to the optimal solution for the original problem.

For TSP, let us dualize the constraints (v). To do this, we proceed as follows. First, we attach *Lagrange multipliers*  $u = (u_{ij})$  to the constraints

and express them as part of (i). The problem then becomes (TSP')

$$\text{minimize } Z(\mathbf{u}) = \sum_{i=1}^n \sum_{j=1}^n c_{ij} y_{ij} + \sum_{i=1}^n \sum_{j=1}^n u_{ij} [f_{ij} - (n-1)y_{ij}] \quad (\text{i})$$

subject to

$$\sum_{i=1}^n y_{ij} = 1 \quad j = 1, \dots, n \quad (\text{ii})$$

$$\sum_{j=1}^n y_{ij} = 1 \quad i = 1, \dots, n \quad (\text{iii})$$

$$\sum_{\substack{j=1 \\ j \neq i}}^n f_{ij} - \sum_{\substack{j=2 \\ j \neq i}}^n f_{ij} = 1 \quad i = 2, \dots, n \quad (\text{iv})$$

$$y_{ij} = 0 \text{ or } 1, f_{ij} \geq 0$$

We require that  $u_{ij} \geq 0$  for all  $i$  and  $j$ . We rewrite  $Z(\mathbf{u})$  as

$$Z(\mathbf{u}) = \sum_{i=1}^n \sum_{j=1}^n \hat{c}_{ij} y_{ij} + \sum_{i=1}^n \sum_{j=1}^n u_{ij} f_{ij}$$

where  $\hat{c}_{ij} = c_{ij} - (n-1)u_{ij}$ . The resulting problem TSP' immediately decomposes into two easily solved problems. The first is a matching problem over the variables  $y_{ij}$  with cost coefficients  $\hat{c}_{ij}$ , and the second is a minimum cost-flow problem with flow variables  $f_{ij}$  and per unit arc flow costs  $u_{ij}$ .

If  $Z^*$  is the optimal value for TSP, and  $Z^*(\mathbf{u})$  is the optimal value for TSP' with Lagrange multipliers  $\mathbf{u}$ , then  $Z^* \geq Z^*(\mathbf{u})$ . This is because if  $(\hat{f}, \hat{y}) = [(\hat{f}_{ij}), (\hat{y}_{ij})]$  is feasible for TSP, then it is obviously feasible for TSP'. Moreover, the conditions  $u_{ij} \geq 0$  and  $\hat{f}_{ij} \leq (n-1)\hat{y}_{ij}$  imply that

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} \hat{y}_{ij} \geq \sum_{i=1}^n \sum_{j=1}^n c_{ij} \hat{y}_{ij} + \sum_{i=1}^n \sum_{j=1}^n u_{ij} [\hat{f}_{ij} - (n-1)\hat{y}_{ij}].$$

Thus,  $Z^* \geq Z^*(\mathbf{u})$ . Indeed, this is true for a Lagrangian relaxation of any mathematical program with suitable restrictions on the Lagrange multipliers. (In this case,  $u_{ij} \geq 0$  suffices. See Fisher [F1] for further details.)

We would like to choose the Lagrange multipliers so that we obtain the largest possible value of  $Z^*(\mathbf{u})$ , in light of the above inequality. That is, we wish to

$$\max_{u \geq 0} Z^*(u). \quad (D)$$

The problem D is referred to as the *Lagrangian dual*. If we denote the solution value to D as  $Z_D$ , we have that  $Z^* \geq Z_D$ , since  $Z^* \geq Z^*(u)$  for any  $u$ . In general,  $Z^* > Z_D$ , and the difference is called the *duality gap*.

It is possible that dualizing a different set of constraints can reduce a duality gap. For example, Gavish and Graves [G2] have dualized all of constraints (iii) except for the case  $i = 1$ . This results in a subproblem which is equivalent to finding a minimum cost 1-arborescence. This is a directed spanning tree with a root at node 1, in addition to a single arc directed into node 1. Golden and Magnanti [G6] also discuss using a 1-arborescence in this context. There is a good algorithm for this problem, and thus, this competing relaxation might be better for our purposes than TSP'.

Subgradient optimization has been widely used for determining an optimal solution to the Lagrangian dual, and its convergence is guaranteed under suitable conditions. The method is based on the result that for any continuous, concave function  $g$ , there is at every point a vector  $\bar{\gamma}$  such that  $g(x) \leq g(\bar{x}) + (x - \bar{x})\bar{\gamma}$ . The vector  $\bar{\gamma}$  is called the *subgradient* of  $g$  at  $\bar{x}$ , and is a generalization of the gradient. It turns out that if, in TSP',  $(\hat{f}, \hat{y}) = [(\hat{f}_{ij}), (\hat{y}_{ij})]$  is optimal for  $\hat{u}$ , then  $\hat{\gamma} = \hat{f} - (n - 1)\hat{y}$  is a subgradient of  $Z^*(u)$  at  $\hat{u}$ . This is so because, if  $(\hat{f}, \hat{y})$  is optimal at  $\hat{u}$ , then  $Z^*(\hat{u}) + (u - \hat{u})\hat{\gamma} = C\hat{y} + \hat{u}\hat{\gamma} + u\hat{\gamma} - \hat{u}\hat{\gamma}$ . However, for any other  $u$ ,  $(\hat{f}, \hat{y})$  may not be optimal. Thus, if  $(f, y)$  is optimal for  $u$ , it follows that

$$\begin{aligned} Z^*(u) &= Cy + u(f - (n - 1)y) \leq C\hat{y} + u(\hat{f} - (n - 1)\hat{y}) \\ &= C\hat{y} + u\hat{\gamma} = Z^*(\hat{u}) + (u - \hat{u})\hat{\gamma}. \end{aligned}$$

Moreover,  $Z^*(u)$  is a continuous, concave function of  $u$ . This can be seen by considering that if  $(\bar{f}, \bar{y})$  is optimal for  $\bar{u} = \alpha u_1 + (1 - \alpha)u_2$ , where  $0 < \alpha < 1$ , it follows that

$$Z^*[\alpha u_1 + (1 - \alpha)u_2]$$

$$\begin{aligned}
&= C\bar{y} + [\alpha u_1 + (1 - \alpha)u_2] [\bar{f} - (n - 1)\bar{y}] \\
&= \alpha C u_1 + \alpha u_1 [\bar{f} - (n - 1)\bar{y}] + (1 - \alpha) C u_2 + (1 - \alpha) u_2 [\bar{f} - (n - 1)\bar{y}] \\
&\geq \alpha Z^*(u_1) + (1 - \alpha) Z^*(u_2).
\end{aligned}$$

Because  $Z^*(u)$  may not be differentiable everywhere, methods of ascent requiring smoothness are not suitable. However, we may use the subgradient to our advantage in much the same manner that the gradient is used to optimize everywhere-differentiable functions. The algorithm generates successive solutions according to the rule

$$x^{i+1} = x^i + \theta^i \gamma^i, \quad i = 0, \dots$$

The term  $\theta^i$  is quite often defined by

$$\theta^i = \frac{\lambda_i [Z_D - Z^*(u_i)]}{\|\gamma^i\|^2},$$

where  $\gamma^i$  is any subgradient of  $g$  at  $x^i$ . Ideally,  $Z_D$  is the optimal value to the Lagrangian dual. However, we may not be able to obtain  $Z_D$  exactly, and an upper bound will suffice. The term  $\lambda_i$  is chosen so that  $0 < \lambda_i \leq 2$ . In our case, we would obtain successive values of  $u^i$ , where  $\gamma^i = (f^i - (n - 1)y^i)$  and  $(f^i, y^i)$  is the optimal solution to TSP' using  $u^i$ , with the relation

$$u^{i+1} = u^i + \theta^i [f^i - (n - 1)y^i].$$

In general, convergence of the subgradient optimization method is guaranteed provided  $\theta^i \rightarrow 0$  and  $\sum_{k=0}^i \theta^k \rightarrow \infty$ . In practice, however, the above definition for  $\theta^i$  is popularly used, and the value of  $\lambda_i$  is halved after a set number of iterations if no improvement results. Note that this halving does not satisfy the convergence criteria given, but it has generally performed well in practice. More detail can be found in Fisher [F1] or Shapiro [S3].

# APPENDIX B

Our discussion is derived from Magnanti, Mireault, and Wong [M2], and Magnanti and Wong [M5]. Because these researchers have had success in applying Benders Decomposition to the Fixed Charge Network Design Problem (NDP), we shall illustrate the method using this problem.

The problem is to select arcs  $y_{ij}$  for a network such that flows  $f_{ij}^k$  can be routed according to demand and supply stipulations. Here  $f_{ij}^k$  represents the flow of commodity  $k$  over arc  $i - j$ , where  $i - j$  is in some set of arcs  $A$ , and  $Q$  is the set of commodities. Both  $y_{ij}$  and  $f_{ij}^k$  have costs attached to them. We let  $O(k)$  be the origin and  $D(k)$  the destination for commodity  $k$ , with  $O(k) \in N$  and  $D(k) \in N$ , where  $N$  is the node set. The flow and arc construction costs are  $c_{ij}^k$  and  $c_{ij}$ , respectively, and  $K_{ij}$  is the capacity of arc  $i - j$ . The formulation is then

$$\text{minimize } Z = \sum_{k \in Q} \sum_{i-j \in A} c_{ij}^k f_{ij}^k + \sum_{i-j \in A} c_{ij} y_{ij} \quad (\text{vii})$$

subject to

$$\sum_{j \in N} f_{ij}^k - \sum_{l \in N} f_{li}^k = \begin{cases} d^k & \text{if } i = O(k) \\ -d^k & \text{if } i = D(k) \\ 0 & \text{otherwise} \end{cases} \quad i \in N, k \in Q \quad (\text{viii})$$

$$\sum_{k \in Q} f_{ij}^k \leq K_{ij} y_{ij} \quad i - j \in A \quad (\text{ix})$$

$$f_{ij}^k \geq 0, y_{ij} = 0 \text{ or } 1 \quad (\text{x})$$

Let us consider the dual of NDP, DNDP. Moving terms to the left in (viii) and multiplying (viii) and (ix) by -1 produces the following dual formulation.

$$\text{maximize } z = \sum_{k \in Q} (u_{D(k)}^k - u_{O(k)}^k) d^k \quad (\text{xi})$$

subject to

$$u_j^k - u_i^k - v_{ij} \leq c_{ij}^k \quad \text{for all } k \in Q, (i, j) \in A \quad (\text{xii})$$

$$K_{ij} v_{ij} \leq c_{ij} \quad \text{for all } (i, j) \in A \quad (\text{xiii})$$

$$u_i^k \text{ unrestricted, } v_{ij} \geq 0. \quad (\text{xiv})$$

If we consider that the conservation of flow constraints (viii) contains a redundant equation for each commodity  $k$ , then we may set  $u_{O(k)}^k = 0$ . The objective (xi) then becomes

$$\text{maximize } z = \sum_{k \in Q} d^k u_{D(k)}^k$$

Now suppose we have a set of values for  $y = (y_{ij})$ . Given these values, NDP becomes a multi-commodity flow problem where constraints (ix) have become bundle constraints. If  $K_{ij} \geq \sum_{k \in Q} d^k$ , then NDP has become  $|Q|$  separate shortest-path problems, one for each commodity  $k$ . If we denote the value of the subproblem formed by a particular  $y$  as  $S(y)$ , then it follows from  $LP$  duality theory that the value of the optimum solution to the dual of the subproblem is a lower bound on  $S(y)$ . This is to say that

$$S(y) \geq \sum_{k \in Q} (d^k u_{D(k)}^k) - \sum_{(i,j) \in A} (v_{ij} K_{ij} y_{ij}) \quad \text{for all dual-feasible } u_i^k, v_{ij}$$

(Note that here we have formed the objective function of the dual to the subproblem of NDP, which has a set value for  $y$ . This gives us the term  $-\sum v_{ij} K_{ij} y_{ij}$  in the objective function, since we have not moved this term to the left as we did to formulate DNDP.) If we let  $u = (u_i^k)$  and  $v = (v_{ij})$ , then we have that

$$S(y) = \text{minimum } z$$

$$z \geq \sum_{k \in Q} (d^k u_{D(k)}^k) - \sum_{(i,j) \in A} (v_{ij} K_{ij} y_{ij}) \quad \text{for all feasible } (u, v)$$

We know that the optimal value to NDP is the minimum of the fixed costs  $c_{ij}y_{ij}$  plus  $S(u)$ , and thus, for NDP (and DNDP), our optimum is found by solving problem MP below.

$C^* = \text{minimum } z$   
subject to

$$z \geq \sum_{(i,j) \in A} c_{ij}y_{ij} + \sum_{k \in Q} (d^k u_{D(k)}^k) - \sum_{(i,j) \in A} (v_{ij} K_{ij} y_{ij}) \text{ for all feasible } (u, v)$$

The decision variables here are  $z$  and  $y$ . This is called the *master problem* and Benders decomposition solves it in the following manner.

We restrict the feasible  $(u, v)$ 's to a small subset of all possible feasible values, and solve the resulting *restricted master problem*, yielding a solution value  $\bar{z}$ . We then solve the NDP subproblem using the resulting  $y$  value, say  $\bar{y}$ . This produces a set of values for  $u$  and  $v$ ,  $(\bar{u}, \bar{v})$ . If it then happens that

$$\bar{z} \geq \sum_{(i,j) \in A} c_{ij} \bar{y}_{ij} + \sum_{k \in Q} (d^k \bar{u}_{D(k)}^k) - \sum_{(i,j) \in A} (\bar{v}_{ij} K_{ij} \bar{y}_{ij}),$$

we have solved MP. This follows from the fact that  $S(\bar{y})$  is the largest value of  $\sum_{k \in Q} (d^k u_{D(k)}^k) - \sum_{(i,j) \in A} (v_{ij} k_{ij} \bar{y}_{ij})$  for all feasible  $(u, v)$ . If  $(\bar{u}, \bar{v})$  does not solve MP, then we add the constraint, or "Benders cut", formed by these vectors, to the restricted master problem and start over.

An additional consideration is that  $\bar{y}$  might not allow a feasible solution in the subproblem of NDP. This would occur if a path of sufficient capacity did not exist between  $O(k)$  and  $D(k)$  for some commodity  $k$ . If this occurs, then there is a cut set  $C$  that separates  $O(k)$  from  $D(k)$ . Suppose we still need a capacity  $\bar{d}^k$  to flow the remainder of commodity  $k$ . We then add the *feasibility constraint*

$$\sum_{(i,j) \in C} K_{ij} y_{ij} \geq \bar{d}^k$$

to the restricted master problem and proceed as described earlier.

Although the procedure for Benders decomposition just outlined is guaranteed to converge, it often does so quite slowly. Magnanti and Wong [M4]



have shown that the convergence may be accelerated by generating the Benders cut creatively. They have shown that it is possible to generate a cut that is superior in the following sense. If we find  $(u', v')$  such that  $du' + v'Ky \geq d\bar{u} + \bar{v}Ky$  for all  $y$ , and for which strict inequality holds for at least one  $y$ , then  $du' + v'Ky$  *dominates* or is *stronger* than  $d\bar{u} + \bar{v}Ky$ . An undominated cut is said to be *pareto-optimal*. (Here  $d = (d^k)$  and  $K = (K_{ij})$ .)

# APPENDIX C

## Cross Decomposition

Cross decomposition is a fusion of Benders decomposition and Lagrangian Relaxation into a unified algorithm for solving mixed integer linear programs. Van Roy (see [V1] and [V2]) developed the technique, in which both the Benders and Lagrangian subproblems pass information to each other and to the master problems in a cohesive, computationally efficient manner. Applications of the technique to problems in capacitated facility location resulted in solutions ten times more quickly obtained than from several recently developed algorithms. In our exposition of this material, we focus on obtaining good *feasible* solutions, notably in the case of a zero duality gap.

Consider problem (Q) below, where  $x$  is an  $m \times 1$  continuous, real-valued vector,  $y$  is an  $n \times 1$  integer vector,  $b$  is  $p \times 1$ , and  $c, d, A$ , and  $B$  are vectors and matrices of conformable dimensions.

$$\min cx + dy$$

(Q)

subject to

$$\begin{aligned} Ax + By &= b \\ x \geq 0, y \in Z &\subseteq \mathfrak{R}^n \end{aligned}$$

We divide the constraints of (Q) into two sets, one with  $p_1$  rows and the other with  $p_2$  rows, where  $p_1 + p_2 = p$ . We also define  $\begin{bmatrix} b_1 \\ b_2 \end{bmatrix} = b$ ,  $\begin{bmatrix} B_1 \\ B_2 \end{bmatrix} = B$ , and  $\begin{bmatrix} A_1 \\ A_2 \end{bmatrix} = A$ , where  $A_1$  has  $p_1$  rows,  $A_2$  has  $p_2$  rows, etc. Also, we assume throughout that the sets  $\{(x, y) : x \geq 0, y \in Z, A_1 x + B_1 y = b\}$

are nonempty and bounded. For our development,  $y$  and  $Ax + B_2y = b_2$  are complicating variables and constraints, respectively. Thus,  $(SQ\hat{y})$  and  $(SD\hat{u})$  below are easily solved problems.

$(SQ\hat{y})$

$$\begin{aligned} & \text{minimize} && cx \\ & \text{subject to} && \\ & && Ax = b - B\hat{y} \\ & && x \geq 0 \end{aligned}$$

$(SD\hat{u})$

$$\begin{aligned} & \text{minimize} && cx + dy + \hat{u}_2(b_2 - A_2x - B_2y) \\ & \text{subject to} && \\ & && A_1x + B_1y = b_1 \\ & && x \geq 0, y \in Z \end{aligned}$$

In Benders decomposition, we solve

$$\begin{aligned} (Q) : & \min_{y \in Z} \min_{x \geq 0} cx + dy \equiv \min_{y \in Z} v(SQy) \\ & \text{subject to} && Ax + By = b \\ & && = \min_{y \in Z} \max_u u(b - By) + dy \\ & \text{subject to} && uA \leq c \\ & && = \min_{y \in Z} \max_{t \in E_Q} u^t(b - By) + dy \\ & && = \min_{y \in Z, v} v \\ (MQ) & \text{subject to} && u^t b + (d - u^t B)y \leq v \quad t \in E_Q, \end{aligned}$$

where  $E_Q$  is the index set of extreme points of  $\{u : uA \leq c\}$ . In our notation,  $(MQ)$  is the full Benders master problem and  $(SQy)$  is the Benders

subproblem. (Notice that we are not considering extreme rays, for the purpose of exposition.)

In Lagrangian Relaxation, we solve

$$(D) \quad \max_{u_2} \min_{x \geq 0, y \in Z} cx + dy + u_2(b_2 - A_2x - B_2y) \equiv \max_{u_2} v(SD_{u_2})$$

$$\text{subject to} \quad A_1x + B_1y = b_1$$

$$= \max_{u_2} w$$

(MD)

$$\text{subject to} \quad cx^t + dy^t + u_2(b_2 - A_2x^t - B_2y^t) \geq w, \quad t \in E_D$$

where  $E_D$  is the index set of the extreme points of the convex hull of the set  $\{(x, y) : x \geq 0, y \in Z, A_1x + B_1y = b_1\}$ . The duality gap is  $v(P) - v(D)$ .

Van Roy has shown that the Benders subproblem  $(SQ\hat{y})$ , where  $\hat{y} \in Z$ , is equivalent to a certain restricted form of the Lagrangian master problem  $(MD)$ , where the constraints of the restricted master are generated from  $\hat{y}$ . Likewise, he has shown that the Lagrangian subproblem  $(SD\hat{u}_2)$  is equivalent to a restricted Benders master problem, where the constraints of the restriction are generated from  $\hat{u}_2$ . We now define precisely this notion of equivalence and detail how the constraints of the restricted master problems are generated.

#### Definition C.1

A problem  $(Q)$  is equivalent to  $(Q')$  with respect to a subset of primal and/or dual variables  $U$  if the optimal solutions of  $(Q)$ , given  $U$ , are optimal for  $(Q')$  and vice versa.

Thus, problem  $(Q)$  and its full master  $(MQ)$  are equivalent with respect to  $y$ , and problems  $(D)$  and  $(MD)$  are equivalent with respect to  $u_2$ . As Van Roy [V1] notes, the definition includes both primal and dual variables, so that any linear program and its dual are equivalent with respect to any subset of primal and dual variables. To show how we generate the constraints for the restricted master problems, we introduce the following definitions.

$$T\hat{u}_2 \equiv \{ \text{All indices } t : u^t = (u_1^t, \hat{u}_2) \text{ is an extreme point of } \{(u_1, \hat{u}_2) : u_1A_1 \leq c - \hat{u}_2A_2\} \}.$$

$\bar{T}\hat{u}_2$  {All indices  $t : u^t = (u_1^t, \hat{u}_2)$  is an extreme point of  $\{u : uA \leq c\}$ }.

We further define  $(MQ\hat{u}_2)$  as the restriction of  $(MQ)$  to constraints generated from indices of  $T\hat{u}_2$ . We define  $T\hat{y}$ ,  $\bar{T}\hat{y}$ , and  $(MD\hat{y})$  similarly.

Figure C-1 illustrates that  $\bar{T}\hat{u}_2 \subseteq T\hat{u}_2$ . For any index  $t$ , if  $t \in \bar{T}\hat{u}_2$  and  $t \in T\hat{u}_2$ , then  $t \in E_Q$ . Thus,  $T\hat{u}_2 \subseteq E_Q$  only if  $\bar{T}\hat{u}_2 = T\hat{u}_2$ . However, any index of  $T\hat{u}_2$  represents a feasible point of  $\{u : uA \leq c\}$ , so even if  $t \in E_Q$ ,  $u^t$  can be expressed as a linear combination of points whose indices are in  $E_Q : u^t = \sum_{k \in E_Q} \lambda^k j^k, \sum_{k \in E_Q} \lambda^k = 1, \lambda^k \geq 0$ . Thus, the constraint generated by  $u^t$  is redundant in the full master problem  $(MQ)$ . Even so, we do have the following sequence of inequalities:

$$v(MP : t \in \bar{T}\hat{u}_2) \leq v(MP\hat{u}) \equiv v(MP : t \in T\hat{u}_2) \leq v(MQ).$$

Thus,  $(MP\hat{u}_2)$  generates a (potentially) better value than  $(MP : t \in \bar{T}\hat{u}_2)$ . An analogous relationship exists for  $\bar{T}\hat{y}$  and  $T\hat{y}$ . Thus,  $v(MD : t \in \bar{T}\hat{y}) \geq v(MD\hat{y}) \equiv v(MD : t \in T\hat{y}) \geq v(MD)$ . We are now in a position to state the main theoretical results.

### Theorem C.1 (Van Roy)

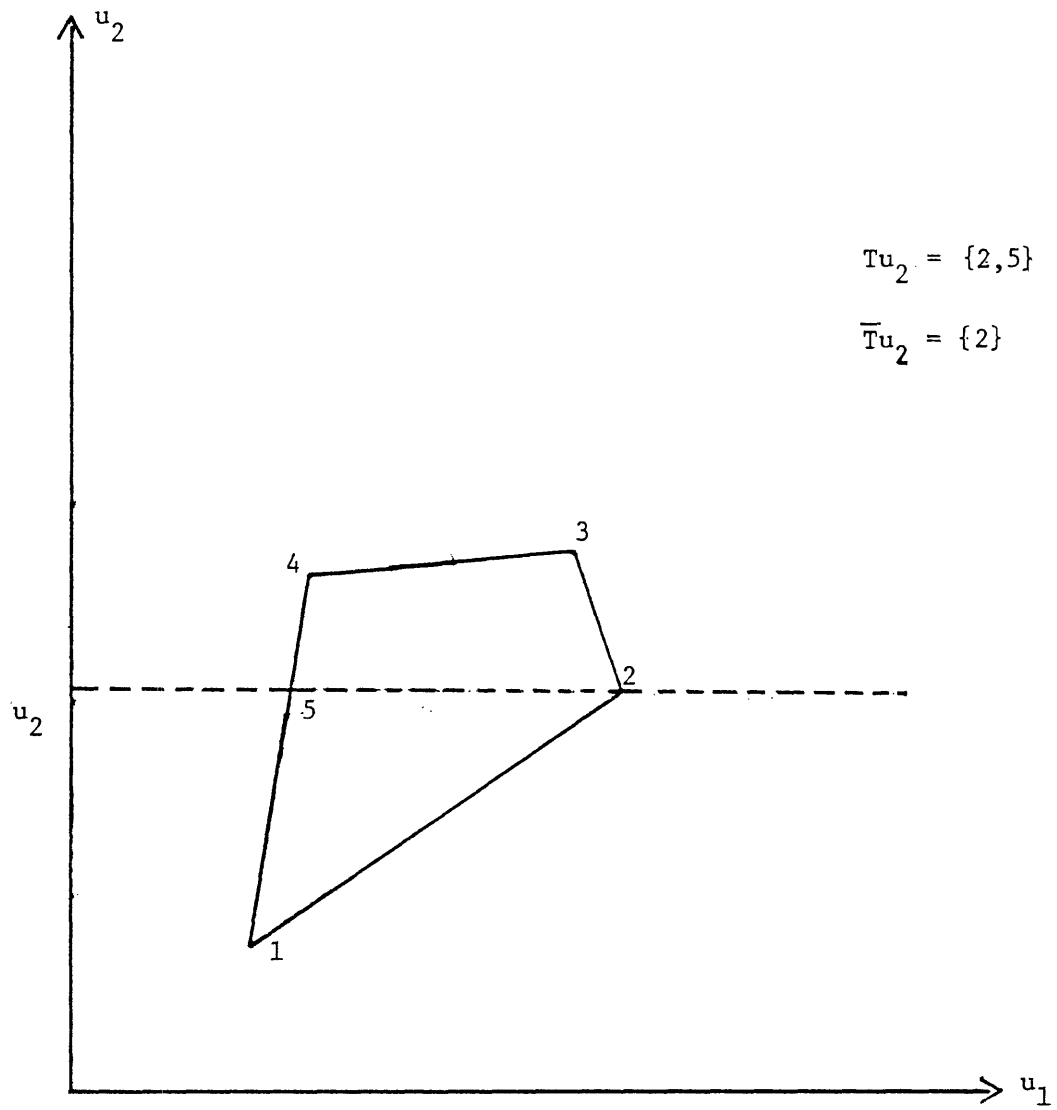
a.) The restricted Benders master problem  $(MQu_2)$  is equivalent to the Lagrangian subproblem  $(SDu_2)$  with respect to  $y$ , and  $v(MQu_2) = v(SDu_2)$ .

b.) The restricted Lagrangian dual  $(MDy)$  is equivalent to the Benders subproblem  $(SQy)$  with respect to  $u_2$ , and  $v(MDy) = v(SQy)$ .

Theorem C.1 unifies Benders decomposition and Lagrangian relaxation in a way that lays the foundation for solving many mixed integer linear programs very efficiently. In fact, the next theorem shows that if the duality gap relative to the constraints  $A_2x + B_2y = b$  is zero, then optimality is potentially verifiable in only one iteration of the cross decomposition algorithm.

### Theorem C.2 (Van Roy)

Let  $(x^*, y^*)$  and  $u_2^*$  be optimal solutions of  $(Q)$  and  $(D)$ , respectively. Then (a), (b), and (c) are equivalent:



Example showing that  $\bar{T}u_2 \subseteq Tu_2$ .

**Figure C-1. Example Showing That  $\bar{T}u_2 \subseteq Tu_2$**

(a) The Lagrangian relaxation relative to the constraints  $A_2x + B_2y = b$  has no duality gap.

(b) There is an optimal dual solution  $\hat{u}$  of  $(SQy^*)$  such that  $v(SD\hat{u}_2) = v(SQy^*)$ .

(c) There is an optimal primal solution  $(\hat{x}, \hat{y})$  of  $(SDu_2^*)$  such that  $v(SQ\hat{y}) = v(SDu_2^*)$ .

Theorem C.2 gives strong motivation to use the dual variables of  $(SQy^*)$  as Lagrange multipliers, since with a zero duality gap at least one optimal Lagrange vector  $\hat{u}_2$  is part of a dual optimal solution  $\hat{u} = (\hat{u}_1, \hat{u}_2)$  for  $(SQy^*)$ .

The next corollary shows that in fact any optimal multiplier  $u_2^*$  suffices to produce an optimal feasible solution to  $(Q)$ , in the case of a zero duality gap. Moreover, we demonstrate that any optimal feasible solution has an optimal Lagrange multiplier when  $v(Q) = v(D)$ .

**Corollary C.3:**

Let  $v(Q) = v(D)$ ; also let  $(x^*, y^*)$  be optimal in  $(Q)$ , and let  $u_2^*$  be optimal in  $(D)$ . Then

(a.) There exists an optimal feasible solution  $(\hat{x}, \hat{y})$  of  $(Q)$  that is optimal in  $(SDu_2^*)$ .

(b.) There exists  $\hat{u}_2$  such that  $(x^*, y^*)$  is optimal in  $(SD\hat{u}_2)$ .

**Proof:**

(Part (a.)) By part (c) of Theorem C.2 there is an optimal solution  $(x^0, \hat{y})$  of  $(SDu_2^*)$  such that  $v(SQ\hat{y}) = v(SDu_2^*)$ . Since  $v(SDu_2^*) \leq v(Q) \leq v(SQ\hat{y})$ , any optimal primal solution  $\hat{x}$  of  $(SQ\hat{y})$  provides an optimal feasible solution  $(\hat{x}, \hat{y})$  to  $(Q)$ . If  $\hat{x}$  is feasible in  $(Q)$ , then  $A\hat{x} = b - B\hat{y}$ . Thus,  $(\hat{x}, \hat{y})$  is optimal in  $(SDu_2^*)$ , since

$$\begin{aligned} v(SDu_2^*) &= cx^0 + d\hat{y} + u_2^*(b - B\hat{y} - Ax^0) \leq c\hat{x} + d\hat{y} + u_2^*(b - B\hat{y} - A\hat{x}) \\ &= c\hat{x} + d\hat{y} = v(SQ\hat{y}). \end{aligned}$$

Since  $v(SDu_2^*) = v(SQ\hat{y})$ , part (a) is proven.

(Part (b.)) By part (b) of Theorem C.2 an optimal dual solution  $\hat{u}_2$  of  $(SQy^*)$  exists such that  $v(SD\hat{u}_2) = v(SQy^*)$ . Since  $v(SQy^*) = cx^* + dy^* = cx^* + dy^* + \hat{u}_2(b - By^* - Ax^*) \geq v(SD\hat{u}_2)$ , it follows that  $(x^*, y^*)$  is optimal in  $(SD\hat{u}_2)$ . This proves part (b).

Corollary C.3 allows us to refine our characterization of optimality with a zero duality gap. Each feasible optimal solution  $(x^*, y^*)$  of  $(Q)$  is optimal in  $(SDu_2^*)$  for some  $u_2^*$ , and any optimal solution  $u_2^*$  of  $(D)$  has an associated feasible solution  $(x^*, y^*)$  of  $(Q)$  as an optimal solution of  $(SDu_2^*)$ . In a very real sense then, using dual optimal solutions to the Benders subproblem will not “eliminate” all optimal feasible solutions to  $(Q)$  from consideration, so that if there is no duality gap we will not simply end up with an optimal multiplier  $u_2^*$  and no associated *feasible* solution.

Given this and earlier information about optimal Lagrange multipliers, we have further reason to use the Benders subproblem dual optimal variables as part of the Lagrangian dual solution process. For, there is not only an excellent physical interpretation for these variables, but also an excellent theoretical interpretation as well, as Theorem C.1 shows. The important implication of this theorem is that passing information back and forth between the subproblems  $(SQy)$  and  $(SDu)$  can accelerate the convergence of either a Lagrangian Relaxation or Benders Decomposition. Thus, alternately solving  $(SQy)$  and  $(SDu)$  seems like a reasonable strategy. However, even with a zero duality gap we could never guarantee convergence. Cycling could occur unless we take preventative measures. We must consequently incorporate an additional technique that is known to converge. The following lemma provides us with a test for convergence and a guide for using the appropriate master problem or subproblem when necessary.

**Lemma C.4 (Van Roy)**

(a.) If  $u^0$  is dual optimal for  $(SQy^0)$ , and  $(\hat{x}, \hat{y})$  is optimal for  $(SDu_2^0)$ , then  $(x^0, y^0) \neq (\hat{x}, \hat{y})$  unless  $v(SQy^0) = v(Q)$ .

(b.) If  $(x^0, y^0)$  is optimal for  $(SDu_2^0)$ , and  $\hat{u}$  is dual optimal for  $(SQy^0)$ , then  $u^0 \neq \hat{u}$  unless  $v(SDu_2^0) = v(Q)$ .

Suppose that we iterate between the problems  $(SQy)$  and  $(SDu)$ . Let  $u^k$  and  $(x^k, y^k)$  be the dual and primal solutions respectively at iteration  $k$ . As an example, let us solve  $(SQy^k)$  first. The dual optimal solution  $u^k$  of  $(SQy^k)$  is then used to generate  $u_2^{k+1}$ . That is, we set  $u_2^{k+1} = u_2^k$ , and then solve  $(SDu_2^{k+1})$ . We then set  $y^{k+2} = y^{k+1}$ , where  $(x^{k+1}, y^{k+1})$  solves  $(SDu_2^{k+1})$ , and so on. Thus, we shall solve  $(SQy^k), (SDu^{k+1}), (SQy^{k+2}),$



$(SDu_2^{k+3}), \dots$  By lemma C.4,  $y^k \neq y^{k+1}$  unless we have attained optimality. Since our construction sets  $y^{k+2} = y^{k+1}$ , the earliest possible replication of a primal solution short of optimality is if  $y^k = y^{k+3}$ . Similar comments apply to a sequence of dual solutions.

We can now state the basic decomposition algorithm. We let  $v_D$  and  $v_Q$  be respectively the best values of  $(SDu)$  and  $(SQy)$  obtained so far. Similarly,  $w_0$  and  $p_0$  represent the current values of the restricted dual and primal master problems, respectively.

### Cross Decomposition Algorithm

#### Step 1: Initialization

Set the iteration counter  $k \leftarrow 0$ . Also, set  $v_Q = w_0 = +\infty$ , and set  $v_D = p_0 = -\infty$ . Initialize the primal and dual master problem constraint index sets,  $T_Q = T_D = \{\}$ . Set  $\alpha \leftarrow 1, \delta \leftarrow 0$ , and select  $u_2^1$ .

Step 2: (a) Set  $k \leftarrow k + 1$ . Solve  $(SDu_2^k)$ , and let  $(x^k, y^k)$  be an optimal solution. Set  $T_D = T_D \cup \{k\}$ , and  $\alpha \leftarrow \alpha + 1$ .  
 (b) If  $v_D < v(SDu_2^k)$ , then set  $v_D \leftarrow v(SDu_2^k)$ ,  
 If  $v_D \geq w_0$ , then  $(D)$  is solved and set  $\delta = 1$ .  
 If  $v_D \geq v_Q$ , then  $(\bar{x}, \bar{y})$  constitutes an optimal solution of  $(Q)$ .  
 Convergence test: If  $\alpha = 4$ , then go to step (4b).  
 Otherwise, set  $y^{k+1} \leftarrow y^k$ , and go to step (3).

Step 3: (a) Set  $k \leftarrow k + 1$ . Solve  $(SQy^k)$ , and let  $x^k$  and  $u^k$  be optimal primal and dual solutions. Set  $T_Q \leftarrow T_Q \cup \{k\}$ , and  $\alpha \leftarrow \alpha + 1$ .  
 (b) If  $v_Q > v(SQy^k)$ , then set  $v_Q \leftarrow v(SQy^k)$ ,  
 $(\bar{x}, \bar{y}) \leftarrow (x^k, y^k)$ , and  $\alpha \leftarrow 1$ .  
 If  $v_Q \geq v_D$  or  $v_Q \geq p_0$ , then  $(\bar{x}, \bar{y})$  is optimal in  $(Q)$ .  
 (c) Convergence test: If  $\delta = 1$ , then go to step (4b);  
 otherwise, if  $\alpha = 4$ , then go to step (4a);  
 otherwise, set  $u_2^{k+1} \leftarrow u_2^k$  and go to step (2).

Step 4: (a) Solve the restricted dual master problem  $(MD_{T_D})$ ,

and let  $(w_0, u_2^{k+1})$  be an optimal solution.

Set  $\delta \leftarrow 0$ , and  $\alpha \leftarrow 1$ .

(b) Solve the restricted primal master problem  $(MQ_{T_Q})$

and let  $(p_0, y^{k+1})$  be an optimal solution.

Set  $\alpha \leftarrow 1$ .

If  $v_Q \leq p_0$ , then  $(\bar{x}, \bar{y})$

is an optimal solution of  $(Q)$ ; otherwise, go to step (3).

The cross decomposition algorithm as presented will converge by virtue of lemma C.4 and the fact that Benders decomposition and Dantzig-Wolfe decomposition are finite. However, we can enhance the algorithm's performance significantly by generating constraints more judiciously. To do this, we introduce the concept of an *efficient cutset*.

**Definition C.2**

An efficient cutset  $T_E$  for a problem  $(M)$  is a set of cuts (constraints) such that

$$(a) \quad v(M_E) \equiv v(M : t \in T_E) = v(M);$$

$$(b) \quad v(M : t \in T_E - t^1) \neq v(M) \text{ for any } t^1 \in T_E.$$

In the general case, one would like to find efficient cutsets for  $Tu_2^k$  and  $Ty^k$  at each iteration of the algorithm. If we can do this in a computationally inexpensive way, we would expect faster convergence.

# REFERENCES

- A1 Andrade, J., Federal Express Corporation, personal communication.
- A2 Assad, A.A., "Models for Rail Transportation", Transportation Research, Vol. 14A, 1980, pp. 205-220.
- A3 Aviation Daily, July 17, 1987, p. 94.
- B1 Ball, M., L. Bodin, and R. Dial, "A Matching Based Heuristic for Scheduling Mass Transit Crews and Vehicles", Transportation Science, Vol. 17, No. 1, February 1983, pp. 4-31.
- B2 Ball, M.O., and U. Derigs, "An Analysis of Alternative Strategies for Implementing Matching Algorithms", Networks, Vol. 13, No. 4, 1983, pp. 517-549.
- B3 Bertossi, A.A., P. Carreresi, and G. Gallo, "On Some Matching Problems Arising in Vehicle Scheduling Models", Networks, Vol. 17, No. 3, 1987, pp. 271-281.
- B4 Bodin, L., B. Golden, A. Assad and M. Ball, "Routing and Scheduling of Vehicles and Crews - The State of the Art", Computers and Operations Research, Vol. 10, No. 2, 1983.
- C1 Clarke, G., and J. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", Operations Research, Vol. 12, 1964, pp. 568-581.
- C2 Corneujols, G., M.L. Fisher, G.L. Nemhauser, "Location of Bank Accounts to Optimize Float: An Analytic Study of Exact and Approximate Algorithms", Management Science, Vol. 23, 1977, pp. 789-810.
- C3 Crowder, H., and M.W. Padberg, "Solving Large-Scale Traveling Salesman Problems to Optimality", Management Science, Vol. 26, 1980, pp. 495-509.
- D1 Dantzig, G.B., D.R. Fulkerson, and S.M. Johnson, "Solutions of a Large-Scale Traveling Salesman Problem", Operations Research, Vol. 2, 1954, pp. 393-410.
- D2 de Neufville, R., "Air Cargo in the 1980's and Beyond - A Manual for Planners", Center for Transportation Studies, M.I.T., 1984.

- D3 Derigs, U., "A Shortest Path Method for Solving Minimal Perfect Matching Problems", Networks, Vol. 11, 1981, pp. 379-390
- D4 Desrosiers, J., P. Pelletier and F. Soumis, "Shortest Path with Schedule Constraints", RAIRO, Recherche Operationnelle, Vol. 17, No. 4, Nov. 1983, pp. 357-377.
- D5 Desrosiers, J., F. Soumis, and M. Derosiers, "Routing with Time Windows by Column Generation", Networks, Vol. 14, 1984, pp. 545-565.
- E1 Erlenkotter, D., "A Dual-Based Procedure for Facility Location", Operations Research, Vol. 26, 1978, pp. 992-1009.
- F1 Fisher, M.L., "The Lagrangian Relaxation Method for Solving Integer Programming Problems", Management Science, Vol. 27, 1981, pp. 1-18.
- F2 Fisher, M.L., A.J. Greenfield, R. Jaikumar, P. Kedia, "Real-Time Scheduling of a Bulk Delivery Fleet: A Practical Application of Lagrangian Relaxation", Department of Decision Sciences, The Wharton School, U. of Pa., October, 1982.
- F3 Fisher, M.L., and D.S. Hochbaum, "Database Location in Computer Networks", JACM, Vol. 27, 1980, pp. 718-735.
- F4 Fisher, M.L., and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing", Networks, Vol. 11, 1981, pp. 109-124.
- F5 Fisher, M., "A Dual Algorithm for the One-Machine Scheduling Problem", Mathematical Programming, Vol 11, 1976, pp. 229-251.
- F6 Fisher, M., W.D. Northup, and J.F. Shapiro, "Using Duality to Solve Discrete Optimization Problems: Theory and Computational Experience", Mathematical Programming Study, Vol. 3, 1975, pp. 56-94
- F7 Fisher, M., and J.F. Shapiro, "Constructive Duality in Integer Programming", SIAM J. Appl. Math., Vol 27, 1974, pp. 31-52.
- G1 Gavish, B., "Topical Design of Centralized Computer Networks - Formulations and Algorithms", Networks, Vol. 12, 1982, pp. 355-377.
- G2 Gavish, B., and S. Graves, "The Traveling Salesman Problem and Related Problems", Working Paper No. 7906, Graduate School of Management, University of Rochester, April 1979.
- G3 Gavish, B., and K. Srikanth, "An Optimal Solution Method for Large-Scale Multiple Traveling Salesmen Problems", Operations Research, Vol. 34 No. 5, 1986, pp. 698-717.
- G4 Geoffrion, A.M., and G.W. Graves, "Multicommodity Distribution System Design by Benders Decomposition", Management Science, Vol. 20, 1974, pp.822-884.

- G5 Golden, B.L., and A.A. Assad, "Perspectives on Vehicle Routing: Exciting New Developments", Operations Research, Vol. 34, No. 5, 1986, pp. 803-810.
- G6 Golden, B., E. Baker, J. Alfaro, and J. Schaffer, "The Vehicle Routing Problem with Backhauling: Two Approaches", Working Paper Series MS/S 85-037, College of Business and Management, University of Maryland at College Park, 1985.
- G7 Golden, B.L. and T.L. Magnanti, Course Notes for Network Optimization, Sloan School of Management, M.I.T., 1980.
- G8 Grötschel, M., and M.W. Padberg, "On the Symmetric Travelling Salesman Problem I: Inequalities", Mathematical Programming, Vol. 16, pp. 265-280.
- G9 Guignard, M., "Fractional Vertices, Cuts, and Facets of the Simple Plant Location Problem", Mathematical Programming Study 12, 1980, pp. 150-162.
- H1 Held, M. and R.M. Karp, "The Traveling Salesman Problem and Minimum Spanning Trees", Operations Research, Vol. 18, 1970, pp. 1138-1162.
- H2 Hinson, J., Federal Express Corporation, personal communication.
- H3 Hinson, J., and S. Mulherkar, "Improvements to the Clarke and Wright Algorithm as Applied to an Airline Scheduling Problem", Technical Report, Federal Express Corporation, 1975.
- J1 Johnson, D.S., J.K. Lenstra, A.H.G. Rinnooy Kan, "The Complexity of the Network Design Problem", Networks, Vol. 8, 1978, pp. 279-285.
- K1 Karmarkar, N., "A New Polynomial-Time Algorithm for Linear Programming", Technical Report, AT&T Bell Laboratories, 1984.
- L1 Lasdon, L.S., Optimization Theory for Large Systems, The Macmillan Company, 1970.
- L2 Lenstra, J.K., and A.H.G. Rinnooy Kan, "Complexity of Vehicle Routing and Scheduling Problems", Networks, Vol. 11, 1981, pp. 221-227.
- M1 Magnanti, T.L., "Combinatorial Optimization and Vehicle Fleet Planning: Perspectives and Prospectives", Networks, Vol. 11, 1981, pp. 179-213.
- M2 Magnanti, T.L., P. Mireault, and R.T. Wong, "Tailoring Benders Decomposition for Network Design", Mathematical Programming Study 26, 1986, pp. 112-154
- M3 Magnanti, T.L., J.F. Shapiro, and M.H. Wagner, "Generalized Linear Programming Solves the Dual", Management Science, Vol. 22, No. 11, 1976, pp. 1195-1203.

- M4 Magnanti, T.L., and R.T. Wong, "Accelerating Benders Decomposition: Algorithmic Enhancement and Model Selection Criteria", Operations Research, Vol. 29, 1984, pp. 464-484.
- M5 Magnanti, T.L., and R.T. Wong, "Network Design and Transportation Planning: Models and Algorithms", Transportation Science, Vol. 18, 1984, pp. 1-55.
- M6 Marsten, R., "The Use of the Boxstep Method in Discrete Optimization", Mathematical Programming Study 3, 1975, pp. 127-144.
- M7 Marsten, R., and M.R. Muller, "Network Design and Aircraft Fleet Planning at Flying Tigers: A Successful Application of Mixed-Integer Programming", MIS Technical Report 80-1, University of Arizona, January, 1980.
- M8 Miliotis, T., "Integer Programming Approaches to the Traveling Salesman Problem", Mathematical Programming, Vol. 10, 1976, pp. 367-378.
- M9 Mirzaian, A., "Lagrangian Relaxation for the Star-Star Concentrator Location Problem: Approximation Algorithm and Bounds", Networks, Vol. 15, No. 1, 1985, pp. 1-20.
- M10 Murphy, J., Federal Express Corporation, personal communication.
- N1 Nemhauser, G.L., and G.M. Weber, "Optimal Set Partitioning, Matchings and Lagrangian Duality", Naval Research Logistics Quarterly, Vol. 26, 1979, pp. 553-563.
- O1 O'Kelly, M.E., "The Location of Interacting Hub Facilities", Transportation Science, Vol. 20, No. 2, May 1986, pp. 92-106.
- P1 Powell, W.B., and Y. Sheffi, "The Load Planning Problem of Motor Carriers: Problem Description and a Proposed Solution Approach", Transportation Research, Vol. 17A, 1983, pp. 471-480.
- R1 Rardin, R.L., and U.I. Choe, "Tighter Relaxations of Fixed Charge Flow Problems", Industrial and Systems Engineering Report Series No. J-79-18, May, 1979.
- R2 Richardson, R., "An Optimization Approach to Routing Aircraft", Transportation Science, Vol. 10, 1976, pp. 52-71.
- R3 Rothfarb, B. and M.C. Goldstein, "The Pre-terminal Telpak Problem", Operations Research, Vol. 19, 1971, pp. 156-169.
- S1 Schrage, L. "Implicit Representation of Variable Upper Bounds in Linear Programming", Mathematical Programming Study 4, 1975, pp. 118-132.

- S2 Schwartz, M., Computer-Communication Network Design and Analysis, Prentice-Hall, Inc., 1977.
- S3 Shapiro, J.F., Mathematical Programming: Structures and Algorithms, Wiley-Interscience, 1979.
- S4 Simpson, R., "Scheduling and Routing Models for Airline Systems", Technical Report, Flight Transportation Laboratory, M.I.T., 1969.
- S5 Singhal, V.M., "Point -to-Point Package Delivery Systems", M.S. Thesis, Dept. of Electrical Engineering and Computer Science, M.I.T., 1984.
- T1 Tansel, B.C., R.L. Francis, and T.L. Lowe, "Location on Networks: A Survey, Parts I and II", Management Science, Vol. 29, 1983, pp. 482-510.
- U1 Urquhart, R.J., "Degree Constrained Subgraphs of Linear Graphs", Ph.D. dissertation, The University of Michigan, Ann Arbor, 1967.
- V1 Van Roy, T.J., "Cross Decomposition for Mixed-Integer Programming", Math. Programming, Vol. 25, 1983, pp. 46-63.
- V2 Van Roy, T.J., "A Cross Decomposition Algorithm for Capacitated Facility Location", Operations Research, Vol. 34, No. 1, 1986, pp. 145-163.
- V3 Van Roy, T.J., and D. Erlenkotter, "A Dual-Based Procedure for Dynamic Facility Location", Management Science, Vol. 28, 1982, pp. 1091-1105.
- W1 Wong, R.T., "Accelerating Benders Decomposition for Network Design", Ph.D. Thesis, M.I.T., 1978.
- W2 Wong, R.T., "A Dual Ascent Approach for Steiner Tree Problems on a Directed Graph", Mathematical Programming, Vol. 28, 1984, pp. 271-287.