To the Graduate Council:

I am submitting herewith a dissertation written by Julian J. Ray entitled "The Flow-Resource Facility Location Problem." I have examined the final copy of this dissertation for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Doctor of Philosophy, with a major in Geography.

Bruce A. Ralston, Major Professor

We have read this dissertation
and recommend its acceptance:

Thomas L. Bell

Charles E. Noon

Accepted for the Council:

Vice Provost
and Dean of The Graduate School

THE FLOW-RESOURCE FACILITY LOCATION PROBLEM

A Dissertation
Presented for the
Doctor of Philosophy
Degree
The University of Tennessee, Knoxville

Julian J. Ray
August 1990

ACKNOWLEDGEMENTS

During the course of my graduate work in the United States, a number of people have assisted me. In particular, I would like to acknowledge my committee chairman, Dr. Bruce A. Ralston, for his support, encouragement, and cooperation throughout the course of this research, and indeed, the whole of my graduate experience in the United States. Sincere thanks and appreciation also go to the other members of my dissertation committee; Dr. Thomas L. Bell, Dr. Sidney J. Jumper, and Dr. Charles E. Noon for their support and suggestions during the preparation of this manuscript. I would also like to acknowledge several members of the Transportation Research Center staff at Oak Ridge National Laboratory who have lent assistance in one way or another. In particular, I would like to extend my appreciation to Dr. Frank Southworth and to Ingrid K. Busch.

I hereby acknowledge with thanks, the United States Military Airlift Command (HQ MAC/DO-ADANS) who supported this research through Inter-Agency Agreement 1660-1660-A1 between HQ. MAC (U.S. Airforce) and U.S. DOE, and the members of the Airlift Deployment Analysis (ADANS) development team at Oak Ridge National Laboratory who have lent assistance in one way or another.

ABSTRACT

The problem described in this dissertation is the Flow-Resource Facility Location Problem (FRFLP). The FRFLP is the problem of determining a minimum cost distribution of flow-resource facilities at the vertices of a network, and of simultaneously determining a set of paths in the network such that a maximum amount of commodities or services can be moved from a set of origins to a set of destinations. Unlike conventional facility location models, the facilities of the FRFLP are neither origins nor destinations for commodities or services moving over the network. Rather, the flow-resource facilities are located at intermediate vertices along the paths between origin and destination pairs. For a particular route to be feasible, flow-resource facilities are required at the intermediate vertices. An example of a flow-resource facility is an aircraft refueler at an airbase. Any vertex could be both origin and a destination as well as a feasible location for flow-resource facilities. The solution to the FRFLP is a distribution of flow-resource facilities at the vertices of the network and a set of paths between the origins and destinations.

A flow-resource facility in this problem is a type of facility which is used to control the loss of flow-volume between origins and destinations in the network. Typically, flow-resource facilities are used in systems where the volume of commodities or services that can be moved directly between any two vertices decreases with increasing distance. A trade-off can then be made between the distance moved and the amount of commodities or services that can be taken. Aircraft, for example, can trade-off fuel for cargo and cargo for fuel. The flow-volume for a given route is, thus, determined by the length of the longest

arc traversed. Aircraft can decrease the maximum length arc of a given trip by diverting to an enroute airbase to refuel, thus, decreasing the maximum weight of fuel required for the longest arc on the route.

In this dissertation, the FRFLP is formulated as a Mixed Integer Programming problem (MIP). The size of the MIP for the FRFLP is shown to be combinatorial with respect to the number of vertices in the network. To this end two specialized algorithms are proposed for solving the FRFLP. The first algorithm is based on a mathematical programming technique termed column generation which allows the FRFLP to be decomposed into a facility location problem and a routing problem submodel. The original statement of the column generation algorithm is modified into a heuristic procedure designed to solve the FRFLP by utilizing a system of surrogate costs which are used to approximate the dual costs from the optimal solution to the restricted master problem. The column generation submodel is solved using a specialized network heuristic designed to construct routes between origin-destinations pairs. The restricted master problem of the first algorithm is solved using conventional methods.

The second algorithm expands on the first algorithm by utilizing a specialized heuristic to solve the facility location problem. A two-stage heuristic is illustrated which is designed to solve the restricted master problem faster and more efficiently than by conventional means. Algorithm II is applied to a real-world problem which consists of locating aircraft refuelers for a military transportation operation. The solution generated by Algorithm II is shown to be within 1.13% of the optimal solution. This problem is based on a realistic data set supplied by the United States Military Airlift Command.

TABLE OF CONTENTS

Chapter	Page
1. Introduction And Literature Review	1
1.1. Statement Of The Problem	6
1.2. Scope Of Research	9
1.3. Literature Review	11
1.3.1. Notation And Assumptions	11
1.3.2. Network Location Problems	12
1.3.2.1. Median Problems	13
1.3.2.2. Center Problems	17
1.3.2.3. Other Network Location Problems	18
1.3.3. Location-Routing Problems	19
1.3.3.1. Optimal Models	19
1.3.3.2. Heuristic Models	20
1.3.3.3. Other Location-Routing Models	24
1.4. Summary Of Chapter 1	27
2. Problem Formulation	28
2.1. Preliminaries	28
2.1.1. Model Assumptions	29
2.1.1.1. Flow-Resource Facilities	29
2.1.1.2. Supply And Demand	31

Chapter	Page

2. (Continued)

2.1.1.3. Network Capacity 32
2.1.2. Optimality Conditions 34
2.1.3. Path Capability 35
2.2. Formulation 37
2.2.1. Additional Notation 37
2.2.2. MIP Formulation 38
2.3. Solution Strategies For The FRFLP 40
2.4. Summary Of Chapter 2 44
3. Algorithm I: Column Generation 45
3.1. Column Generation In Linear Programs 45
3.2. Column Generation And The FRFLP 49
3.2.1. The Column Generation Heuristic 49
3.2.2. The Path Submodel 54
3.2.3. Statement Of Algorithm I 60
3.3. Test Of Algorithm I 66
3.4. Summary Of Chapter 3 75
4. Algorithm II: Location-Allocation 78
4.1. Location-Allocation Submodel 78
4.1.1. Generating An Initial Feasible Solution 79
4.1.2. Improving The Initial Solution 84
4.2. Statement Of Algorithm II 86
4.3. Test Of Algorithm II 88

Chapter Page

4. (Continued)

4.4. Summary Of Chapter 4 89

5. Test Problem 2 92

5.1. Problem Description 92

5.2. Solution To Test Problem 2 95

5.3. Calibrating δ 110

5.4. Summary Of Chapter 5 116

6. Conclusions And Final Comments 117

6.1. Conclusions 117

6.1.1. The FRFLP As A Design Tool 117

6.1.2. Column Generation As An Appropriate Solution Technique 119

6.1.3. A Comparison Of Algorithm I and Algorithm II 123

6.2. Extensions And Future Work 124

6.3. Final Comments 126

List Of References 127

Appendix 132

Vita 134

LIST OF TABLES

Table	Page
3.1. Algorithm I Progress For Test Problem 1	73
4.1. Algorithm II Progress For Test Problem 1	90
5.1. Test Problem 2: Airbase Characteristics	94
5.2. Test Problem 2: Channel Characteristics	96
5.3. Algorithm II Progress For Test Problem 2	97
5.4. Objective Function Values For Varying Values Of δ	113

LIST OF FIGURES

Figure	Page
3.1. Example Seven Vertex Network	51
3.2. Example Network Showing The Labeling Algorithm	58
3.3. Flow Diagram For Algorithm I	64
3.4. Test Problem 1: Network	67
3.5. Algorithm I Iterations For Test Problem 1	68-72
3.6. Algorithm I Iteration 6 For Test Problem 1	76
4.1. Flow Diagram For Algorithm II	87
5.1. Test Problem 2: Network And Airbases	93
5.2. Test Problem 2: Distribution Of Aircraft Refuelers	99
5.3. Test Problem 2: Solution By Channels	100-109
5.4. Objective Function Values For Varying Values Of δ	114

CHAPTER 1

Introduction And Literature Review

Recent trends in locational research have been toward developing models which combine a locational decision with one or more other decisions. The most common examples of combined locational models are Location-Routing problems. Examples of location-routing problems include the warehouse location problem with truck dispatch, the location of bus garages, and the siting of emergency vehicles. The aim of these models is to produce not only good facility locations but also to base the decision to choose any such location on good vehicle routes. The objective functions in these models usually involve minimizing some combined disutility of locational and travel cost.

The Flow-Resource Facility Location Problem (FRFLP) studied in this dissertation is one such combined location model. The FRFLP is the problem of determining a minimum cost distribution of flow-resource facilities at the vertices of a network, and of simultaneously determining a set of paths in the network such that a maximum amount of commodities or services can be moved from a set of origins to a set of destinations. Unlike conventional facility location models, the facilities of the FRFLP are neither origins nor destinations for commodities or services moving over the network. Rather, the flow-resource facilities are located at intermediate vertices along the paths between origin and destination pairs. Any vertex could be both an origin and a destination as well as a feasible location for flow-resource facilities. The solution to the FRFLP is a distribution of flow-resource facilities at the vertices of the network and a set of paths between the origins and destinations.

A flow-resource facility in this problem is a type of facility which is used to control the loss of flow-volume along paths between origins and destinations in the network. Typically, flow-resource facilities are used in systems where the volume of commodities or services that can be moved between any two vertices decreases with increasing distance. A trade-off can then be made between the distance moved and the amount of commodities or services that can be taken. Aircraft, for example, can trade-off fuel for cargo and cargo for fuel. The flow-volume for a given route is, thus, determined by the length of the longest arc traversed. For a particular route to be feasible, flow-resource facilities may be required at the intermediate vertices. An example of a flow-resource facility is an aircraft refueler at an airbase. Aircraft can decrease the maximum length arc of a given trip by diverting to an airbase and refueling, thus, decreasing the maximum weight of fuel required for the longest arc on the route. The facilities are termed flow-resource facilities as they enhance the network as a medium for commodities or services flowing over the network, rather than acting as a generator of intervertex movement.

The routing component of the FRFLP involves selecting a set of paths between the origin and destination of each pair of vertices with a demand for the movement of commodities or services between them. Any path which passes through an intermediate vertex is viable if, and only if, a flow-resource facility with excess capacity is located at the vertex. Spatial competition arises at the vertices of the network when the number of flow-resources is limited and demand for flow-resources exceeds the available supply.

The mathematical programming model presented in this dissertation is the result of a research effort currently being undertaken at Oak Ridge National Laboratory (ORNL) on behalf of the United States Military Airlift Command (MAC). ORNL is developing a series of aircraft scheduling models designed, in part, to aid the MAC planners in designing Operational Plans (OPLANs) which become effective in the event of military contingencies. Each OPLAN consists of a demand for troops and cargo supplies to be moved from U.S. airbases to some area of operation. Large OPLANs typically have a 60 to 90 day planning horizon and could involve many thousands of troops, thousands of tons of cargo, and hundreds of airbases.

One of the more important aspects of an OPLAN design is the distribution of airlift resources throughout the airbase network and the estimates of demand placed on those resources. Such resources are required to maintain the flow of air traffic and include maintenance crews, refuelers, fuel supplies, loading vehicles, parking spaces, and flight crews. The success of an OPLAN, defined in terms of the expedient delivery of cargo and personnel, relies on the distribution of airlift resources throughout the network. Due in large part to the complexity of the problem, the MAC planners currently have no analytical framework by which they can effectively analyze the efficiency of a particular airlift resource distribution. Resource allocation is currently based on a 'rule-of-thumb' approach by which a first cut at distributing the resources is made and the effectiveness of the distribution determined by simulating the flow of the OPLAN. The simulation involves routing and scheduling of military aircraft and estimating the demand placed on the airlift resources. Changes in the distribution of airlift resources are made accordingly and the process is reiterated until a reasonable level of effectiveness is achieved. This process typically takes several days.

There is an implicit interdependence between the distribution of the airlift resources and the effectiveness of an OPLAN. Take, for example, the distribution of aircraft refuelers. A trade-off can be made between the amount of fuel carried by an aircraft and the maximum distance an aircraft can fly. The weight of the fuel required for a particular flight decreases the amount of cargo that the aircraft can carry. Thus, longer flights requiring more fuel leave less weight available for cargo than shorter flights with smaller fuel requirements. The amount of cargo that can be flown between an origin and a destination is then determined, in part, by the fuel requirement for the longest leg flown. If necessary, an aircraft can be diverted to an *enroute* airfield where more fuel can be taken onboard and the aircraft can then continue its journey. This is an important issue during the design of an OPLAN as the efficiency of the military aircraft (in terms of their cargo carrying potential) is dependent on the location of refueling points. An inefficient distribution of refuelers might increase the number of trips required to deliver the cargo for each origin-destination pair. Generally, a fixed number of aircraft are allocated to an OPLAN and each aircraft is scheduled to make multiple trips between the origins and destinations in the network. As time is a critical factor in OPLAN design, it is desirable that each aircraft make as few trips as possible; thus, a distribution of refuelers which maximizes the cargo carrying potential of the aircraft over the network as a whole should be a goal of the MAC planners.

The flight of an aircraft through a set of airbases can be modeled as a network composed of vertices (airbases) and arcs (feasible flight legs) between adjacent vertices. The trade-off between (1) the distance and duration of a single origin-to-destination trip and (2) the amount of cargo delivered to the destination by a single aircraft can be modeled using alternative paths through the network. The most efficient path from the origin to the destination might be the shortest path, the path which minimizes the maximum length leg flown, or some other path depending on specific modeled criteria such as a minimum number of vertices through which aircraft are forced to pass.

Each path evaluated between an origin and a destination must be feasible with respect to the distribution of the airlift resources over the network. For example, an aircraft cannot stop and refuel at an airfield where no refuelers are available; similarly, an aircraft cannot land at an airfield where no parking spaces are available. Thus, redistributing the airlift resources among the vertices of the network could achieve a different level of efficiency for the aircraft. More concisely, a finite set of feasible alternative paths for the aircraft are determined by the underlying airlift resource distribution. Changing the resource distribution could change the set of path options available. Thus, the most efficient path for an aircraft is determined over an optimal set of airlift resource allocations. There are, therefore, two interdependent aspects to the problem: (1) the distribution of the airlift resources at the vertices (airbases) of the network; and (2) the most efficient path for an aircraft given the underlying resource distribution. The problem becomes complex when competition for airlift resources occurs. Such competition will occur when the number of airlift resources at any single vertex is constrained and the demand exceeds the maximum supply of airlift resources.

The problem of interdependence and resource conflict can be addressed as a mathematical programming problem where the optimal solution to the problem is one which determines the best distribution of resources at the vertices of a network such that a flow-based measure of efficiency is maximized. In the case of OPLANs, the problem is that of distributing aircraft refuelers so that a maximum weight of cargo could be delivered from the set of origins to the set of destinations. The resulting mathematical programming model, termed the Flow-Resource Facility Location Problem, simultaneously solves the distribution and routing aspects of this problem.

1.1. Statement Of The Problem

The FRFLP belongs to the class of network location-routing problems and is characterized by the following:

Central control of all transportation infrastructure is exercised by a decision maker termed the *carrier*. The carrier is concerned with maximizing some measure of *benefit*.

The demand for commodities or services occurs in the form of origin-destination pairs termed *channels*.

The flow of a commodity or service over any arc in the network decreases with increasing distance.

Flow-resource facilities can be strategically placed at the vertices of the network by the carrier to support the flow of commodities or services over particular channels.

There is a cost (fixed charge) associated with locating a facility at a vertex.

Flow in the network takes place along discrete capacitated simple paths. Each path is channel specific and any such path may pass through none, one, or more flow-resource facilities.

Within the context of the FRFLP, the notion of benefit is associated with cost-efficient utilization of the transportation infrastructure. The following two examples illustrate the basic concept of the FRFLP.

EXAMPLE 1: Airlift Support: Consider an air freight carrier contracted to deliver a predetermined weight of cargo from a set of origins to a set of destinations. In an effort to maximize the profits to be gained from delivering a fixed set of channel demands, the carrier seeks to minimize the total cost associated with the delivery. The costs incurred by the carrier include fuel, crew, ground support (loading and unloading operations), and maintenance for the aircraft. It is assumed that a maximum cargo potential for each leg of a flight can be determined for each of the carrier's aircraft. The cargo potential of a flight consists of the unused capacity after the minimum required weight of fuel is taken on board to reach the next destination. To this end, the aircraft can trade fuel for cargo or cargo for fuel.

The carrier has to choose between the cargo payload of each aircraft and the distance to be flown. Heavier payloads will require more frequent stops for refueling. Conversely, lighter payloads allow a more direct path to be flown at the expense of possibly wasting aircraft cargo capacity. It is assumed that the carrier incurs a cost of prepositioning refueling resources on the network and the cost of the resources is independent of their utilization. Suppose a service crew is able to refuel up to six aircraft in a day and the carrier is forced to pay a full day's wage for every crew that is only partially used. Under these circumstances an efficient utilization would require locating the crews at a central service facility and diverting six aircraft through that location. By determining the maximum number of crews that can operate at a given airfield, the problem can be formulated as a fixed charge facility location problem.

EXAMPLE 2: Wide Area Computer Networks: Consider a wide area computer network (WAN) where there is an increasing probability of data loss during transfer as the length of a link between a pair of computers increases. The demand for data transfer between any two cities (*i.e.*, over a channel) during some time interval $[t_1,t_2]$, can be estimated in thousands of *packets* where a packet is a discrete, known number of bytes. To overcome the problems of data loss, *routers* can be located at certain cities. Routers are computers with a predetermined capacity for handling packets of data. The sole role of a router is to receive and deliver packets of data originating at other computers. Assuming high setup costs for each router (purchasing the computer and connecting it to the WAN), the FRFLP would be to locate a minimum number of routers required to minimize the probability of data loss over the network as a whole. It is assumed that a finite number of routers could be located at any one site and the number of routers that can be purchased (due to budgetary limitations) is limited.

The above examples illustrate two alternative forms of benefit. In the first example, benefit is measured as the profit to be gained from minimizing delivery cost. Benefit in the latter case is associated with minimizing the probability of data loss. Costs incurred in each case stem from the location of flow-resource facilities at the vertices of the network. The facilities in both cases are located in response to a reduction in benefit associated with increasing intervertex movement.

In essence, the FRFLP is composed of two submodels. The first submodel involves siting a finite number of capacitated facilities at the vertices of a network. The second submodel involves finding a feasible set of paths in the network such that the demand for the movement of cargo between the origins and destinations is satisfied. Each such path requires some level of support from the flow-resource facilities located at the vertices of the network. Since the pattern of flows is dependent on the location of the facilities and, reciprocally, the location of facilities is determined by the demand exerted on each vertex by the set of routes, the FRFLP involves the simultaneous optimization of both the location and the routing submodels.

1.2. Scope Of Research

The FRFLP belongs to a class of mathematical programming models termed Location-Routing Problems (LRPs). LRPs are typically extensions of median-type location models. Such models involve simultaneously locating facilities in space and evaluating routes between supply and demand vertices such that an average weighted measure of separation is minimized. These types of models are typically hard to solve. Location models involving discrete choices are known to be in the class *NP*[1] and as such, no known algorithms exist to solve them in polynomial time. Routing models involving discrete choices are also in the class *NP*. This combination of two difficult, interdependent

---

[1] For a discussion on the classes *P* and *NP* in combinatorial programming see Nemhauser and Wolsey (1988).

problems has lead to a diverse set of solution methods. These solution methods are discussed in subsection 1.3.3.

The primary aim of this research effort is to develop a mathematical statement of the FRFLP and to devise a solution procedure capable of solving large FRFLPs quickly and efficiently. In doing the research, a review was undertaken of the literature pertinent to the FRFLP. The review includes examination of literature in both pure network location problems and those which involve routing. The majority of LRPs are extensions of network median models and the FRFLP described below is related to one such model termed the multi-modal, multi-commodity generalized median model. The FRFLP model is formulated as a Mixed Integer Programming problem in Chapter 2 along with a description of preliminary concepts used in the FRFLP formulation.

The remainder of the dissertation is devoted to methods for solving the FRFLP model. Two heuristics are developed. An algorithm developed in Chapter 3, Algorithm I, utilizes a mathematical programming technique termed column generation. This algorithm is described and applied to a test problem. Chapter 4 expands on the column generation algorithm by developing a second algorithm, Algorithm II. This algorithm involves a location heuristic designed to solve the locational aspect of the FRFLP with less computational effort than is required by Algorithm I. Algorithm II is applied to the same test problem as was Algorithm I. Chapter 5 applies Algorithm II to a real world airlift problem and discusses calibrating the algorithms. Chapter 6 compares the algorithms developed in the preceding chapters, reviews the FRFLP model, and discusses future extensions to the research.

1.3. Literature Review

The development of Location-Routing models can be traced back at least twenty years although the amount of research in total has been relatively small. The majority of models which could be classified as LRPs are usually extensions of network location models where the extension consists of a routing or path subcomponent. The next sections examine the class of network location problems and the few LRPs that have been defined to date.

1.3.1. Notation and Assumptions

The models described in the next sections have the following notation and assumptions in common: The network being considered is a graph $G = \{V, A\}$ consisting of a set of vertices, $V = \{v_1, v_2, ..., v_n\}$, and a set of undirected arcs, A, each with length $c(i,j) \geq 0$. Further, we assume that at most one arc connects any two vertices, no arc starts and ends at the same vertex, and each arc is continuously rectifiable in the sense that the length $c(x,y)$ between any pair of points x and y on arc (i,j), as measured along the arc, is defined to be $c(x,y) = | c(i,x) - c(i,y) |$ (Hooker, Garfinkel, and Chen, 1988). Let d be the distance function between any two distinct points x and y on G where $d(x,y)$ is defined as the shortest path from x to y. The distance function d has the following properties for any x and y on G:

(a) $d(x,y) \geq 0$ with $d(x,y) = 0$ iff $x = y$

(b) $d(x,y) = d(y,x)$

(c) $d(x,y) \leq d(x,u) + d(u,y)$ for any $u \in V$

Let the *location vector* $X_p = \{x_1, x_2, \ldots, x_p\}$ be a set of locations, not necessarily distinct, on G for p facilities. Let G^p be a p-fold Cartesian product of G and $f(.)$ be a real valued cost function over the set G^p.

1.3.2. Network Location Problems

Network location problems typically fall into one of two classes: *medians* or *centers*. Median models minimize an average measure of spatial separation between vertices and the closest facility on the network. Such measures include minimizing the average travel time, average travel cost, or the average distance. Center problems, conversely, minimize a maximum measure of spatial separation between vertices and facilities on the network such as the maximum travel time, distance, or cost, from any vertex to the closest facility. Several excellent references deal with extensions and generalizations of the basic median and center problems on the network. Such extensions can involve probabilistic assumptions regarding network travel time and vertex demands, and/or nonlinear utility functions associated with intervertex travel: Handler and Mirchandani (1979), Tansel, Francis and Lowe (1983a, 1983b), Hooker, Garfinkel, and Chen (1988). The majority of research within the field of network location models involves the identification of sets of locations to which an optimal distribution of facilities belongs. Such sets, termed Finite Dominating Sets (FDS), help in solving complex optimization problems by restricting the solution space to an identifiable finite number of alternative location vectors.

1.3.2.1. Median Problems

A location vector, X_p, which minimizes the average (weighted) distance between the set of vertices and the facilities is termed a *p-median* of G. The objective function of the p-median problem can be stated as:

$$f(X_p) = \sum_{i=1}^{n} w_i . d(v_i, X_p) \tag{1.1}$$

where $d(v_i,X_p) = \text{Min}_{x_j \in X_p} \{d(v_i,x_j)\}$ and w_i is a nonnegative weight associated with vertex i. The optimization problem corresponding to (1.1) is to find the location vector X_p^* such that:

$$f(X_p^*) \leq f(X_p) \ \forall \ X_p \in G^P \tag{1.2}$$

Note that in (1.1) the set of facility locations is restricted to the vertices of G, that is, $X_p \subset V$. This problem is termed the *vertex restricted p-median.* When the location vector is unrestricted, that is, a facility could reside at a point along an arc, the resulting model is termed an *absolute p-median* of G. Hakimi (1964, 1965) proved that there exists an FDS which is an absolute p-median of G which consists entirely of vertices of G. This is significant as there exists a finite number of alternative solutions to (1.2) and the vertex and absolute p-median models are equivalent. When w_i is unity for all i = 1,2,...,n, the resulting problem is said to be *unweighted.*

Goldman (1969) examined the case where a vertex might be a source and/or destination for goods and services being transported across the network. These goods and services might, furthermore, require one or two stages of processing. The resulting model, termed the *two-stage generalized p-median*, can be stated as:

$$(1.3) \qquad f(X_p) = \sum_{(s,d \in V)} \mathrm{Min}_{(x_i, x_j \in X_p)} \left\{ w(s,d).d(s,x_i) + w^t(s,d).d(x_i,x_j) + w^*(s,d).d(x_j,d) \right\}$$

where $w(s,d)$, $w^t(s,d)$, and $w^*(s,d)$ are the nonnegative weights associated with source-to-facility, facility-to-facility, and facility-to-destination movements over ordered pairs of vertices (s,d), $s,d \subset V$, respectively.

Goldman proved that under the more general conditions of the two-stage model, Hakimi's result also holds, that is, there exists a location vector $X_p^* = \{x_1, x_2, \ldots, x_n\} \subset V$ such that $f(X_p^*) \leq f(X_p)$ for all $X_p \in GP$. Hakimi and Maheshwari (1972) later generalized Goldman's results to the case where, for each ordered pair of vertices (s,d), there is a set, $M(s,d)$, of distinct commodities to be shipped from s to d, and each commodity, $u(s,d) \in M(s,d)$, requires $g[u(s,d)]$ stages of processing. The resulting model is termed the *multi-commodity multi-stage generalized p-median problem.* In this model, it is assumed that each center is capable of processing any stage for any commodity. Hakimi and Maheshwari proved that an FDS exists for this model consisting only of vertices of the graph.

Several authors have defined network p-median models within an integer programming framework. Balinski (1965) was the first to formulate the absolute p-median model as a Mixed Integer Programming Problem (MIP) where there was a penalty (fixed cost) associated with locating facilities at vertices. Let x_{ij} be defined as the proportion of the demand at vertex i assigned to a facility at vertex j and let y_j be 1 if a facility is located at vertex j and 0 otherwise. Balinski's model can then be stated as:

$$\text{(1.4)} \qquad \text{Min } z = \sum_{i=1}^{n} \sum_{j=1}^{n} d(v_i,v_j).x_{ij} + \sum_{j=1}^{n} q_j.y_j$$

such that:

$$\text{(1.5)} \qquad \sum_{j=1}^{n} x_{ij} = 1 \qquad \forall\, i$$

$$\text{(1.6)} \qquad \sum_{j=1}^{n} y_j = p$$

$$\text{(1.7)} \qquad y_j - x_{ij} \geq 0 \qquad \forall\, i \text{ and } j$$

$$\text{(1.8)} \qquad 0 \leq x_{ij} \leq 1,\ y_j \in \{0,1\} \qquad \forall\, i \text{ and } j$$

where q_j is the nonnegative fixed cost for locating a facility at vertex j. Revelle and Swain (1970) presented a general form of Balinski's model as an equivalent pure Integer Programming (IP) problem under the assumption that $q_j = 0$ for all vertices $v_j = v_1, v_2, \ldots, v_n$.

This model is stated as:

$$\text{(1.9)} \qquad \text{Min } z = \sum_{i=1}^{n} \sum_{j=1}^{n} d(v_i, v_j).x_{ij}$$

such that:

$$\text{(1.10)} \qquad \sum_{j=1}^{n} x_{ij} = 1 \qquad \forall\, i$$

$$\text{(1.11)} \qquad \sum_{j=1}^{n} x_{jj} = p$$

$$\text{(1.12)} \qquad x_{jj} - x_{ij} \geq 0 \qquad \forall\, i \text{ and } j$$

$$\text{(1.13)} \qquad x_{ij} \in \{0,1\} \qquad \forall\, i \text{ and } j$$

Geoffrion and Graves (1974) define the *multi-commodity distribution design problem*. This problem is an extension of the multi-stage p-median model with the addition of capacity constraints for the facilities and all-or-nothing assignments of demand nodes to facilities. An interesting extension of the Geoffrion model is to allow customers to be allocated directly to a supplier of a particular commodity when such an assignment is cost effective. The authors formulate the problem as a fixed charge MIP model and solve it using Benders' Decomposition. The authors applied the model to a medium-sized real world distribution problem.

1.3.2.2. Center Problems

A location vector, X_p, which minimizes the maximum (weighted) distance between any vertex and the closest facility is termed a *p-center* of G. The objective function of the p-center problem can be stated as:

$$f(X_p) = \mathrm{Max}_{i \in V} \{w_i . d(v_i, X_p)\} \tag{1.14}$$

where $d(v_i,X_p) = \mathrm{Min}_{x_j \in X_p} \{d(v_i,x_j)\}$ and w_i is a nonnegative weight associated with vertex i. The optimization problem corresponding to (1.14) is to find the location vector X_p^* which satisfies (1.2). Note that unlike the p-median problems in subsection 1.3.2.1, the facilities in the p-center problem are not restricted to the vertices of G. That is, the FDSs corresponding to the network p-median and network p-center problems are not, in the general case, equivalent. Any location vector X_p^* satisfying (1.2) where $X_p^* \in G$ is termed an *absolute p-center* of G. The case where the location vector is restricted to the vertices of G is termed the *vertex restricted p-center* of G. When w_i is unity for all i = 1,2,...,n the problem is said to be *unweighted.*

Hakimi (1964) defined the *absolute 1-center weighted problem* in which the location vector consists of a single center. For the absolute p-center problem, Minieka (1970) and Garfinkel, Neebe, and Rao (1977) proved that an FDS consists of the set of node locations augmented by the set of *center bottleneck points*. The set B_c of center bottleneck points is defined as follows: A point x on an arc (u,w) is in B_c when for two distinct vertices v_i and v_j:

(1.15) $$w_i.d(v_i,x) = w_j.d(v_j,x)$$

and $d(v_i,x)$ and $d(v_j,x)$ do not both decrease when x is moved slightly in either direction (Hooker *et al.* 1988). The optimal location vector which satisfies $f(X_p^*) \leq f(X_p)$ for all $X_p \in GP$ must be in the set $V \cup B_c$.

1.3.2.3. Other Network Location Problems

Halpern (1978) defined the network *cent-dian* problem. This problem is a combination of both the network p-median and the network p-center problem. The cost function of the cent-dian (or medi-center model as it is also known) is a convex combination of the median cost function and the center cost function (Hooker *et al.* 1988). The objective function of the cent-dian problem can be stated as follows:

(1.16) $$f(X_p) = (1-\gamma) \sum_{i=1}^{n} w_i.d(v_i,X_p) + \gamma.\mathrm{Max}_{i \in V} \{ w_i.d(v_i,X_p) \}$$

where $d(v_i,X_p) = \mathrm{Min}_{x_j \in X_p} \{d(v_i,x_j)\}$, w_i is a nonnegative weight associated with vertex i, and $0 \leq \gamma \leq 1$. Halpern proved that an FDS consisting of the set of vertices augmented by the set of center bottleneck points exists for the 1-facility cent-dian problem (Hooker *et al.* 1988).

1.3.3. Location-Routing Problems

The literature pertinent to Location-Routing problems (LRPs) can be classified into three broad groups: (1) research aimed at providing optimal solutions to typically smaller, hypothetical problems; (2) research aimed at developing heuristic procedures to solve generally larger applied problems; and (3) research where LRPs are solved as an element within a larger scope of research. These three groups are treated independently below.

1.3.3.1. Optimal Models

Lawrence and Pengilly (1969) were among the first to examine the interdependence between facility location problems and vehicle routing in the context of the Hamiltonian Location Problem (HLP). The HLP involves selecting the best vertex to start a set of m traveling salesman tours to service n vertices. The authors address the problem by solving n m-traveling salesman problems (m-TSPs). The Hamiltonian Location Problem has been more fully researched by Laporte and Nobert (1981) who developed an exact algorithm to simultaneously solve the locational and the routing sub-problems. Laporte's algorithm consists of defining an LP relaxation of the problem which is solved using branch-bound techniques. Subtour elimination constraints are added to the LP as necessary. Using this approach the authors optimally solved models for $20 \leq n \leq 50$ and $2 \leq m \leq 5$ in reasonable time.

Laporte, Nobert, and Pelletier (1983) report an exact algorithm for solving general LRPs. Several LRP formulations are developed by the authors, all of which are variations on the HLP. Among these generalizations are fixed costs for depot locations and multiple visits to a vertex by more than one salesman. These generalized problems are all solved using the same technique: the problems are formulated as IPs and solved using branch-bound strategies and the selective introduction of Gomory Cuts[2].

In a later paper, Laporte, Nobert, and Arpin (1984) describe an exact algorithm for solving general LRPs with multiple side constraints. The side constraints involve capacitated vehicles, lower and upper bounds on the number of depots and the number of vehicles at any depot, and fixed costs for both depots and vehicles. In this algorithm, the solution method utilizes a tree-search procedure where LP relaxations are solved at each node of the tree and subtour breaking constraints and 'chain barring' constraints are added as necessary. A chain barring constraint is one which requires that a particular route starts and ends at the same depot. Invalid chains, such as those involving two vertices or those which start and end at different depots, are removed by adding a cut to the LP relaxation.

1.3.3.2. Heuristic Models

A variety of heuristic approaches have been designed to solve LRPs, the majority of which are applications-oriented. Jacobsen and Madsen (1980) report three heuristics designed to solve a two-stage LRP. This particular application was to locate distribution points and design vehicle routes for a newspaper delivery service. This research involved a two-stage routing model where, at one level vehicles leave the centralized (predetermined)

---

[2] A *cut* as applied to an integer or mixed-integer linear program is a constraint which, when added to the problem, has the following properties. (1) The constraint cuts off the optimal LP solution. (2) The constraint does not remove any feasible integral solutions. (3) After possibly many cuts, the last constraint makes the integer optimum an extreme point. R. Gomory (1958) was the first to introduce the cutting-plane approach for pure integer programs. For more information regarding cutting-plane approaches, the reader is directed to Nemhauser and Wolsey (1988) Chapter II Section 4 and to Minoux (1986) Chapter 7 Section 3.

printing office and drop loads of newspapers at distribution points (to be determined). The second level routing problem involves developing vehicle routes to pick up newspapers at the delivery points and distribute them to sales outlets which are also predetermined. In this application, the location and demand of the sales outlets are known, as is the location and supply of the printing office. The problem was to locate a set of intermediate distribution points, estimate the demand for newspapers at each such point (where the demand exerted at such a distribution point is the sum of the demand at all sales outlets allocated to that point) and develop a set of primary (printing office to distribution points) and secondary (distribution points to sales outlets) vehicle routes such that a minimum transportation cost is incurred. Vehicle capacity and timing constraints added to the complexity of this problem.

To solve the two-stage LRP, Jacobsen and Madsen developed three heuristics: a 'Tree-Tour' method, an 'Allocation-Savings' method and a 'Savings-Drop' method. The Tree-Tour heuristic was based on the observation that routing patterns resembled spanning trees in sub-networks of the original graph. A 'greedy' heuristic was developed which sequentially added nodes and arcs into the solution based on the principle of maintaining spanning trees. Vehicle routes were then constructed by adding the arcs required to complete cycles and so create viable circular tours.

The Allocation-Savings heuristic was a three stage approach involving the 'Savings' route construction heuristic originally developed by Clarke and Wright (1964). The three stages are as follows:

Stage 1: A prespecified number of distribution points are located using a median type location-allocation model. In this stage, sales outlets are allocated to the distribution points.

Stage 2: A modified savings algorithm is used to construct secondary tours starting from the distribution points located in stage 1.

Stage 3: Primary tours are formed using the Savings algorithm.

The distribution points were located using a median type model in rectilinear space. The procedure terminated early if the solution became infeasible.

The Savings-Drop heuristic was also developed using a three stage approach. As in the Allocation-Savings Heuristic, a modified Clarke-Wright algorithm is used to develop vehicle routes. However, the distribution points are located using the DROP method developed by Feldman, Lehrer, and Ray (1966). The three stages are as follows:

Stage 1: Secondary tours are formed using the savings heuristic, with the printing office as an origin for the vehicles.

Stage 2: Distribution points are located at the sales outlets corresponding to the first sales outlet on each primary tour. The DROP algorithm is then used to reduce the number of distribution points.

Stage 3: Primary tours are developed using the savings algorithm.

The study showed that the three stage methods were superior to the Tree-Tour method and that the multi-stage methods produced solutions which were comparable to each other.

Perl and Daskin (1985) published what is perhaps the most complete approach to LRPs. The authors present a mixed integer programming (MIP) formulation for the Warehouse Location-Routing Problem (WLRP). This problem involves simultaneously determining the number and location of up to m facilities on a network, the allocation of n demand nodes to these facilities, and the optimal routing of k vehicles to service the demands. The problem formulation involves constraining the maximum trip length of each vehicle, the maximum capacity of each vehicle, and the service capacity of the facilities. Recognizing the computational intractability of the problem, the authors develop a three stage iterative heuristic to solve the problem.

The complete WLRP is decomposed into two subproblems; a location problem (Stage 1) and a routing-allocation problem (Stage 2). An initial stage (Stage 0) involves solving a relaxation of the problem to generate an initial feasible solution.

Stage 0: An relaxation of the WLRP is solved assuming that all potential facility sites are used. This step then constructs an initial set of feasible routes which minimizes the delivery costs.

Stage 1: Warehouses are relocated based on route segments developed from the previous stage.

Stage 2: Given new depot locations from stage 1, demand nodes are reallocated to facilities and new routes are constructed.

An iterative procedure is used which solves stage 0 to generate an initial feasible solution, then alternately solves stage 1 and stage 2 until no epsilon change in the objective function value is observed. The solution technique was applied to a regional warehouse system involving 318 demand nodes and 4 depot locations.

1.3.3.3. Other Location-Routing Models

All LRP models described above involve siting one or more facilities as distribution centers. There has been little development of LRPs where this is not the case. This subsection introduces three disparate LRPs which involve siting facilities which are not simply origins or destinations on the network. Levy (1987) presented the *walking-line-of-travel problem* (WLTP) which is a combined location and arc-routing problem. Unlike conventional LRPs which typically have multiply-embedded TSPs, the WLTP involves solving multiply-embedded Chinese Postman problems[3]. The WLTP is based on a problem faced by the U.S. Postal Service. Essentially, the WLTP involves defining the daily work schedules of postal carriers. Each work schedule involves driving from a central depot to an intersection in a predesignated distribution area. The postal carrier must then leave the vehicle at the intersection and walk a sequence of roads, deliver mail, and return to the vehicle when finished. Upon returning to the vehicle the carrier could select a new sequence of roads to traverse or could reposition his vehicle at a new intersection. Each

[3] Chinese Postman Problems are arc-routing models where the objective is to define a tour over the arcs of a network such that each arc is traversed at least once. An optimal chinese postman tour is one which minimizes the total distance traveled.

walking tour should be about one hour in duration and a daily work schedule should be no longer than five hours in duration. The primary objective of the WLTP is to minimize the number of carriers required to deliver all the mail. Secondary objectives in the problem involve minimizing the difference in time taken to complete any daily work schedule. To solve the problem, Levy developed two specialized heuristics based on set partitioning algorithms. Both of them provided similar and satisfactory results.

Montreuil and Ratliff (1988) examined the location of Input/Output stations within an industrial facility. The aim of the model is to maximize the efficiency of flow between spatially separated production processes on a factory floor. Examples of such Input/Output facilities are doors and loading bays where loads enter or leave a work area. The objective in this study is to minimize the weighted sum of rectilinear distances between all O-D pairs within the facility. The problem is formulated as an IP multi-facility location problem and solved using a specialized network heuristic.

Hodgson (1981) developed a location model based on the generalized p-median model (Goldman 1969) for determining the optimal set of locations of facilities intermediate to the journey to work. In this instance, child care facilities were sited at locations which minimized the weighted sum of trip deviation. Hodgson developed a simple transformation which enabled the model to be solved as a weighted p-median using a node substitution heuristic after Tietz and Bart (1968).

The FRFLP is related to the generalized p-median formulations discussed in subsection 1.3.2.1. In particular, the FRFLP bears closest resemblance to the two-stage generalized p-median formulation by Goldman (1969) and the multi-stage multi-commodity generalized p-median formulation by Hakimi and Maheshwari (1972) for the following reasons.

The demands for movement in the network occur between origins and destinations rather than between facilities and demand vertices.

The FDSs corresponding to the generalized p-median problems consist entirely of the set of vertices of the graph. The facilities in the FRFLP are assumed to be located at the vertices of the network. Thus, the FDS of the FRFLP is also restricted to the set of vertices.

Facilities are being located at central sites (medians) such that the flow of commodities or services moving between many origin-destination pairs can be processed through centralized facilities.

The algorithms designed to solve the FRFLP are based on the iterative heuristics used by many of the authors for solving difficult location-routing problems. In particular, the algorithms developed in Chapters 3 and 4 bear closest resemblance to the heuristic designed by Perl and Daskin (1985) for solving the warehouse location-routing problem. Perl and Daskin's algorithm maintains feasibility at every stage and iteratively improves the current solution by alternately solving a locational submodel followed by a routing model. This is similar to the approach which is implicit in the column generation algorithms. In essence, the FRFLP is decomposed into a restricted master problem which solves the locational component, and a column generating subproblem which solves the routing aspects of the FRFLP.

1.4. Summary Of Chapter 1

This chapter introduced the background and motivation for the development of the Flow-Resource Facility Location Problem. The problem was stated and two examples given to illustrate the concepts underlying the FRFLP. The first example was an airlift planning problem where airlift facilities were located on a network consisting of airbases and flight legs such that a maximum volume of flow in the network was achieved. The second example consisted of a wide-area computer network where routing computers were being located to minimize the probability of data loss during the transmission of data from one computer to another. In order to operationalize the FRFLP model, it must be stated in a form which can be solved using mathematical programming techniques. The following chapter presents such a formulation and discusses solution procedures.

CHAPTER 2

Problem Formulation

This chapter presents the Flow-Resource Facility Location Problem as a mixed integer programming problem. First, the assumptions underlying the FRFLP are defined and the concepts required to solve the FRFLP as an MIP are introduced. In Section 2.2 the FRFLP is formulated as an MIP and the additional notation is described. Section 2.3 offers a general discussion on the formulation and describes the problems inherent in solving the problem by conventional methods.

2.1. Preliminaries

This section defines the assumptions and preliminary concepts required such that the FRFLP model can be formulated as a mathematical programming model. In subsection 2.1.1 the assumptions inherent in the statement of the FRFLP are discussed. The optimality conditions of the FRFLP are described in subsection 2.1.2 while the concept of path capability is described in subsection 2.1.3.

2.1.1. Model Assumptions

This section outlines the assumptions implicit in the FRFLP model. The assumptions can be categorized into three groups: those pertaining to the flow-resource facilities being located on the network, those pertaining to the supplies and demands for commodities or services in the network, and those pertaining to the allocation of flow capacity to the network.

2.1.1.1. Flow-Resource Facilities

The flow-resource facilities being located on the network are assumed to have the following attributes.

> Each flow-resource facility incurs a non-negative fixed-charge (in dollars) when it is located on the network.
>
> The cost of each flow-resource facility being located on the network is independent of the usage of the facility.
>
> Flow-resource facilities can only be located at the vertices of the network.
>
> Flow-resource facilities being located at a vertex are assumed to be homogeneous. The characteristics of flow-resource facilities may vary between vertices.

An upper bound exists on the number of flow-resource facilities that can be located at any vertex. The upper bound may be infinity.

Each flow-resource facility provides a unit level of service for each path utilizing the facility.

The allocation of flow-resources on the network is assumed to incur a nonnegative cost. The FRFLP is stated as a benefit maximization problem; thus, the total sum of location costs detract from the total revenue to be gained by the carrier. Further, as the cost of each facility being located is independent of the usage of that facility, excess facilities located on the network will incur a cost over and above the minimum cost required to make the solution feasible. Thus, the number of flow-resource facilities located on the network should be minimized.

Flow-resources can only be located at the vertices of the network, thus, the FRFLP is assumed to be vertex-restricted. Hence, a feasible location vector will consist entirely of a set of vertices of the graph.

Flow-resource facilities being located at a vertex are assumed to be homogeneous with respect to each vertex. However, the characteristics of flow-resource facilities may vary between vertices. Thus, the fixed charge for allocating a flow-resource facility to a vertex and the service capacity of the flow-resource facility may vary between vertices but will remain constant for all facilities at any single vertex. The number of flow-resource facilities that could be located at a single vertex is assumed to be limited by an upper bound. This upper bound must be nonnegative and could be infinite. If an upper-bound on the number of flow-resource facilities at a vertex is infinite, the vertex is assumed to be uncapacitated. A finite, nonnegative number of flow-resource facilities are assumed to be available for distribution.

Lastly, each flow-resource facility is expected to provide a unit level of service to each path utilizing the facility. Thus, the service capacity of a single flow-resource facility is assumed to be nonnegative, integer, and finite. For example, if the service capacity of a flow-resource facility located at vertex k is six and only one flow-resource facility is located at vertex k, then at most six feasible paths may have vertex k as an intermediate vertex.

2.1.1.2. Supply And Demand

The assumptions pertaining to the supply and demand of commodities or services within the FRFLP are as follows.

> The supply and demand for commodities or services on the network occur between pairs of distinct vertices. The supply and demand associated with any such pair is termed a *channel demand.*
>
> Channel demand is inelastic with respect to the cost of transporting commodities or services between the origin and the destination vertices of channels.

The channel demand is assumed to be nonnegative and inelastic with respect to the cost of transporting commodities or services between the origin and destination. Thus, the demand for movement remains constant although the supply of commodities or services, in terms of the capacity allocated in the network for the channel, will vary depending on the spatial competition for flow-resources at the vertices of the network from other channels.

2.1.1.3. Network Capacity

The assumptions pertaining to the allocation of network capacity are as follows.

Network capacity is allocated in the form of simple paths between the origin and destination of a particular channel.

There is a benefit related to the potential revenue associated with the path capability of each path.

The volume of commodities or services that can be moved over any arc in the network is constant and does not vary between channels.

A path requires a unit of service from a flow-resource facility at each intermediate vertex.

Any unique path between the origin and destination of a channel may be used more than once if such a solution is desirable.

The set of paths in the solution should be optimal with respect to the location of facilities on the network.

In a feasible solution to the FRFLP, each channel will be allocated a set of paths. (The set of paths could possibly be equal to the empty set.) Each path in the solution has a nonnegative maximum capacity, termed the *path capability*, for the movement of commodities or services over it. Thus, the *network capacity* allocated to any particular channel is equal to the sum of the capabilities of the paths associated with the channel in the feasible solution. Any path allocated to a channel decreases the channel's demand by an amount equal to the path capability. The remaining channel demand, termed the *residual demand*, is unrestricted.

The volume of commodities or services that can move over any arc in the network is assumed to be constant and finite, regardless of the particular channel. Each arc is then assumed capacitized with a nonnegative upper bound. An implicit lower bound of zero is also associated with each arc. A path using a sequence of such arcs has a capability equal to the capacity of the constraining arc on the path. This is dealt with more explicitly in subsection 2.1.3.

Each path in the solution requires a level of support from the flow-resource facilities on the network. A simple path consists of a sequence of r vertices and (r - 1) arcs. It has a single origin, a single destination and (r - 2) intermediate vertices. At each such intermediate vertex the path requires a unit of service from a flow-resource facility located at that vertex. Thus, for any such path to be feasible, at least (r - 2) flow-resource facilities are required to be located on the network. A path in the network for a particular channel may be used multiple times if such a solution is optimal. Thus, cost-efficient paths may be represented more than once in the optimal solution. However, each instance of a path in the optimal solution requires a unit of support from a flow-resource facility at each intermediate vertex.

Lastly, the set of paths in the solution should be optimal with respect to the distribution of flow-resource facilities on the network. Similarly, the distribution of flow-resources should be optimal with respect to the set of paths in the final solution. This is discussed more fully in the following section.

2.1.2. Optimality Conditions

From the statement of the FRFLP in Chapter 1 and the assumptions discussed in the preceding subsection, the FRFLP model can be seen to consist of two optimization problems: (1) given a set of paths in the network, find a minimum cost location vector of flow-resource facilities such that the set of paths is feasible; and (2) given a location vector of flow-resource facilities, find a set of paths such that the total revenue associated with the network capacity is maximized. A solution to the former problem is trivial: The total number of units of service required at a vertex k is the number of paths in the solution with vertex k as an intermediate vertex on the path. Determine this for all $k \in V$. If the minimum number of flow-resource facilities required to satisfy the service needed at any vertex k exceeds the maximum number of flow-resource facilities that can be allocated at k, or if the minimum number of flow-resource facilities required on the network as a whole is greater than the maximum number of facilities available, then the set of paths is infeasible for the FRFLP.

Conversely, the latter problem, that of creating paths in the network with respect to a location vector such that a measure of benefit is maximized, is a much more difficult problem. Each path in the solution set must be not only feasible with respect to the location vector but also optimal with respect to other paths in the solution. That is, there must exist no alternative paths in the network which could be placed into the solution set such that the total benefit associated with the solution could be increased.

To this end, the FRFLP can be seen to consist of two subproblems each of which is dependent on the other for a globally optimal solution. An optimal location vector is defined with respect to a set of paths, and conversely, an optimal set of paths is defined only in the context of a location vector. Thus, a globally optimal solution to FRFLP must involve the simultaneous optimization of both subproblems. That is, a solution to FRFLP is optimal if and only if both the distribution of flow-resource facilities and the set of paths in the solution are simultaneously chosen such that the benefit associated with the resulting network capacity is maximized. The next subsection examines the concept of path capability and how path capability is linked to benefit maximization.

2.1.3. Path Capability

The FRFLP model uses a concept of *path capability* to determine the maximum potential benefit that could be associated with any given path in the network. The capability of any path p in G is determined as a function of the maximum capacity of the constraining arc on the path. Let p be a simple path in G starting from some vertex $v_i \in V$, and ending at some vertex $v_j \in V$, $i \neq j$. Without loss of generality we assume $i < j$. The path consists of a sequence of arcs and vertices $v_i,(v_i,v_{i+1}),v_{i+1},...,v_{j-1},(v_{j-1},v_j),v_j$ in which each vertex of V appears at most once. The path capability, C_p, of p can be defined as:

$$C_p = \mathrm{Min}_{(x,y)\in p}\{k(x,y)\} \tag{2.1}$$

where k(x,y) is the maximum capacity of arc (x,y). C_p is thus defined as the capacity of the constraining arc in the path. The benefit of path p, in terms of the FRFLP model, is then a function of C_p. Let b_p^m be the maximum potential benefit for path p, then $b_p^m = R_{ij}.C_p$ where R_{ij} is the revenue in dollars per unit of capacity associated with any path between the two distinct vertices v_i and v_j. The actual benefit of path p is dependent on the costs associated with making p feasible with respect to the location of facilities on the network.

Within the FRFLP, cost is associated with the location of facilities at the vertices of the network. That is, the cost, y, of a set of paths, P, is determined as the sum of all facility costs (fixed charges) incurred. For k = 1,2,...,n, let l_k^* be the minimum number of facilities located at vertex k such that P is feasible. If the cost of each facility located at vertex k is given as y_k then the cost of P can be calculated as:

$$y = \sum_{v_k \in V} y_k . l_k^* \tag{2.2}$$

For the network as a whole, the total benefit can be increased by either decreasing the cost associated with the location vector, or by increasing the network capacity in G by redefining the location vector and finding a more efficient set of paths.

2.2. Formulation

This section formulates the FRFLP model as an MIP. Subsection 2.2.1 describes the additional notation associated with the model and subsection 2.2.2 actually describes the model.

2.2.1. Additional Notation

Let W_{ij} be the demand for commodities or services between two distinct vertices v_i and v_j in V. Any pair of vertices v_i, v_j, where $W_{ij} > 0$ is termed a channel and is written [i,j]. Let P = {all simple paths in G}. For any channel [i,j], let P_{ij} be the subset of P consisting of all feasible paths from v_i to v_j in G. Define P_k as the subset of P consisting of all paths where vertex k is an intermediate vertex in the path.

The decision variables are defined as follows:

Z_p = the number of times path p is used.

L_k = the number of homogeneous facilities located at vertex k.

S_{ij} = the amount of overassigned capacity for channel [i,j].

The remaining notation is as follows:

Y_k = the fixed cost in dollars for opening a facility at vertex k.

Q_k = the maximum number of paths that can be serviced by a facility at vertex k.

M_k = the maximum number of facilities that can be located at vertex k.

M = the maximum number of facilities that can be located on V.

All other definitions are as previously defined.

2.2.2. MIP Formulation

The FRFLP model can be formulated as below:

$$(2.3) \quad \text{Max } z = \sum_{i}^{n} \sum_{j}^{n} \sum_{p \in P_{ij}} R_{ij}.C_p.Z_p - \left(\sum_{v_k \in V} Y_k.L_k + \sum_{i}^{n} \sum_{j}^{n} R_{ij}.S_{ij} \right)$$

such that:

$$(2.4) \quad \sum_{p \in P_k} Z_p - Q_k.L_k \leq 0 \qquad \forall\ k = 1,2,\ldots,n.$$

$$(2.5) \quad \sum_{p \in P_{ij}} C_p.Z_p - S_{ij} \leq W_{ij} \qquad \forall\ [i,j].$$

(2.6) $L_k \leq M_k$ $\forall\ k = 1,2,\ldots,n.$

(2.7) $\sum_{v_k \in V} L_k \leq M$

(2.8) $Z_p \geq 0$, integer $\forall\ p \in P.$

(2.9) $L_k \geq 0$, integer $\forall\ k = 1,2,\ldots,n.$

(2.10) $S_{ij} \geq 0$ $\forall\ [i,j].$

The objective function (2.3) consists of three terms. The first term seeks to maximize the benefit associated with the volume of flow in the network. The second term in (2.3) levys the setup costs, or fixed charges, in terms of the number and location of facilities on the network. The third term in (2.3) penalizes over-assigning capacity to each channel. For some channel [i,j], there may exist no combination of paths in the set P_{ij} that matches the demand, W_{ij}, exactly. To this end, there is a tradeoff between the amount of demand not satisfied and the amount of excess capacity allocated to the channel. Thus, the model should strive to select a set of paths from P_{ij} which most closely matches the demand, W_{ij}, subject to the competing demand for facilities from other channels.

The first constraint set (2.4) ensures that the number of paths passing through the k^{th} vertex is no greater than the amount of capacity allocated to the vertex. Constraint set (2.5) ensures that the demand for capacity over each channel is accounted for. Constraint sets (2.6) and (2.7) are generalized upper bounds on the number of facilities allocated to each vertex and the total number of facilities located on the network respectively. (2.8)-(2.10) provide optimality conditions on the variables.

2.3. Solution Strategies For The FRFLP

The FRFLP model can potentially be solved using general purpose integer programming techniques such as branch-bound or enumeration methods. However, the formulation presented above requires enumerating the sets P, P_{ij} for all [i,j], and P_k for all $k = 1,2,\ldots,n$. Consider a fully connected network consisting of n vertices. The number of arcs in the network is given as n(n - 1). With each channel [i,j] there is associated an origin, v_i, a destination v_j, and a subnetwork of (n - 2) intermediate vertices. Any path p leaving the origin v_i and entering the subnetwork can do so at any of the (n - 2) intermediate vertices. Similarly, the path must leave the subnetwork at any of the (n - 3) remaining vertices. The number of alternative paths between v_i and v_j is equal to the number of TSP tours over all possible subsets of the (n - 2) intermediate vertices. For some network consisting of n vertices the number of TSP tours can be derived as:

$$(n-1)! = (n-1) \times (n-2) \times (n-3) \times \ldots \times 1 \qquad (2.11)$$

as starting at some vertex k, there are (n - 1) vertices to choose to visit next. At vertex k + 1 there are now (n - 2) choices to visit. At the k + 2 vertex there are (n - 3) remaining choices. This pattern is repeated until exactly one vertex remains to be visited. For the FRFLP formulation, the number of alternative paths over any defined subset of the intermediate vertices is k! where k is the cardinality of the set of vertices in the subnetwork, as there are k choices for the starting vertex.

For some channel [i,j] the number of alternative, unique subsets of intermediate vertices, including the empty set, is given as:

$$(2.12) \qquad \binom{n-2}{1} + \binom{n-2}{2} + \ldots + \binom{n-2}{n-3} + \binom{n-2}{n-2} + 1$$

where the notation $\binom{n}{r}$ (spoken n *choose* r) is calculated as $\frac{(n)!}{(r)!(n-r)!}$ From this we can derive the number of unique paths in the set P_{ij}. We know from the above that the number of possible paths in a subnetwork consisting of k vertices is given as k!. Thus, the number of unique paths in G for each channel [i,j] is given as:

$$(2.13) \quad |P_{ij}| = \binom{n-2}{1}(1)! + \binom{n-2}{2}(2)! + \ldots + \binom{n-2}{n-3}(n-3)! + \binom{n-2}{n-2}(n-2)! + 1$$

where the notation |s| denotes the cardinality of set s. Note, the last term (+ 1) is due to the direct path from v_i to v_j. From (2.12) we can calculate the cardinality of set P, the set of all simple paths in G. We know that there are n(n - 1) possible channels in G and there are $|P_{ij}|$ possible paths per channel. The total number of simple paths is then:

$$(2.14) \qquad |P| = n(n-1) \times |P_{ij}|$$

which is a large number for even very small networks. Consider a fully connected network where n = 6. For some channel [i,j] there are 65 unique paths, that is $|P_{ij}| = 65$ and the total number of simple paths, |P|, in G is 1950.

If the FRFLP were going to be solved using branch-bound methods such as those available in commercial solvers[4], the integrality restrictions on the constraint sets (2.8) and (2.9) would have to be relaxed. Furthermore, as each path can potentially be used more than once, IP techniques for modeling general integer variables as vectors of binary variables would have to be used. This necessitates the calculation of a reasonable upper bound for each such general integer variable. If some general integer variable Z_p can take on the values $0,1,2,\ldots,Z_p^u$ where Z_p^u is a valid upper bound on Z_p, then define l such that:

(2.15) $$2^{l-1} \leq Z_p^u < 2^l$$

The general integer variable Z_p can then be represented as:

(2.16) $$Z_p = Z_{p1} + 2(Z_{p2}) + 4(Z_{p3}) + 8(Z_{p4}) + \ldots + 2^{l-2}(Z_{pl-1}) + ((Z_p^u\text{-}2)_{l-1}Z_{pl}$$

where each Z_{pi} is a binary variable (Marsten 1988).

This poses a problem of determining a valid upper bound Z_p^u for each Z_p. The method illustrated above for emulating general integer variables using a vector of binary variables is dependent on being able to supply a good upper bound for each such variable. Two cases must be considered. The first case occurs when the number of intermediate vertices on path p is zero, that is, p is a direct path from v_i to v_j. In this case, let Z_p^u be the maximum number of times path p would be used where:

---

[4] Two commonly used Linear and Discrete optimization packages available are XA developed by Sunset Software, 1613 Chelsea Rd, Suite 153, San Marino CA 91108 and ZOOM developed by XMP/ZOOM Software Inc. P.O. Box 58119, Tucson AR 85732.

(2.17) $$Z_p^u = \left\lceil \frac{W_{ij}}{C_p} \right\rceil$$

and W_{ij} and C_p are the channel demand and the path capability as previously defined.

The second case arises where one or more facility sites are incident with path p. In such cases, Z_p^u can be bound by the minimum service potential over all the facility vertices on p. Let K_p be the set of all facility vertices incident on path p, then

(2.18) $$Z_p^u = \mathrm{Min}_{v_k \in K_p} \{M_k . Q_k\}$$

where M_k and Q_k are the maximum number of facilities available to be located at vertex k and the service potential of each such facility as previously defined.

It can be seen from the above discussion that attempting to solve a FRFLP using general-purpose integer programming techniques would be prohibitively expensive in terms of the computer resources required. A full statement of even trivial problems would involve thousands of general integer variables which is beyond the scope of all common commercial software currently available. To this end, specialized solution techniques should be devised to solve the FRFLP problem. Chapters 3 and 4 develop two algorithms which use the structure of FRFLP to more easily solve the combinatorial problem.

2.4. Summary Of Chapter 2

This chapter presented the FRFLP formulation as a Mixed Integer Programming problem. The assumptions and concepts underlying the development of the FRFLP as an MIP were stated and illustrated. The mathematical statement of the FRFLP was followed by a discussion on the tractability of solving the FRFLP using conventional techniques such as general purpose MIP solvers. The discussion showed that the potential number of variables prohibits the use of conventional techniques and introduced the need for a specialized algorithm to solve the model. To this end, a heuristic is developed in the following chapter which exploits some of the characteristics of the FRFLP model.

CHAPTER 3

Algorithm I: Column Generation

This chapter presents a mathematical programming technique termed column generation to solve the FRFLP model heuristically. Section 3.1 describes the column generation procedure as it applies to pure linear programming problems. Section 3.2 describes the adaptations of the column generation technique designed to accommodate the MIP formulation of FRFLP. Section 3.3 shows an example of Algorithm I applied to a seven vertex test problem.

3.1. Column Generation In Linear Programs

The column generation technique, originally introduced by Gilmore and Gomory (1963) for the cutting stock problem, is a decomposition procedure which is frequently used to solve large scale linear programs. We assume the LP to be solved has the following structure:

(3.1) $$\text{Max } z = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

such that:

(3.2) $$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \le b_i \qquad \forall\ i = 1,2,\ldots,m$$

(3.3) $$x_j \ge 0 \qquad \forall\ j = 1,2,\ldots,n$$

where:

n = number of variables.

m = number of constraints.

A = m x n matrix of structural coefficients (a_{ij}).

c = row vector of costs, $\mathbf{c} = \{c_1, c_2, \ldots, c_n\}$.

b = column vector of right hand sides, $b = \{b_1, b_2, \ldots, b_m\}^T$.

z = real valued objective function.

Column generation methods are best applied to systems of type (3.1)-(3.3) where the number of variables, n, is much larger than the number of constraints, (n » m), and the full statement of (3.1) is too large to be stated in matrix format. However, we assume that the coefficient matrix, **A**, is known explicitly; that is, the number of columns of **A** is finite and we could state column A_j for any $j = 1,2,\ldots,n$.

Because of the large number of variables, direct solution of (3.1), termed the *master problem*, by the simplex method is inappropriate. Instead, the column generation procedure is designed to generate columns A_j only as needed. If we assume that some variables $x_{J+1}, x_{J+2},..., x_n$ are non-basic with value 0 then we can then define a new problem:

(3.4) $$\text{Max } z^J = c_1x_1 + c_2x_2 + ... + c_Jx_J$$

such that:

(3.5) $$a_{i1}x_1 + a_{i2}x_2 + ... + a_{iJ}x_J \leq b_i \qquad \forall\ i = 1,2,...,m$$

(3.6) $$x_j \geq 0 \qquad \forall\ j = 1,2,...,J$$

Problem (3.4)-(3.6) is termed a *restricted master problem* as we are concerned with maximizing the objective function value, z^J, over a subset of the variables in the master problem. An optimal solution to (3.4) with $x_{J+1} = x_{J+2} = ... = x_n = 0$ is feasible for (3.1) and, as such, potentially optimal for the master problem. Let π^J be the vector of shadow prices, $\pi^J = \{\pi_1,\pi_2,...,\pi_m\}$, associated with the optimal solution of (3.4). We assume that there exists an algorithm, called a *generating algorithm* which, given π^J, efficiently finds column s such that:

(3.7) $$\underline{c}_s = \text{Max}_{(j=J+1,J+2,...,n)} \left\{ c_j - \pi^J.A_j \right\}$$

where $\underline{c}_s$ is the reduced cost associated with column s. Using the complementary slackness conditions of the simplex method, we know that in maximization problems, if the reduced cost of any variable, x_s, is strictly positive, then the variable can enter the basis. Thus, we would like to generate columns, A_s, which have positive reduced costs as we know each such column will enter the basis and improve the objective function. Equality (3.7), termed the *subproblem*, is an optimization problem in itself. Typically, the details of the subproblem depend on the structural characteristics of the master problem being studied and can be any type of optimization problem including nonlinear, dynamic, and integer programming problems (Bradley, Hax, and Magnanti 1977, p 540).

Given an algorithm which solves (3.7), the column generation algorithm can be stated as an iterative procedure which alternately solves problems (3.4) and (3.7) until an optimal solution is found. The algorithm is stated as:

STEP 1: Solve the restricted master problem, (3.4)-(3.6), using the simplex method. Let $\pi^J = \{\pi_1, \pi_2, \ldots, \pi_m\}$ be the optimal shadow prices to (3.4). Go to step 2.

STEP 2: Solve the subproblem (3.7). If $\underline{c}_s < 0$ stop: z^J is optimal for (3.1) with $x_{J+1} = x_{J+2} = \ldots = x_n = 0$. If $\underline{c}_s \geq 0$ go to step 3.

STEP 3: Add column s into (3.4) as the J + 1st variable. Set J = J + 1 and go to step 1.

We know the column generation algorithm is finite because n is finite and the subproblem is valid only if it can evaluate all variables $x_j = 1,2,\ldots,n$. The efficiency of the column generation algorithm is predicated on the following assumptions: (1) that optimal solutions to (3.1) can be found by generating relatively few columns, and (2) the subproblem (3.7) can be solved efficiently (Bradley, *et al.* 1977). If this is not the case, we would expect to encounter the same problems inherent in the master problem (3.1) and will be unable to solve the model.

3.2. Column Generation And The FRFLP

The column generation procedure has been successfully utilized in a variety of routing and scheduling problems. Examples of such problems are the vehicle routing problem (Cullen, Jarvis, and Ratliff 1981) and the vehicle routing problem with time windows (Desrosiers *et al.* 1984). Both of these applications are mixed integer rather than pure linear programming problems. To solve these discrete problems, the original column generation procedure has been adapted to a heuristic procedure for use with Integer Programming problems (Cullen *et al.* 1981). Because the FRFLP is stated as an MIP, we use the adapted column generation heuristic to solve the FRFLP model.

Subsection 3.2.1 discusses the column generation heuristic developed to solve the FRFLP. Subsection 3.2.2 describes the subproblem and states a path algorithm which is suitable for generating potential new columns.

3.2.1. The Column Generation Heuristic

The options available for designing a column generation technique for the FRFLP are discussed as follows. (1) This option involves treating the FRFLP as a linear program and applying the column generation procedure in its exact form to the LP until the convergence criterion is satisfied. (2) This option treats the FRFLP as an integer program and requires development of a column generation heuristic for improving the quality of the solution.

The difficulty in applying option 1 to the FRFLP is illustrated by the following example. Two channels are specified over a network of seven vertices (Figure 3.1). Channel 1 requires movement of cargo from vertex 1 to vertex 3 and channel 2 requires movement of cargo from vertex 2 to vertex 4. Let the paths of maximum capacity for the two channels be:

Channel [1,3]: 1,(1,5),5,(5,3),3

and

Channel [2,4]: 2,(2,7),7,(7,4),4

If only one facility with a capacity of two paths needs to be located then the optimal solution to the linear program will be to locate half of the facility at vertex 5 and the other half at vertex 7. Because the integrality restrictions on the number and location of facilities this represents an infeasible solution. However, if the LP starts out with these two paths as the candidate paths, the procedure will terminate without generating any new columns. The integer optimal solution may be to locate the facility at an intermediate vertex, say vertex 6, so that both channels can use it without causing too much diversion on the paths, resulting in only a marginal reduction in the capacities of these paths. Since the LP has difficulty in identifying new columns that have a potential for utilizing a facility to its capacity, option 2 is selected as the preferred approach for generating new columns.

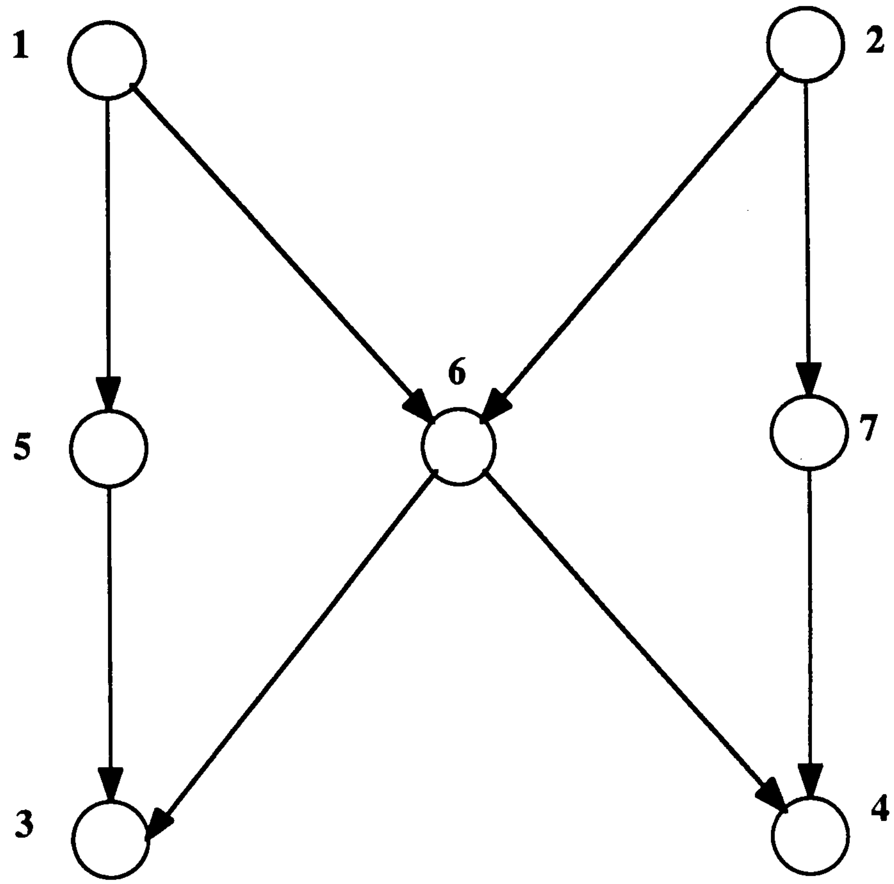


Figure 3.1

Example Seven Vertex Network

The column generation procedure cannot guarantee optimality for discrete linear models due to the problems of extracting the true shadow prices from the optimal solution. The dual values associated with an LP model are valid only if the LP model contains a full description of the convex polytope associated with the problem. In the case of IP and MIP models, a complete facet description of the integer polytope is usually unavailable unless all the relevant facet defining constraints have been added to the problem. If this is not the case, then the dual solution to a discrete problem solved using LP techniques might contain incorrect values as the complete (discrete) problem is not fully defined.

The column generation heuristic constructs new columns with respect to the optimal solution of the integer program such that the new columns are potential candidates to enter into the solution. The characteristics of the integer optimal solution that have an impact on the potential of a new column for improving the current solution are as follows:

> If one or more facilities in the current integer solution are not fully utilized, then new paths passing through these facilities can be added to the solution to cover any remaining demand W_{ij} without incurring the cost Y_k of opening a facility.

> If facilities at some vertex k are fully utilized, then adding a new path incident with vertex k will require opening a new facility with a cost Y_k. However, if the number of facilities at vertex k has already reached M_k, then adding a new path at v_k will necessarily require 'bumping' an existing path from the current solution.

These ideas are utilized in the proposed heuristic that constructs new paths to be added to the existing set of paths for the next iteration of the column generation algorithm.

The path construction algorithm seeks to find the most beneficial path for a given channel. The benefit of any path p is estimated as a function of the capability of the path, C_p, less the sum of *surrogate costs* of the facilities required by the path. The surrogate costs of facilities are assigned heuristically by taking into account the current integer optimal solution in the following manner:

Let SC_k be the surrogate cost of using a facility at vertex k. Then:

$SC_k = 0$	if there exists an underutilized facility at vertex k in the current solution.
$SC_k = Y_k$	if all facilities at vertex k are fully utilized and the number of facilities at vertex k is less than M_k.
$SC_k = \delta Y_k$	if all facilities at vertex k are fully utilized and M_k facilities are located at vertex k. $\delta \geq 1$.

Thus, the estimated benefit, b'_p, of a path p from v_i to v_j is defined by: .

(3.8)
$$b'_p = R_{ij}.C_p - \sum_{k \in K_p} SC_k$$

The heuristic constructs a path from the origin to the destination of a channel that attempts to maximize the estimated benefit b'_p. The parameter δ models the increase in cost, over and above Y_k, incurred by bumping a path from the current solution. This parameter is necessary as the true dual costs associated with opening an extra facility at some vertex k are unknown. In general, as δ increases from 1 a greater penalty is associated with creating paths which require 'bumping' existing paths from the solution. As δ increases, the path submodel is forced to generate alternative routes through the network. The effects of δ on the convergence speed of the column generation algorithm and value of the objective function are explored more fully in Section 5.3. The next subsection describes the path submodel in detail.

3.2.2. The Path Submodel

The path submodel can be stated as a multicriteria optimization problem with two objectives: (1) a maximization problem where the path capability is being maximized, and (2) a minimization problem where the sum of facility costs incurred along the path is being minimized.

Since the surrogate cost, SC_k, incurred at each intermediate vertex k on path p is non-negative, the estimated benefit, b'_p, can be shown to be monotonically nonincreasing while traversing the path from the origin to the destination. In this case, the labeling algorithm of Dijkstra (1959) can be modified to find a suitable path for a given channel. The modified form of Dijkstra's algorithm was developed from the statement of the path algorithm in Murty (Forthcoming). Dijkstra's original labeling algorithm is stated for determining the shortest path between two distinct vertices in a directed network. The proof of the validity of Dijkstra's algorithm for the shortest path problem utilizes the property of nondecreasing cumulative length/cost from the origin during the path construction process.

The problem of maximizing benefit where benefit is strictly nonincreasing along the path is equivalent to minimizing cost where the cost is nondecreasing along the path.

Let G = {V, A, c, s, t} be a directed network consisting of a set of vertices, $V = \{v_1, v_2, ..., v_n\}$, a set of arcs, A, with capacity, $c(i,j) \geq 0$, and s and t be an origin and a destination vertex respectively ($s,t \in V$). Let each vertex v_i, $i = 1,2,...,n$, have a label which will hold the following information:

P(i) = the predecessor of vertex i (*i.e.*, the vertex preceding vertex i in the subpath from the origin, s, to v_i). This is originally set to null (-).

c_i = the capacity of the constraining arc from the source, s, to the current vertex i.

t_i = the total cost of all facilities located such that the path from s to i is feasible.

b_i' = the estimated benefit associated with the path from s to v_i.

Each vertex can be in one of three mutually exclusive states: Unlabeled, Temporarily Labeled, and Permanently Labeled. The path algorithm utilizes three sets, X, Y, and Z which correspond to the sets of Unlabeled, Temporarily Labeled, and Permanently Labeled vertices respectively. At each iteration of the algorithm, exactly one vertex is moved from set Y (Temporarily Labeled vertices) to set Z (Permanently Labeled vertices). The path algorithm terminates when the set Y is empty. Upon termination, one of two cases could occur. If vertex t, the destination, has been Permanently Labeled, (that is, $t \in Z$), then a unique path, termed the *path of maximum benefit* has been found between s and t. If, on the other hand, t has not been labeled, then no path exists between s and t and the network is unconnected. The algorithm consists of six steps:

STEP 0: Set $X = V$ and $Y = Z = \varnothing$.

STEP 1: Label the origin s with $P(s) = -$, $c_s = \infty$, $t_s = 0$, $b'_s = \infty$, and put it into set Z, *i.e.*, make it permanently labeled. For each j such that $(s, j) \in A$, label v_j with $P(j) = s$, $t_j = SC_j$, $c_j = c(s,j)$ and $b'_j = R_{st}.c_j$. Add vertex j into the set Y. Go to Step 2.

STEP 2: If $Y = \varnothing$ go to Step 5, else select vertex i which satisfies $b'_i = \mathrm{Max}_{v_j \in Y}\{b'_j\}$, breaking ties arbitrarily. Move vertex i from set Y into set Z and set $\gamma = 0$ if excess capacity exists at v_i, else set $\gamma = 1$. If $\gamma = 1$ and the number of facilities located at v_i is equal to the maximum number of facilities that could be located at that vertex, set $\delta_i = \delta$ else set $\delta_i = 0$. Go to step 3.

STEP 3: For each $j \in Y$, $(i,j) \in A$ calculate the potential estimated benefit, b''_j, as $b''_j = R_{st}.(\mathrm{Min}\ \{c_i, c(i,j)\}) - t_i$. If $\delta_i > 0$ set the potential estimated benefit $b''_j = b''_j - (\gamma.Y_i\ .\delta_i)$ else set $b''_j = b'_j - (\gamma\ .Y_i)$. If $b''_j > b'_j$ update the label at j with $P(j) = i$, $b'_j = b''_j$, $t_j = ti + (\gamma.Y_i.\delta_i)$ if $\delta_i > 0$; otherwise set $t_j = t_i + (\gamma.Y_i)$, and set $c_j = \mathrm{Min}\ \{c_i, c(i,j)\}$. Go to step 4.

STEP 4: For each $j \in X$ and $(i,j) \in A$, label j with $P(j) = i$, $b'_j = R_{st}.(\mathrm{Min}\ \{c_i, c(i,j)\}) - t_i$, $t_j = t_i + (\gamma.Y_i.\delta_i)$ if $\delta_i > 0$; otherwise set $t_j = t_i + (\gamma.Y_i)$, and set $c_j = \mathrm{Min}\ \{c_i, c(i,j)\}$. Move j from X to Y. Go to step 5.

STEP 5: If $t \in Z$, terminate. If $t \notin Z$, but $Y = \varnothing$, there exists no path from s to t, terminate. If $Z \neq V$ and $Y \neq \varnothing$ go to step 2.

The path algorithm stated above is not optimal for all cases. Consider the network depicted in Figure 3.2. The numbers in parentheses along the arcs indicate the arc capacity, and the labels associated with the i^{th} vertex consist of the predecessor, the maximum capability, the number of facilities required, and the total estimated benefit respectively. It is assumed that no facilities are currently located on the network and the estimated path benefit is calculated as:

$$b'_p = (100 \times C_p) - (100 \times t_i) \tag{3.9}$$

where t_i is the number of facilities required to get to vertex i from s. The origin, s, is vertex 1 and the destination vertex, t, is 4. The algorithm progresses in the following manner:

STEP 1: The origin is labeled with $(-,\infty,0,\infty)$ - no predecessor, an infinite arc capacity, no facilities and an infinite benefit, and moved into the set Z. Vertices 2 and 3 are scanned and labeled with (1,25,1,2400) and (1,22,1,2100) respectively and moved into the set Y.

STEP 2: Vertex 2 is selected from set Y and moved into set Z.

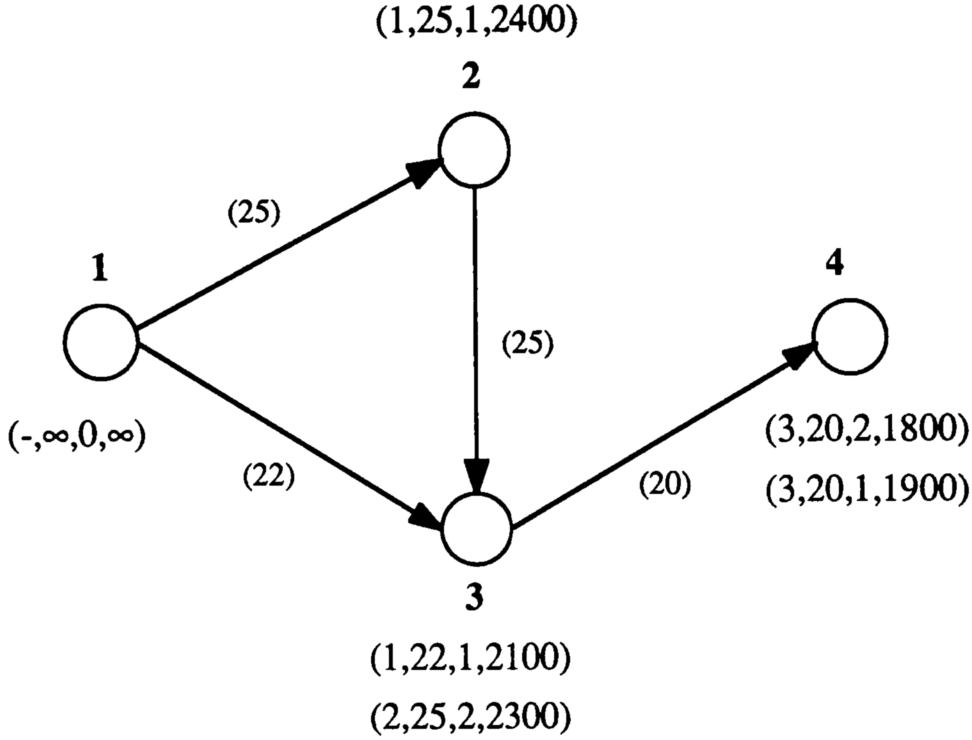


Figure 3.2

Example Network Showing The Labeling Algorithm

STEP 3: Vertex 3 is scanned from 2. A new label (2,25,2,2300) is created which has a larger objective value than the present label at vertex 2. The label at vertex 2 is updated.

STEP 4: No further vertices can be labeled.

STEP 5: The destination t has not been labeled and the set Y is not empty.

STEP 2: Vertex 3 is selected from set Y and moved into set Z.

STEP 3: Set Y is empty.

STEP 4: Vertex 4 is scanned and labeled with (3,20,2,1800). Vertex 4 is moved into set Y.

At the next iteration the destination is fixed and the algorithm terminates. The path of maximum capacity returned by the algorithm is 1,(1,2),2,(2,3),3,(3,4),4 with a capability $C_p = 20$ and requiring two facilities, one at vertex 2 and one at vertex 3. The estimated benefit of the path $b'_p = 1800$. However, had the label at vertex 3 not been updated in step 3 when vertex 2 was being fixed, vertex 4 could have been labeled from 3 with (3,20,1,1900) which has a higher valued objective function. Thus, the algorithm behaved in a suboptimal manner and the optimal path is in fact 1,(1,3),3,(3,4),4 with a benefit of 1900 and requires only that a single facility be located (at vertex 3). This suboptimality occurs because the estimated benefit of a path is dependent on two criteria.

The issue of suboptimality in multicriteria path problems has been addressed recently in the literature. Berman, Einav, and Handler (1990) identify a similar problem, termed the *constrained bottleneck problem* which is solved using parametric techniques. Carraway, Morin and Moskowitz (1990) state a Generalized Dynamic Programming approach to solve a multicriteria version of the shortest path problem. The column generating submodel within Algorithm I could be modified to include one or the other of these approaches. However, both of these approaches are computationally expensive and, as such, are only applicable to small problems since the number of operations required to achieve optimal solutions using these methods increases rapidly with the problem size. Moreover, even if the column generating algorithm were modified to find the optimal solution, the issue of inexact dual costs from the restricted master problem would still make the column generation approach heuristic. To this end, the labeling algorithm described above is used as the column generating algorithm as it has the following advantages: the labeling algorithm is fast and efficient, so, many columns can be generated with little computational effort for even large networks; and the labeling algorithm behaves in a predictable manner and, as such, can be controlled using surrogate costs.

3.2.3. Statement of Algorithm I

Algorithm I is an iterative procedure which alternates between solving a restricted master problem and the path-of-maximum-benefit submodel.

Define:

r = iteration index.

P^r = set of paths at the r^{th} iteration.

P^r_{ij} = subset of P^r consisting of paths between v_i and v_j.

P^r_k = subset of P^r containing paths which include v_k as an intermediate vertex.

L^r_k = number of facilities assigned to the k^{th} vertex at the r^{th} iteration.

The restricted master problem is stated as:

$$\text{(3.10)}\quad \text{Max } z^r = \sum_{i}^{n} \sum_{j}^{n} \sum_{p \in P^r_{ij}} R_{ij}.C_p.Z_p - \left(\sum_{v_k \in V} Y_k.L^r_k + \sum_{i}^{n} \sum_{j}^{n} R_{ij}.S_{ij} \right)$$

such that:

$$\text{(3.11)}\qquad \sum_{p \in P^r_k} Z_p - Q_k.L^r_k \leq 0 \qquad \forall\ k = 1,2,\ldots,n.$$

$$\text{(3.12)}\qquad \sum_{p \in P^r_{ij}} C_p.Z_p - S_{ij} \leq W_{ij} \qquad \forall\ [i,j].$$

$$\text{(3.13)}\qquad L^r_k \leq M_k \qquad \forall\ k = 1,2,\ldots,n.$$

(3.14) $$\sum_{v_k \in V} L_k^r \leq M$$

(3.15) $$Z_p \in \{0,1\} \qquad \forall\ p \in P^r.$$

(3.16) $$L_k^r \geq 0, \text{ integer} \qquad \forall\ k = 1,2,\ldots,n.$$

(3.17) $$S_{ij} \geq 0 \qquad \forall\ [i,j].$$

The restricted master problem above differs from the full statement of the FRFLP model (2.3)-(2.10) in the following ways: (1) the set P^r no longer consists of unique paths in G. Rather, a single path in G can be represented by a subset of the variables in the **Z** vector; and (2) each variable Z_p is binary. That is, it can take the values 0 or 1 rather than being a general integer variable as in the full statement of FRFLP (2.3). These changes enable the path subproblem to generate as many 'copies' of a path as is required without having to worry about valid upper bounds on the number of times each path might be used.

The subproblem can be stated as:

(3.18) $$\text{Max}_{p \in P} \left\{ R_{ij} \cdot C_p - \sum_{k \in K_p} SC_k^r \cdot \gamma_k \cdot \delta_k \right\}$$

where:

$$\gamma_k = \begin{cases} 0 & \text{if no new facilities are required at k} \\ 1 & \text{otherwise.} \end{cases}$$

If $\gamma_k = 0$ then δ_k is also 0, otherwise δ_k is set as below:

$$\delta_k = \begin{cases} 1 & \text{if } L_k^{r-1} < M_k \\ \delta & \text{otherwise } (\delta \geq 1) \end{cases}$$

Algorithm I can be stated as below. Figure 3.3 shows a flow diagram of the iterative procedure.

STEP 1: Set $r = 0$, $P^r = \varnothing$, $P_{ij}^r = \varnothing$ for all [i,j] and set $P_k^r = \varnothing$ and $L_k^r = 0$ for all $k = 1,2,\ldots,n$. Go to Step 2.

STEP 2: Let $r = r + 1$, and set $L_k^r = L_k^{r-1}$ for each k. Go to step 3.

STEP 3: For each channel [i,j] solve the subproblem (3.18). If a new path p was found update the sets P^r, P_{ij}^r and P_k^r and update all L_k^r. When all [i,j] have been processed go to step 4.

STEP 4: If $P^r = P^{r-1}$ go to step 7. Else go to step 5.

STEP 5: Solve the restricted master problem using (3.10) subject to (3.11)-(3.17). If $z^r - z^{r-1} < \varepsilon$, go to step 7. Else go to step 6.

STEP 6: Update L_k^r from the optimal solution to step 5. Go to step 2.

STEP 7: STOP, the last solution from step 6 is the best.

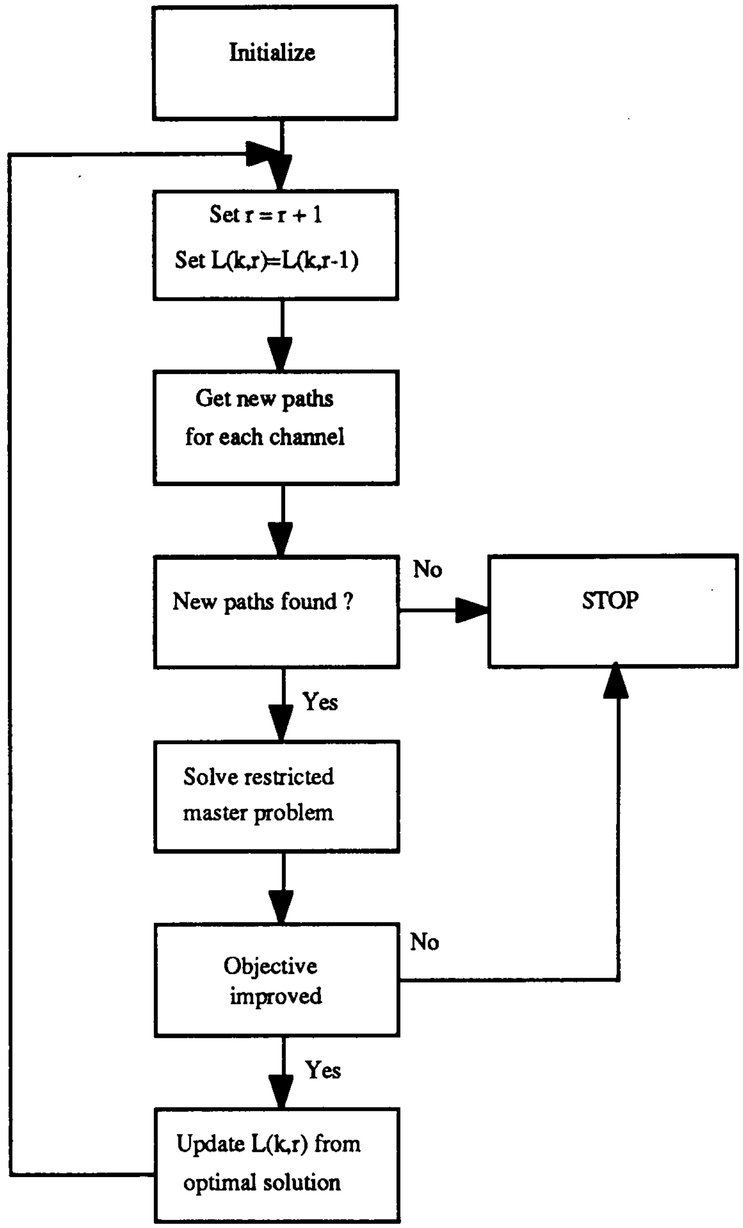


Figure 3.3

Flow Diagram For Algorithm I

The path submodel is used to generate one path for each channel in the problem at each iteration of the algorithm. The original statement of the column generation procedure generates a single column at each major iteration. However, as the majority of the solution time for the above heuristic is spent in solving the restricted master problem, we would like to make as few major iterations as possible. The restricted master problem is solved by relaxing the integrality conditions on the binary variables and using branch-bound methods.

Algorithm I can be regarded as an augmenting approach to solving the FRFLP. Each iteration of the column generation algorithm attempts to augment the network capacity by generating additional paths which could be added to the set of paths in the optimal solution of the last restricted master problem. The path submodel attempts to generate the best path for each channel given the solution to the last restricted master problem, that is, the paths which are most likely to augment the network capacity. At each iteration, the restricted master problem reorganizes the location of the facilities, given the new set of paths, so that the benefit is maximized. As no paths are ever removed from the set P^r and the restricted master problem is maximizing the objective function, the solution to the r^{th} - 1 iteration is also feasible at the r^{th} iteration. Thus, between each successive iterations $z^{r-1} \leq z^r$ and the objective function is shown to be monotonically nondecreasing. The algorithm terminates when the objective function does not improve, that is, when $z^{r-1} = z^r$.

Algorithm I starts (Step 1) with an initial feasible solution where $L_k^0 = 0$ for all $k = 1,2,\ldots,n$, and $P_B^0 = P^0 = \varnothing$. The objective function, z^0, for iteration 0 equals 0. The submodel attempts to generate new paths for each channel (Step 3). If no new paths are found, then we know that the objective function cannot improve as $P^r = P^{r-1}$, so the algorithm terminates. If one or more new paths were generated in Step 3, then the restricted master problem is resolved (Step 5) with the new paths added to the set of feasible paths P^r. The restricted master problem returns the new set of paths which are in the solution and the new minimum cost location vector $\mathbf{L^r} = \{L_1^r, L_2^r, \ldots, L_n^r\}$. If no appreciable increase in the objective function was found, the algorithm terminates as the surrogate costs, SC_k, for the next iteration (r + 1) will be the same as for the last iteration and the path submodel will, therefore, return the same set of incoming paths. The next section shows an application of Algorithm I to a small test problem.

3.3. Test Of Algorithm I

This section illustrates Algorithm I as applied to Test Problem 1, a FRFLP with seven vertices. The test network is shown in Figure 3.4 where the number next to an arc (i,j) represents c(i,j), the maximum number of units of commodity that could be moved over that arc in tons. The following data is used: The maximum number of paths serviced by a facility, Q_k, is 2, the maximum number of facilities at any vertex, M_k, is 2, and the cost of each facility Y_k, is 100 dollars for all k. The total number of facilities that could be located, M, is 6, and the revenue per unit capacity, in dollars, is: $R_{ij} = 100$ for all channels [i,j] in G. The parameter δ is initially set to 10.

The test case has two channels with demand $W_{(1,7)} = 100$ and demand $W_{(2,7)} = 100$. Figure 3.5 and Table 3.1 show the progression of the column generation algorithm. The values RW_{ij} indicate the residual demand for each channel [i,j] at the end of the r^{th} iteration.

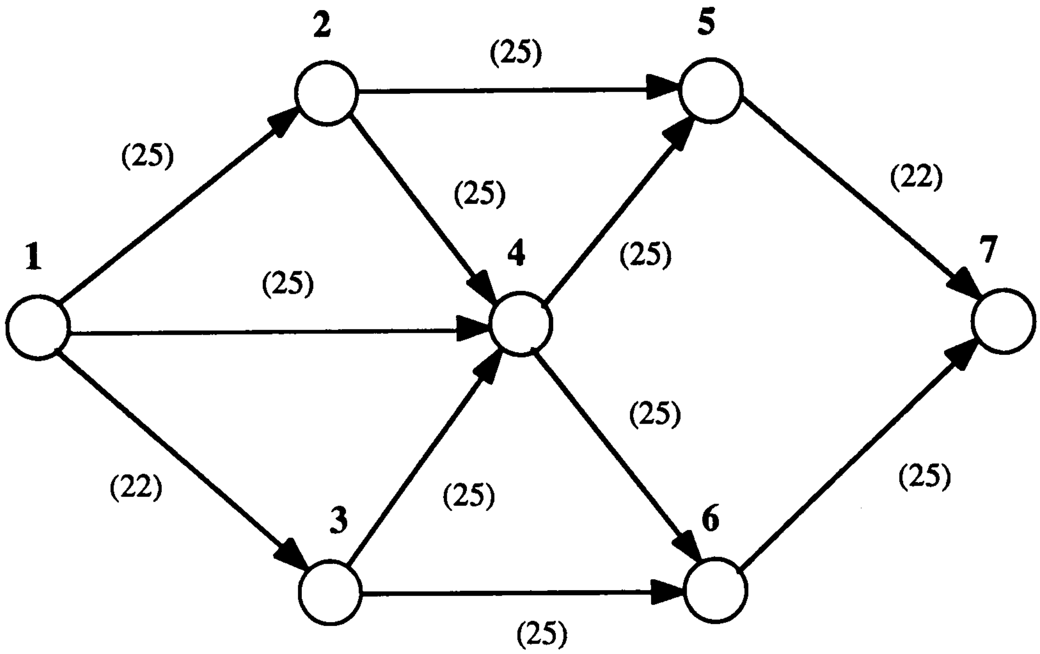
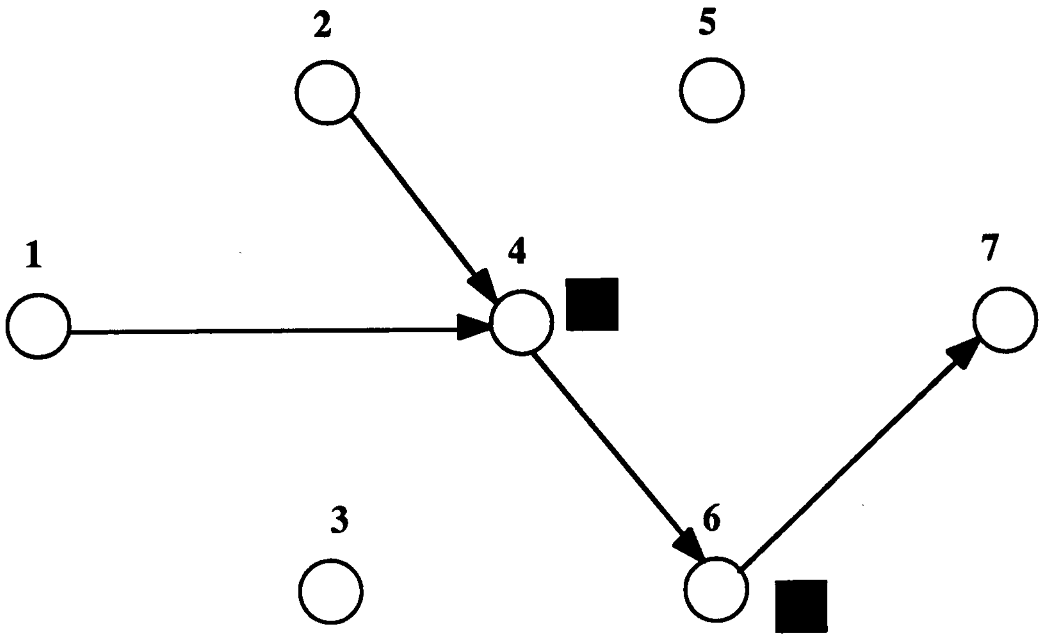


Figure 3.4

Test Problem 1: Network



Flow-Resource Facility (Partially Used)

Flow-Resource Facility (Fully Used)

1 Path

2 Paths

3 Paths

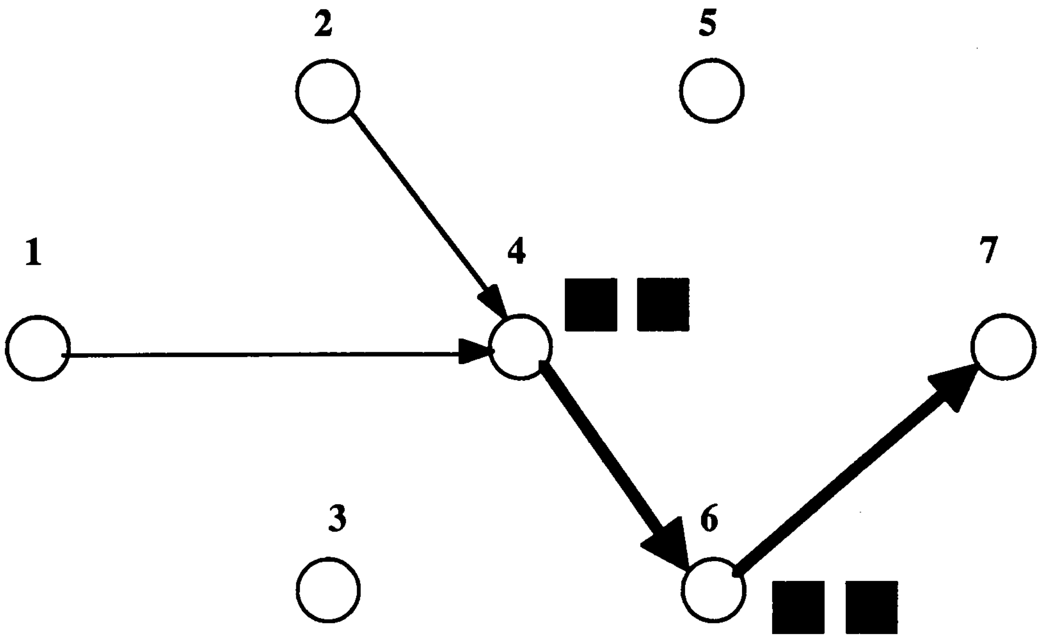
4 Paths

RW(1,7) = 75
RW(2,7) = 75

Figure 3.5

Algorithm I Iterations For Test Problem 1

(A. Iteration 1)



Flow-Resource Facility (Partially Used)

Flow-Resource Facility (Fully Used)

1 Path

2 Paths

3 Paths

4 Paths

RW(1,7) = 50
RW(2,7) = 50

Figure 3.5

Algorithm I Iterations For Test Problem 1

(B. Iteration 2)

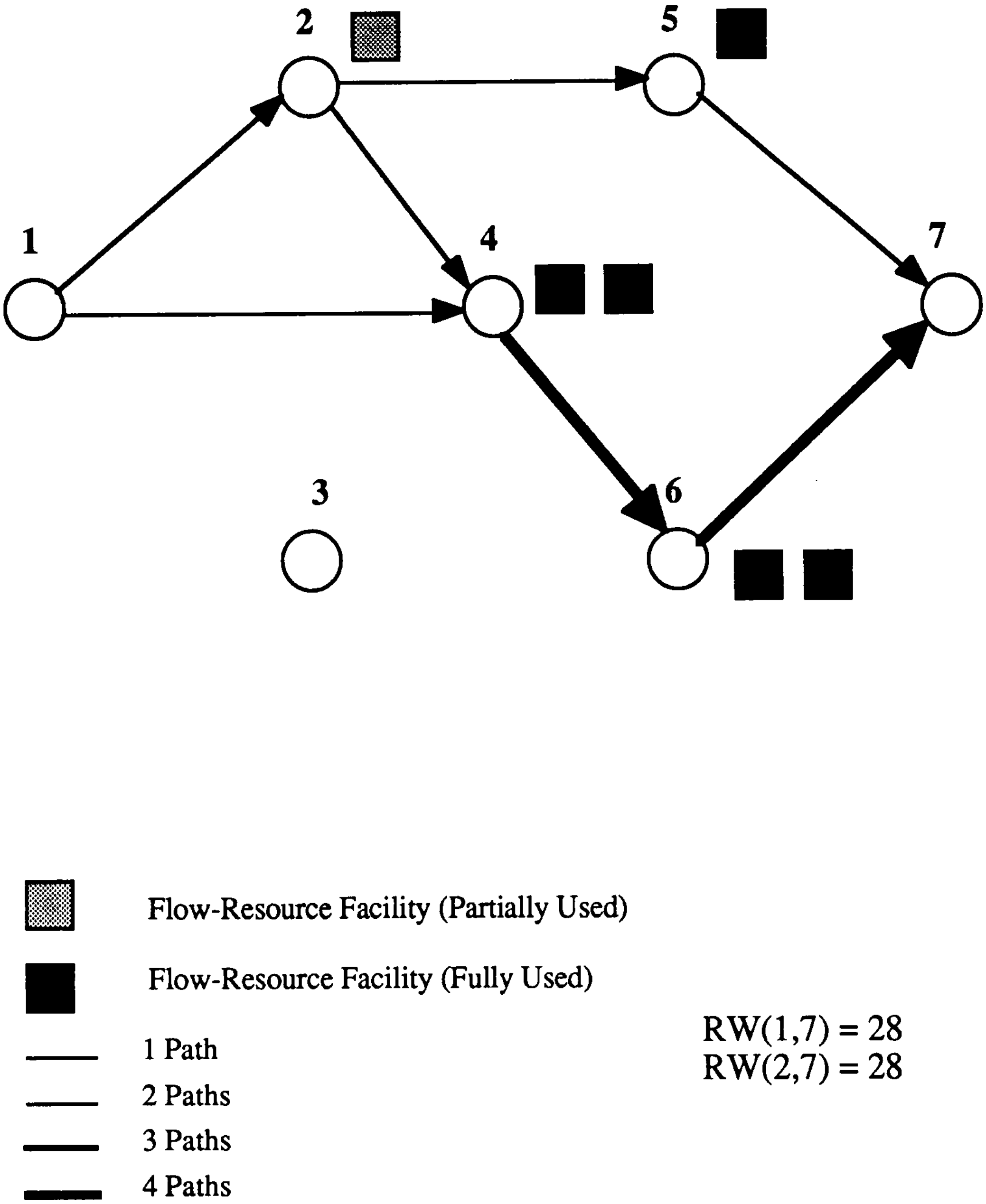
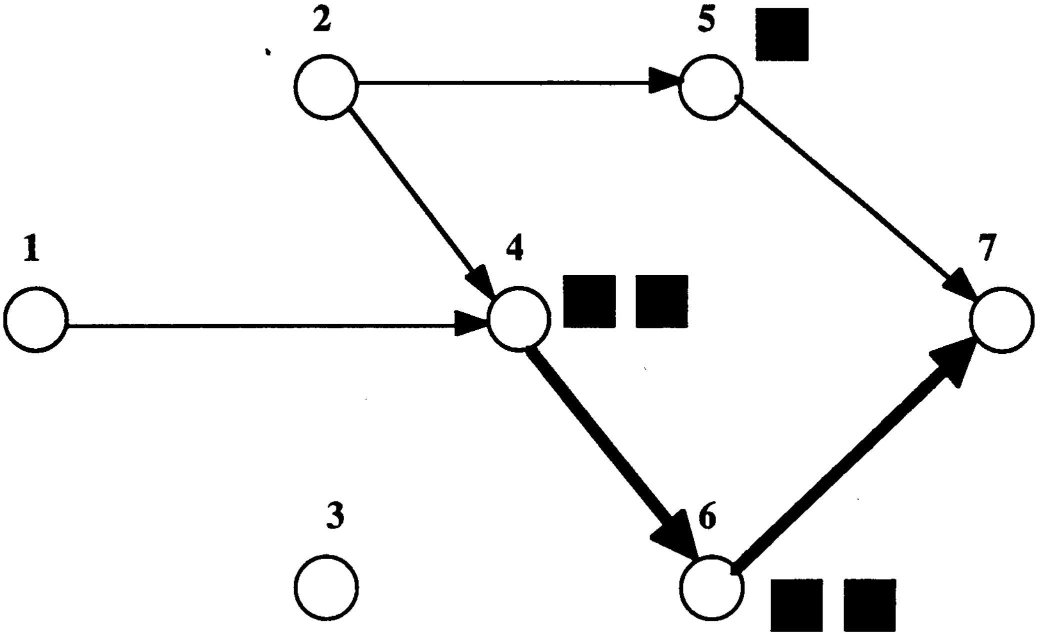


Figure 3.5

Algorithm I Iterations For Test Problem 1

(C. Iteration 3)



Flow-Resource Facility (Partially Used)

Flow-Resource Facility (Fully Used)

1 Path

2 Paths

3 Paths

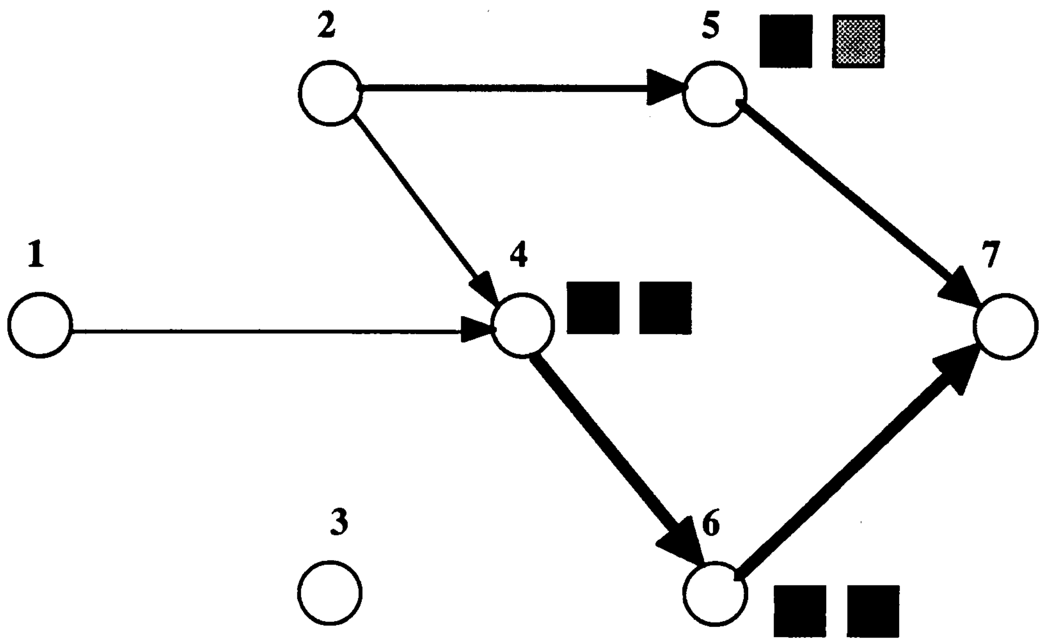
4 Paths

RW(1,7) = 50
RW(2,7) = 6

Figure 3.5

Algorithm I Iterations For Test Problem 1

(D. Iteration 4)



Flow-Resource Facility (Partially Used)

Flow-Resource Facility (Fully Used)

1 Path

2 Paths

3 Paths

4 Paths

RW(1,7) = 50
RW(2,7) = -16

Figure 3.5

Algorithm I Iterations For Test Problem 1

(E. Iteration 5)

Table 3.1

Algorithm I Progress For Test Problem 1

r	NUMBER OF PATHS (TOTAL)	NUMBER OF PATHS USED	OBJECTIVE VALUE	TOTAL NETWORK CAPACITY	NUMBER OF FACILITIES LOCATED	EXCESS CAPACITY
1	2	2	4800.0	50.0	2	0
2	4	4	9600.0	100.0	4	0
3	6	6	13800.0	144.0	6	1
4	8	6	13900.0	144.0	5	0
5	10	7	14400.0	166.0	6	1
6	12	7	14400.0	166.0	6	1

r = 1 Initially the path submodel generates two paths: 1,(1,4),4,(4,6),6,(6,7),7 for the first channel with a capacity of 25 tons, and path 2,(2,4),4,(4,6),6,(6,7),7 for the second channel also with a capacity of 25 tons. These paths are passed to the IP model, and the optimal solution is found with both paths in the solution and a single facility at vertices 4 and 6, both fully used. Figure 3.5 Iteration 1.

r = 2 As both vertices 4 and 6 could accommodate another facility, the paths generated by the column generation submodel were the same as for iteration 1. All four paths are used in the optimal solution and two facilities, both used to capacity, are located at vertices 4 and 6. Figure 3.5 Iteration 2.

r = 3 The path submodel generates two new paths: 1,(1,2),2,(2,5),5,(5,7),7 for the first channel and path 2,(2,5),5,(5,7),7 for the second channel. Both paths have a capacity of 22 tons. All six paths are used in the optimal solution. Two facilities are located at 4 and 6 respectively, both used to capacity. A single facility is located at vertex 2 and another at vertex 5. The facility at 5 is capacitated while that at vertex 2 has one unit of excess capacity. Figure 3.5 Iteration 3.

r = 4 As vertices 2 and 5 are potential sites for facilities, the path submodel generates the same paths as the previous iteration. The IP consolidates flow through vertex 5 on the second channel. This step does not augment the flow in the network, but the solution requires one less facility to be located. Vertex 5 has a single facility fully used. Figure 3.5 Iteration 4.

$\underline{r = 5}$ The path submodel again generates paths through vertices 2 and 5 recognizing that both vertices are potential sites for facilities. The IP locates the remaining facility at vertex 5. Figure 3.5 Iteration 5.

$\underline{r = 6}$ The path submodel generates paths through vertices 2 and 5 but the IP objective cannot be improved. The procedure terminates.

Test Problem 1 was solved on a Sun Sparc 4/330 Workstation using a total of 5.8 cpu seconds of which 5.6 cpu seconds were spent solving the restricted master problem. The restricted master problem was solved using Sun/XA. (Footnote 4). The example shows the amount of cargo capacity in the network monotonically increasing up to the fourth iteration. Here, the algorithm consolidates facilities at vertex 5 without expanding the network capacity. Iteration 5 adds a single flow-augmenting path to the network. Figure 3.6 shows the results of rerunning iteration 6 with M increased to 7 from 6. The objective function improves to 18,100 from 14,400 and the total residual demand for capacity drops from 50 to 12 tons.

3.4. Summary Of Chapter 3

This chapter introduced Algorithm I as a means of solving the FRFLP model heuristically. Algorithm I is based on a column generation procedure which has been modified to work with discrete problems. These modifications included using surrogate costs to approximate the dual prices from the optimal MIP and generating many new columns at each iteration of the column generation algorithm. The submodel was stated in detail, as was an algorithm suitable for solving the submodel derived from a shortest path

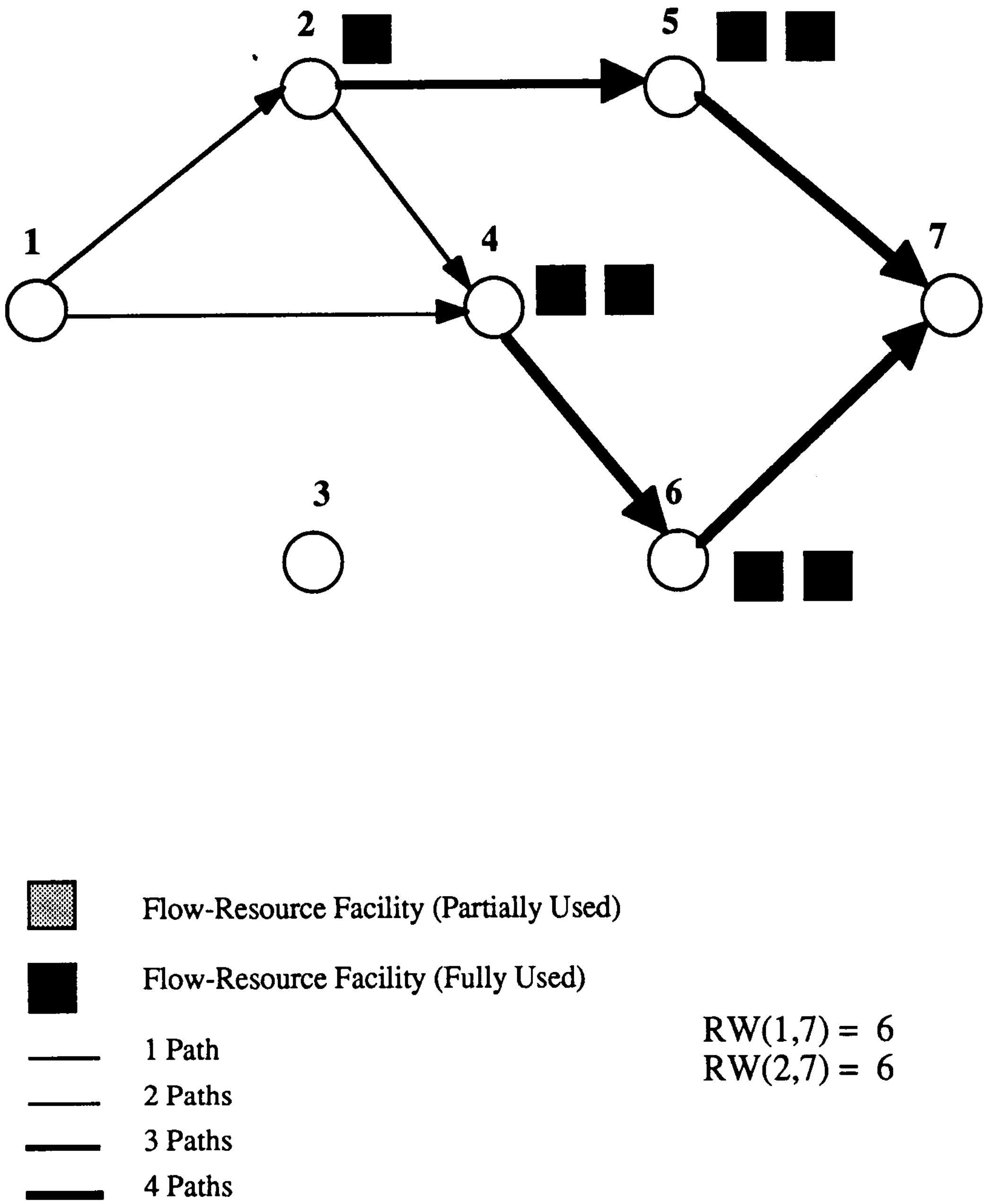


Figure 3.6

Algorithm I Iteration 6 For Test Problem 1 algorithm. The issue of suboptimality in the path problem is discussed further in section 6.1.2. The column generation heuristic was then outlined and applied to Test Problem 1 which consists of seven vertices and two channels. The next chapter introduces a location-allocation heuristic which replaces the MIP phase of Algorithm I in order to reduce the total execution time and be able to solve larger, more realistic, problems.

CHAPTER 4

Algorithm II: Location-Allocation

The algorithm described in this chapter uses a Location-Allocation model to locate flow-resource facilities at the vertices of a network. Like Algorithm I, a column generation procedure is used to iteratively generate new paths. However, Algorithm II solves the restricted master problem using a two-stage heuristic, thus speeding up the total execution time taken to solve the FRFLP models. Subsection 4.1 describes the two-stage heuristic. Subsection 4.2 describes the complete Algorithm II while subsection 4.3 applies the new algorithm to the seven vertex test problem of Chapter 3.

4.1. Location-Allocation Submodel

The location-allocation submodel of Algorithm II is a substitution (or swapping) algorithm which has been used in a number of successful heuristics. Among the more well-known substitution algorithms are the Tietz and Bart Vertex Substitution heuristic (Tietz and Bart 1968) and the k-Opt heuristics for the Traveling Salesman Problem (Lin and Kernighan 1973). These algorithms are improvement heuristics which, given an initial feasible solution, attempt to improve the objective function by replacing in-solution arcs or nodes with out-of-solution arcs or nodes. In the case of the Vertex Substitution Algorithm, vertices which are in the initial feasible solution (*i.e.*, have facilities assigned to them) are replaced in turn by each of the vertices without a facility assigned to them. The marginal change in the objective function is calculated for each possible swap, and the swap which gives the greatest improvement in the objective function, if any, is made. The arc-substitution algorithms for TSPs by Lin and Kernighan use the same technique. In-solution arcs are replaced by out-of-solution arcs and the swap leading to the greatest marginal improvement in the objective function, if any, is kept. In the TSP algorithms, a solution is said to be k-optimal if, by replacing any k in-solution arcs with any k out-of-solution arcs, the current feasible solution cannot be improved.

Both the Vertex Substitution and the k-Opt algorithms require an initial feasible solution before the improvement algorithms can be used. Generally, initial solutions are generated using *greedy* or *random* algorithms. The former myopically attempt to find a feasible solution by adding elements into the solution set in an sequential manner. The latter approach uses a randomly generated solution. Algorithm II described in this chapter requires such an initial feasible solution. Subsection 4.1.1 describes a greedy algorithm used to generate feasible solutions and Subsection 4.1.2 describes the substitution algorithm used to improve the feasible solutions generated.

4.1.1. Generating An Initial Feasible Solution

Let P^r be the set of all paths generated by the r^{th} iteration. The set P^r can be partitioned into a set of in-solution paths, P^r_B, and a set of out-of-solution paths, P^r_{NB}, where $P^r_B \cup P^r_{NB} = P^r$ and $P^r_B \cap P^r_{NB} = \varnothing$. Given a restricted master problem of the form (3.10)-(3.17), a feasible location vector must have the following attributes:

No more than M facilities can be located on G.

No more than M_k facilities can be located at any vertex k.

The total demand exerted on any vertex k by the subset of the set of paths P^r_k which have vertex k as an intermediate vertex on the path, can be no greater than the capacity allocated to vertex k, $(L_k \times Q_k)$.

Similarly, the set of paths in a feasible solution to (3.10) must have the following attributes.

No more than $| P^r |$ paths can be in the set P^r_B of in-solution paths.

Each path, $p \in P^r_B$, must have a portion of a facility's capacity allocated to it at every intermediate vertex on the path.

The path capability attributed to a path $p \in P^r_B$ must be no greater than the actual capability of path p.

Having defined the requirements for a feasible solution, we can design a greedy algorithm to generate a location vector and a set of in-solution paths.

Define:

r = Iteration counter.

M^r = Number of facilities available to be located in the r^{th} iteration.

L_k^r = Number of facilities located at vertex k in the r^{th} iteration.

V_k^r = Available residual capacity at vertex k in the r^{th} iteration.

W_{ij}^r = Residual demand for channel [i,j] in the r^{th} iteration.

P_B^r = Set of in-solution paths in the r^{th} iteration.

P_{NB}^r = Set of out-of-solution paths in the r^{th} iteration.

The greedy algorithm is an iterative procedure composed of four steps.

STEP 0: Initialization. Set $r = 1$, $P_{NB}^r = P^r$, $P_B^r = \varnothing$, $M^r = M$, and set $L_k^r = 0$ and $V_k^r = 0$ for all $k = 1,2, ...,n$.

STEP 1: Find path a $p \in P_{NB}^r$ such that $b_p > 0$. If such a path cannot be found stop. Else go to step 2.

STEP 2: For each intermediate vertex, k, on path p, if $V_k^r > 0$ then set $V_k^{r+1} = V_k^r - 1$, else set $L_k^{r+1} = L_k^r + 1$, $M^{r+1} = M^r - 1$, and $V_k^{r+1} = Q_k - 1$. Update the residual channel demand for path p such that $W_{ij}^{r+1} = W_{ij}^r - C_p$ where v_i and v_j are the origin and destination, respectively of path p. Go to step 3.

STEP 3: Update the path sets, set $P_B^{r+1} = P_B^r \cup \{p\}$ and $P_{NB}^{r+1} = P_{NB}^r \setminus \{p\}$. Set $r = r + 1$. Go to step 1.

Step 0 initializes the solution by ensuring that no facilities are allocated and that no paths are in the solution. The set of out-of-solution paths is initialized to the complete set of paths and the iteration counter is set to 1. Step 1 selects a path from the set of out-of-solution paths such that the benefit to be gained from the path is strictly positive. (The benefit of any path is calculated subject to the allocation of facilities in the r^{th} iteration and the set of in-solution paths already selected.) If no path with a positive benefit can be found, then the algorithm terminates in the r^{th} iteration with P_B^r as the set of in-solution paths and the location vector $L^* = L^r = \{l_1, l_2, ..., l_p\}$, where l_i is the location of the i^{th} facility, as the best location vector. Otherwise, step 2 updates the location vector L^r such that the current solution with p added to the set of in-solution paths is feasible. Step 2 is also responsible for opening any more facilities and altering the residual capacity of each affected facility. The residual channel demand is also updated in step 2. Step 3 moves path p from the set of out-of-solution paths to the set of in-solution paths and increments the iteration counter.

At each iteration, a solution feasible for (3.10)-(3.17) is maintained and each subsequent iteration calculates the benefit of each path with respect to the last feasible solution. The benefit attributed to each path at the r^{th+1} iteration is calculated as follows:

$$\text{Benefit} = \text{Flow Benefit} - \text{Facility Costs}$$

where the flow benefit, b_p^f, of path p from v_i to v_j is calculated as:

$$b_p^f = R_{ij}.C_p - R_{ij}.S_{ij} \tag{4.1}$$

where S_{ij} is the amount of capacity overassigned to channel [i,j]. The facility costs, y_p, of path p are calculated as:

$$y_p = \sum_{k \in K_p} Y_k.\delta_k \tag{4.2}$$

where

$$\delta_k = \begin{cases} 1 & \text{if } V_k^r = 0 \text{ and } L_k^r < M_k \text{ and } M^r < M \\ \infty & \text{if } V_k^r = 0 \text{ and either } L_k^r = M_k \text{ or } M^r = M \\ 0 & \text{otherwise.} \end{cases}$$

Upon termination of the greedy algorithm, the set P of paths is partitioned into a set of in-solution paths, P_B, and a set of out-of-solution paths, P_{NB}, and a minimum cost location vector L^* which is feasible with respect to P_B is created. The sets P_B and P_{NB} and the location vector L^* serve as inputs to the improvement algorithm described in the next subsection.

4.1.2. Improving The Initial Solution

This subsection details the substitution algorithm used to improve the quality of the solution generated by the greedy heuristic described in subsection 4.1.1. Given a location vector L^* and the sets P_B and P_{NB} described above, the substitution algorithm iteratively calculates the marginal benefit attributed to replacing, in turn, each in-solution path with an out-of-solution path. For every feasible substitution, the change in the objective function value is calculated. The substitution which maximizes the potential increase in the objective function is made and the location vector updated accordingly.

Define:

$p_i \oplus p_j$ = The marginal increase in the objective function value by substituting path p_i for path p_j in the current solution.

s = the number of substitutions made.

The substitution algorithm can be stated as:

STEP 1: Set s = 0.

STEP 2: Select the next path, p_n, from P_{NB}. If all paths in P_{NB} have been processed and $s = 0$ stop. If all paths in P_{NB} have been processed and $s > 0$ go to step 1. Otherwise go to step 3.

STEP 3: Find path $p_b \in P_B$ such that $p_b = \text{Arg Max}_{p \in P_B} \{p_n \oplus p\}$. If $p_n \oplus p_b \leq 0$, go to step 2, otherwise go to step 4.

STEP 4: Set $P_B = (P_B \setminus \{p_b\}) \cup \{p_n\}$ and set $P_{NB} = (P_{NB} \cup \{p_b\}) \setminus \{p_n\}$. Update the vectors L^* and V_k. Set $s = s + 1$ and go to step 2.

The algorithm starts by selecting the first path in the list of out-of-solution paths and calculates the change in the objective function that would be made by substituting the out-of-solution path for every in-solution path in the solution. If a suitable swap is found, the solution is updated. When one or more possible swaps are found, the swap which maximizes the improvement in the objective function is made. Ties are broken arbitrarily.

If no swap was made for the out-of-solution path, the next out-of-solution path is selected and tested against the list of in-solution paths. When a single pass through the set of out-of-solution paths results in a substitution, the algorithm restarts by testing the first out-of-solution path against the first in-solution path. The algorithm terminates when a single pass through the list of out-of-solution paths does not produce any substitutions.

4.2. Statement Of Algorithm II

This section describes the complete form of Algorithm II. Figure 4.1 is a flow diagram of the algorithm which is stated below:

<u>STEP 0</u>: Set $r = 0$, $P^r = \varnothing$, $P^r_{ij} = \varnothing$ for all [i,j]. Set $P^r_k = \varnothing$ and $L^r_k = 0$ for all $k = 1,2,...,n$. Go to step 1.

<u>STEP 1</u>: Set $r = r + 1$, set $L^r_k = L^{r-1}_k$ for each k. Go to step 2.

<u>STEP 2</u>: Solve the submodel (3.18) for each channel [i,j]. If a new path p is found, update the sets P^r, P^r_{ij}, and P^r_k. When all channels have been processed go to step 3.

<u>STEP 3</u>: Generate a location vector L^r and a set of in-solution paths P^r_B using the greedy algorithm described in subsection 4.1.1. Go to step 4.

<u>STEP 4</u>: Use the improvement algorithm described in subsection 4.1.2 to improve the value of the objective function. Let z^r be the highest valued objective function found in step 4. If $z^r - z^{r-1} < \varepsilon$ stop, else go to step 1.

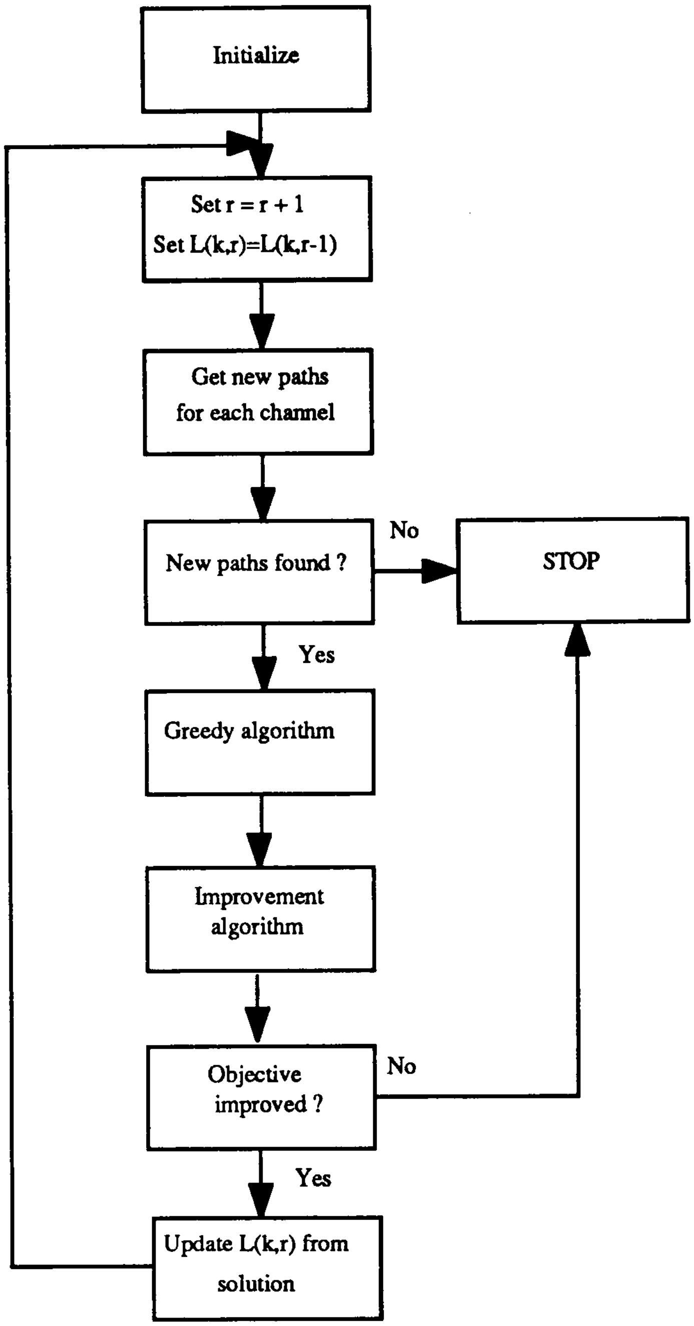


Figure 4.1

Flow Diagram For Algorithm II

Algorithm II works on the same principal as Algorithm I. At each iteration the algorithm tries to improve the objective function by generating those paths which are most likely to augment the network capacity given the last feasible location vector (Step 2). These paths are added to the set of available paths, P^r, and the restricted master problem is resolved. The greedy algorithm is used to generate a new location vector and set of in-solution paths (Step 3). The improvement algorithm (Step 4) then attempts to improve the quality of the solution by examining all possible substitutions of out-of-solution for in-solution paths. The objective function returned from step 4 is not, however, guaranteed to be monotonically nondecreasing since the greedy and substitution algorithms could easily get stuck at a local optima.The risk of this happening can be reduced by storing the set of in-solution paths, P_B^{r-1}, and the location vector, L^{r-1}, at each iteration and using them as a feasible solution to the r^{th} iteration. In such a case, the greedy algorithm should consider only those paths which have been returned by the path submodel. As substitutions are made only if they increase the value of the objective function, the value of the objective function for Algorithm II can then be shown to be monotonically nondecreasing. As with Algorithm I, the algorithm terminates when either no appreciable change in the value of the objective function between successive iterations occurs, or no paths are returned by the path submodel.

4.3. Test Of Algorithm II

This section illustrates Algorithm II as applied to Test Problem 1 described in Section 3.3. The test network is shown in Figure 3.4. As in Section 3.3 the following data is used: The maximum number of paths serviced by a facility, Q_k, is 2, the maximum number of facilities at any vertex, M_k, is 2, and the cost of each facility Y_k, is 100 dollars for all k. The total number of facilities that could be located, M, is 6, and the revenue in dollars per unit capacity allocated is given as: $R_{ij} = 100$ for all [i,j]. The parameter δ is initially set to 10. The test case has two channels with demand $W_{(1,7)} = 100$ and demand $W_{(2,7)} = 100$.

The test problem was solved on a Sun Sparc 4/330 Workstation in 0.2 cpu seconds. Algorithm II replicated the iterations of Algorithm I exactly with the exception of Iteration 4 (Figure 3.5 and Table 4.1). During iteration 4 Algorithm II initially allocated the same solution as in iteration 3 with a total volume of flow of 144 units and an objective function value of 13800. During the Improvement phase of Algorithm II, however, a better solution was identified and path 1,(1,2),2,(2,5),5,(5,7),7 was swapped for path 2,(2,5),5,(5,7),7 and the partially used facility allocated to vertex 2 was removed from the solution. The improvement phase consequently improved the objective function value from 13800 to 13900 while the total capacity of the network remained at 144 units.

4.4. Summary Of Chapter 4

This chapter introduced Algorithm II as a means of solving the FRFLP model for larger problems. Algorithm II solves the restricted master problem of the column generation algorithm (Algorithm I) with an iterative two stage location-allocation heuristic. This two stage process allows larger problems to be solved in shorter time as the highly intensive branch-bound procedures for solving the MIP model are replaced by a faster specialized heuristic. For the seven vertex Test Problem 1, Algorithm II generates the same solution as

Table 4.1

Algorithm II Progress Test Problem 1

r	NUMBER OF PATHS (TOTAL)	NUMBER OF PATHS USED	OBJ. VALUE BEFORE SUB	OBJ VALUE AFTER SUB	NUMBER OF SUBS	TOTAL NETWORK CAPACITY	NUMBER OF FACILITIES LOCATED	EXCESS CAPACITY
1	2	2	4800.0	4800.0	0	50.0	2	0
2	4	4	9600.0	9600.0	0	100.0	4	0
3	6	6	13800.0	13800.0	0	144.0	6	1
4	8	6	13800.0	13900.0	1	144.0	5	0
5	10	7	14400.0	14400.0	0	166.0	6	1
6	12	7	14400.0	14400.0	0	166.0	6	1

did Algorithm I but in less total cpu time. Chapter 5 introduces Test Problem 2, a problem derived from a real airlift scheduling problem.

CHAPTER 5

Test Problem 2

This chapter applies Algorithm II to a test problem derived from a real-world airlift scheduling problem. The airlift problem described involves flowing cargo from airbases in the United States to airbases in Northern Africa. The flow-resources being allocated are aircraft refuelers at airbases primarily in the Northern Atlantic and Northern Europe. Section 5.1 describes the problem in more detail and section 5.2 describes the solution to the problem. Section 5.3 discusses calibrating the parameter δ and its effects on the solutions generated by the heuristic.

5.1. Problem Description

Test Problem 2 involves allocating aircraft refuelers to airbases in the Northern Atlantic and North Europe. The objective of the problem is to maximize the number of tons of cargo that could be flown from airbases in the United States (origins) to airbases in North Africa (destinations). The network is composed of sixteen airbases, ten channels, and 69 directed arcs (Figure 5.1, Table 5.1 and the Appendix). The set of airbases are composed of four origins in the U.S., three destinations in Northern Africa, and nine enroute bases. All but three of the airbases (1,15, and 16) are feasible refueler sites.

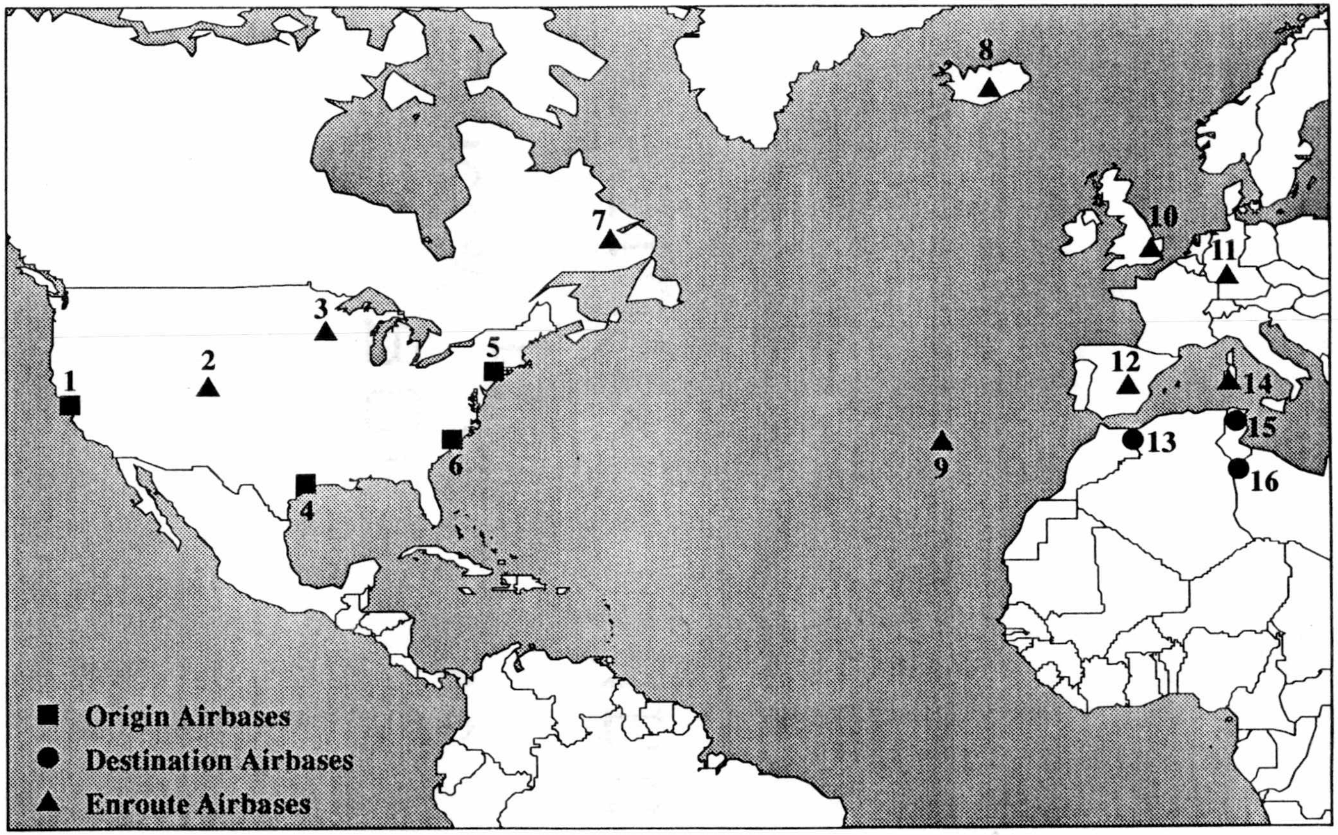


Figure 5.1

Test Problem 2: Network And Airbases

Table 5.1

Test Problem 2: Airbase Characteristics

VERTEX	AIRBASE	MAXIMUM FACILITIES	FACILITY CAPACITY	FACILITY COST (Dollars)	NUMBER OF FACILITIES LOCATED	EXCESS CAPACITY
1	TRAVIS AFB	0	0	0.0	0	0
2	STAPLETON AFB	2	6	100.0	0	0
3	MINN. ST. PAUL AFB	2	6	100.0	1	0
4	KELLY AFB	2	6	100.0	0	0
5	J.F.K. AFB	2	6	100.0	2	0
6	POPE AFB	2	6	100.0	0	0
7	AIR REFUELING	4	6	100.0	4	0
8	KEFLAVIK AFB	4	6	100.0	4	0
9	LAJES AFB	4	6	100.0	4	0
10	LAKENHEATH AFB	2	6	100.0	1	4
11	RAMSTEIN AFB	2	6	100.0	0	0
12	TOREJON AFB	2	6	100.0	0	0
13	ROTA AFB	2	6	100.0	0	0
14	DECIMOMANNU AFB	2	6	100.0	0	0
15	EL MAOU AFB	0	0	0.0	0	0
16	EL BORMA AFB	0	0	0.0	0	0

Columns 1 through 5 of Table 5.1 describe the airbase characteristics. Column 2 is the name of the US airforce base (AFB), column 3 corresponds to the value M_k, column 4 corresponds to the value Q_k and column 5 corresponds to the value Y_k. Columns 1 through 3 of Table 5.2 describe the channel characteristics. Columns 1 and 2 define the channel origins and destinations respectively and column 3 corresponds to the value W_{ij}. The test problem was initially run with the following parameters:

R_{ij} = 100 dollars for all channels [i,j].

δ = 10.

M = 16.

5.2. Solution To Test Problem 2

Table 5.3 shows Algorithm II in progress as it solved the test problem. The problem was solved on a Sun Sparc 4/330 Workstation in 0.2 cpu seconds. Algorithm II was written in C and compiled using the Sun C compiler version 2.0. Column 1 of Table 5.3 is the iteration number, r. In total, eight iterations of the column generation algorithm were required for Algorithm II to converge. Columns 2 and 3 show the total number of paths generated, $| P^r |$, and the number of paths in the solution, $| P_B^r |$, at the r^{th} iteration, respectively. Columns 4 and 5 show the value of the objective function before and after the substitution phase of Algorithm II. Column 6 shows the total number of substitutions made at each iteration. The best objective function value obtained was 98000.0 dollars. Column 7 shows the total network capacity allocated at the r^{th} iteration. This value is the summation

Table 5.2

Test Problem 2: Channel Characteristics

CHANNEL ORIGIN	CHANNEL DESTINATION	CHANNEL DEMAND (Tons)	RESIDUAL DEMAND (Tons)
1	13	100.0	-2.0
1	15	100.0	1.0
1	16	100.0	1.0
4	13	100.0	0.0
4	15	100.0	1.0
4	16	100.0	1.0
5	13	100.0	0.0
5	15	100.0	-2.0
5	16	100.0	-1.0
6	16	100.0	0.0

Table 5.3

Algorithm II Progress For Test Problem 2

r	NUMBER OF PATHS (TOTAL)	NUMBER OF PATHS USED	OBJ VALUE BEFORE SUB	OBJ VALUE AFTER SUB	NUMBER OF SUBS	TOTAL NETWORK CAPACITY	NUMBER OF FACILITIES LOCATED	EXCESS CAPACITY
1	10	10	21300.0	21300.0	0	217.0	4	4
2	20	20	43000.0	43000.0	0	437.0	7	5
3	30	24	51700.0	52000.0	3	528.0	8	0
4	40	30	63300.0	63400.0	1	634.0	10	0
5	50	40	80100.0	81100.0	3	828.0	13	2
6	60	49	94900.0	94900.0	0	872.0	15	0
7	70	50	98000.0	98000.0	0	1001.0	16	4
8	80	50	98000.0	98000.0	0	1001.0	16	4

of the path capacities in the solution set P_B^r. The maximum network capacity allocated was 1001 tons of which 5 tons were overallocated (See Table 5.2). The last two columns, 8 and 9, show the total number of facilities located on the network at the r^{th} iteration and the total excess path capacity of the location vector L^r at the r^{th} iteration.

Figure 5.2 illustrates the spatial distribution of the sixteen refuelers across the network. Four refuelers are located at airbase 7, 8, and 9. None of these three airbases has excess capacity allocated. Two refuelers are located at base 5 and one at base 10, again with no excess capacity. A single refueler is located at base 3 with an excess capacity of four paths. In terms of the channel demands, three of the ten channels ([4,13], [5,13], and [6,16]) were allocated capacity exactly equal to the demand. Three channels ([1,13], [5,15], and [5,16]) had excess an excess capacity of 2, 2, and 1 tons allocated to them respectively, a total of 5 tons over capacity. The remaining channels ([1,15], [1,16], [4,15], and [4,16]) had a residual demand of 1 ton each for a total of four residual tons. Figure 5.3 shows the pattern of paths and the distribution of facilities allocated for each channel in the final solution.

The results described above show that Algorithm II produced an intuitively good solution for the test problem. To test the quality of the solution generated by Algorithm II, an upper bound on the optimal solution to the full statement of the FRFLP ((2.3)-(2-10)) was generated for the test problem. As the FRFLP is stated as a maximization problem, the linear relaxation of the MIP acts as a valid upper bound on the optimal value to the master problem. The full statement of the MIP was generated by using a k-paths routine (Shier

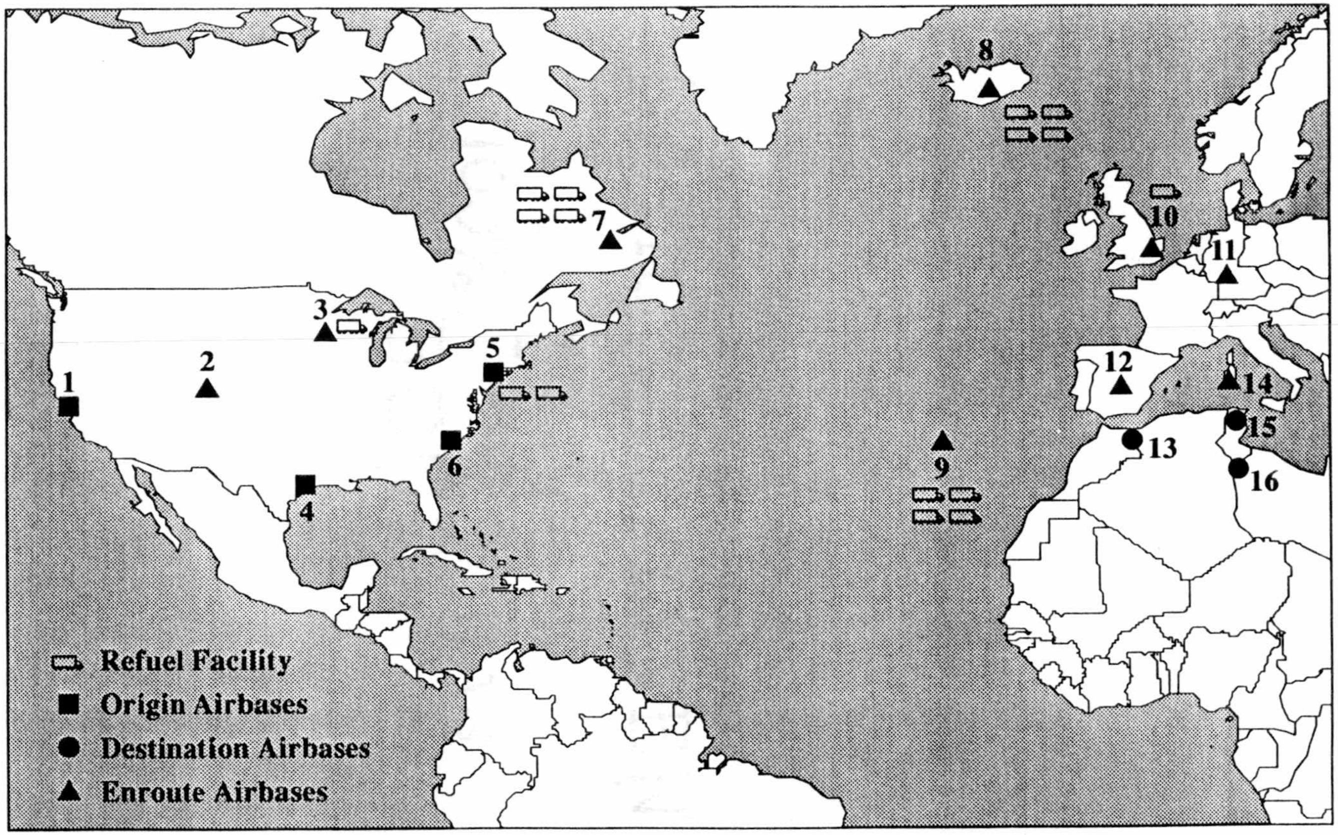


Figure 5.2

Test Problem 2: Distribution Of Aircraft Refuelers

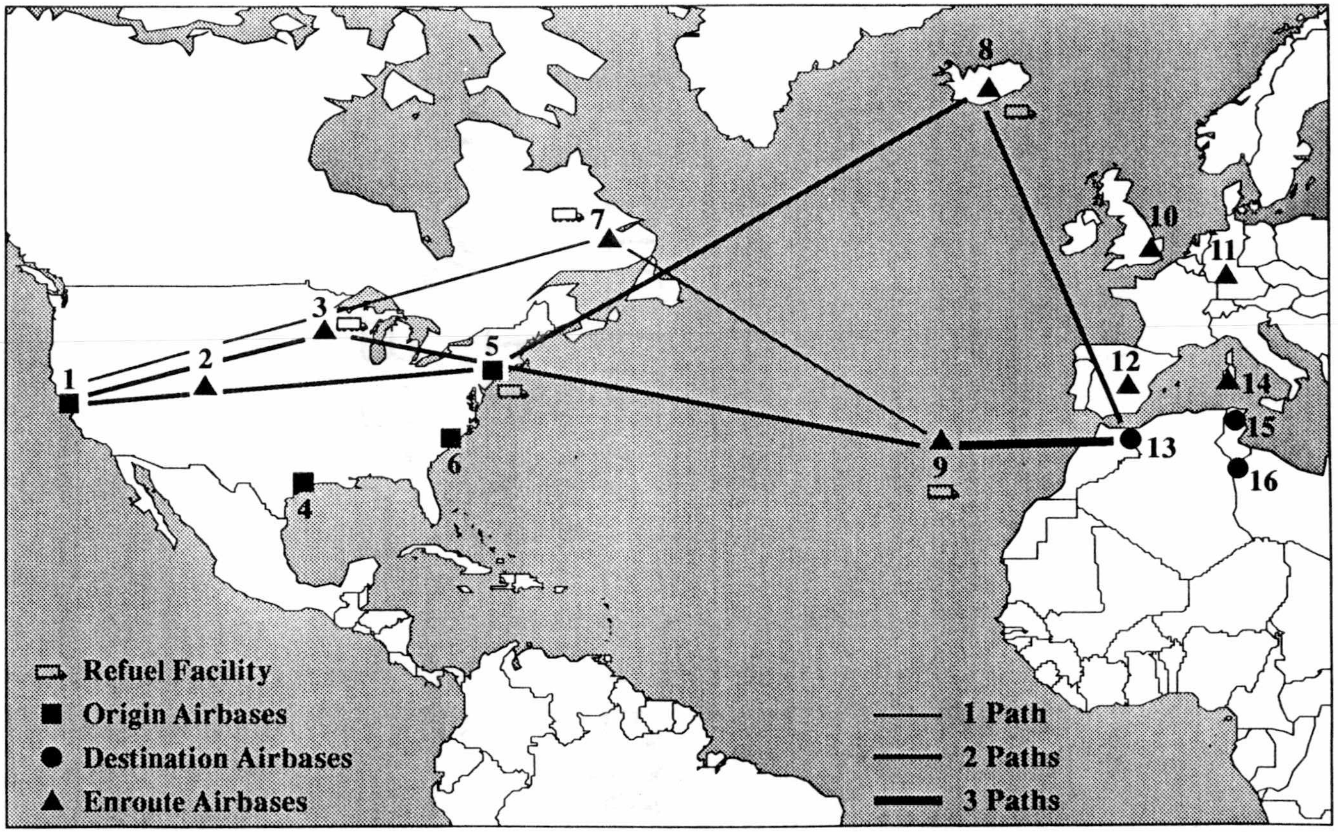


Figure 5.3

Test Problem 2: Solution By Channels

(A. Channel [1,13])

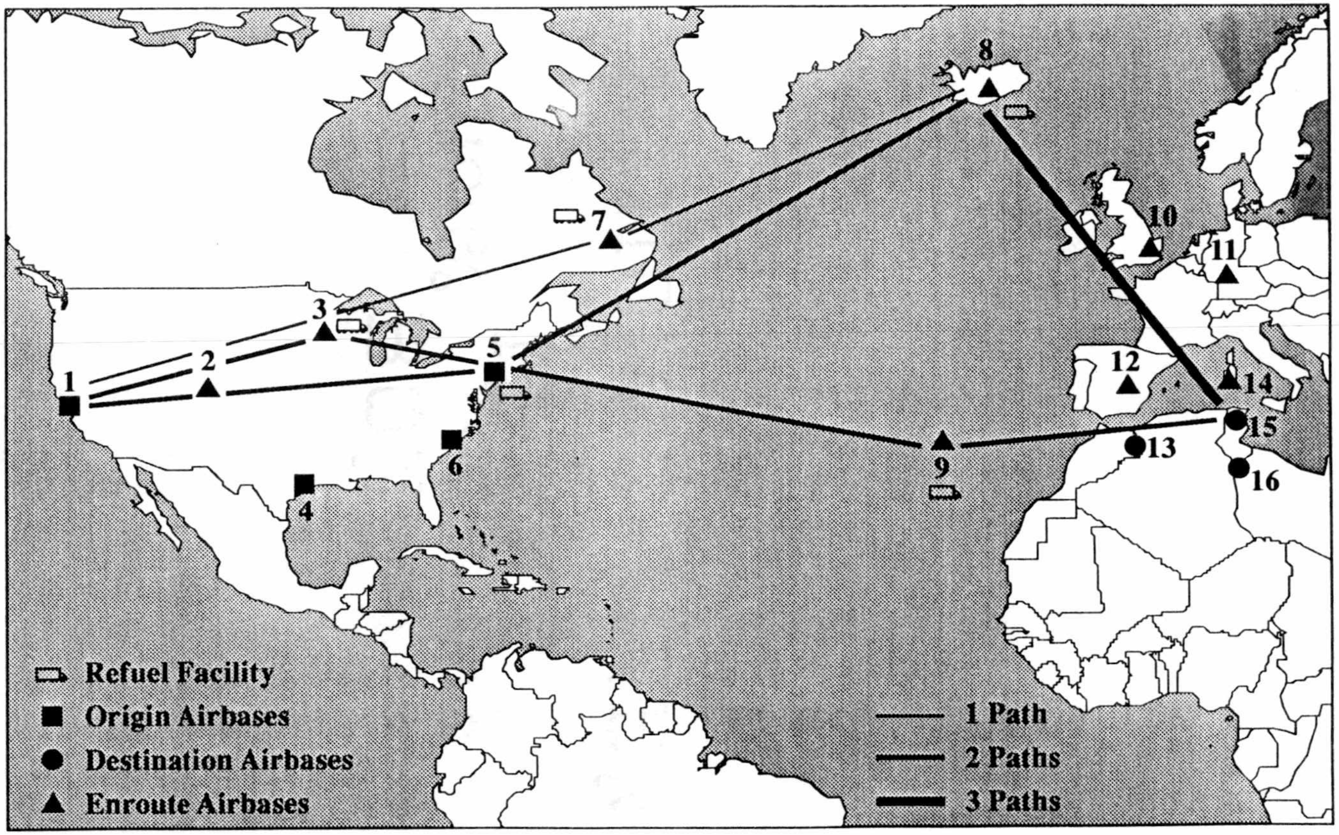


Figure 5.3

Test Problem 2: Solution By Channels

(B. Channel [1,15])

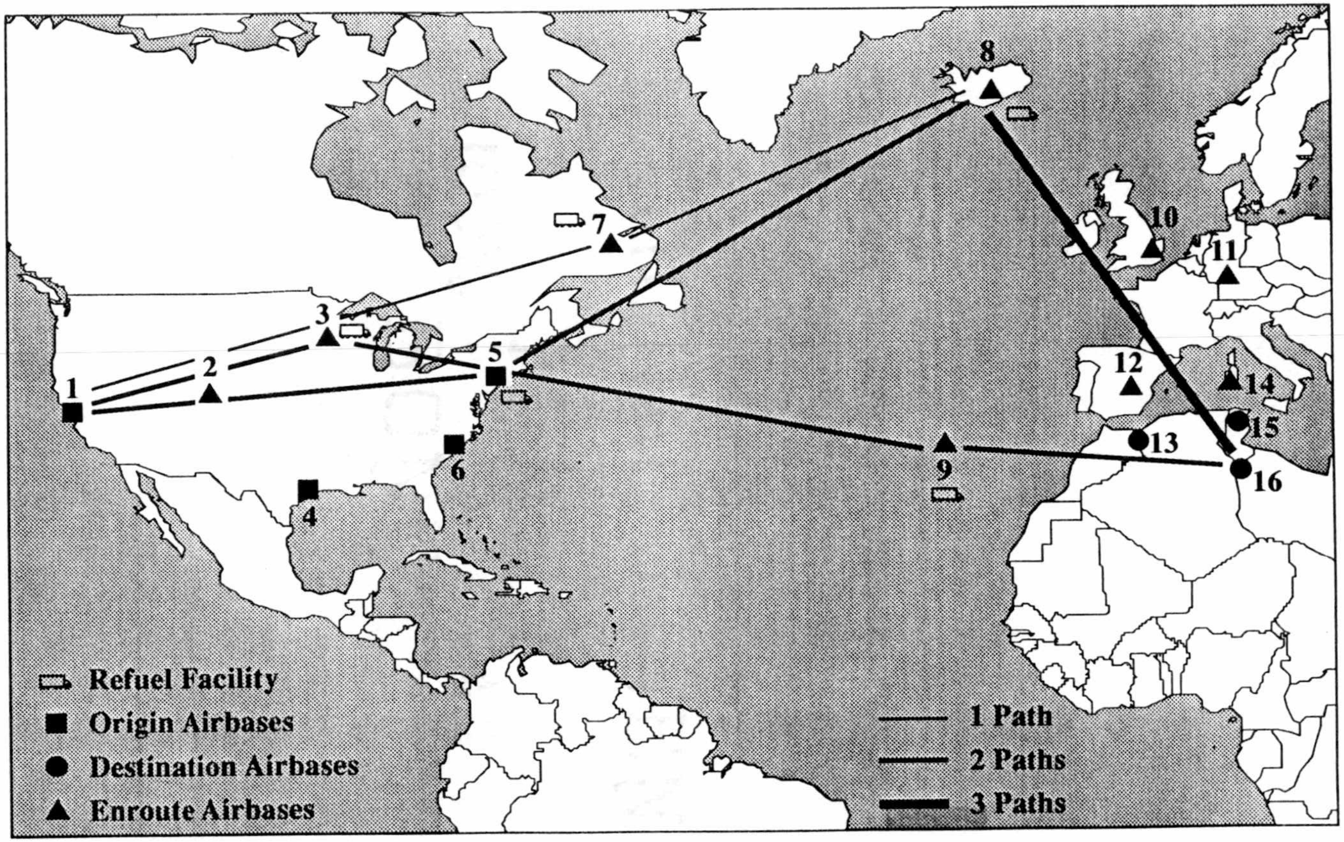


Figure 5.3

Test Problem 2: Solution By Channels

(C. Channel [1,16])

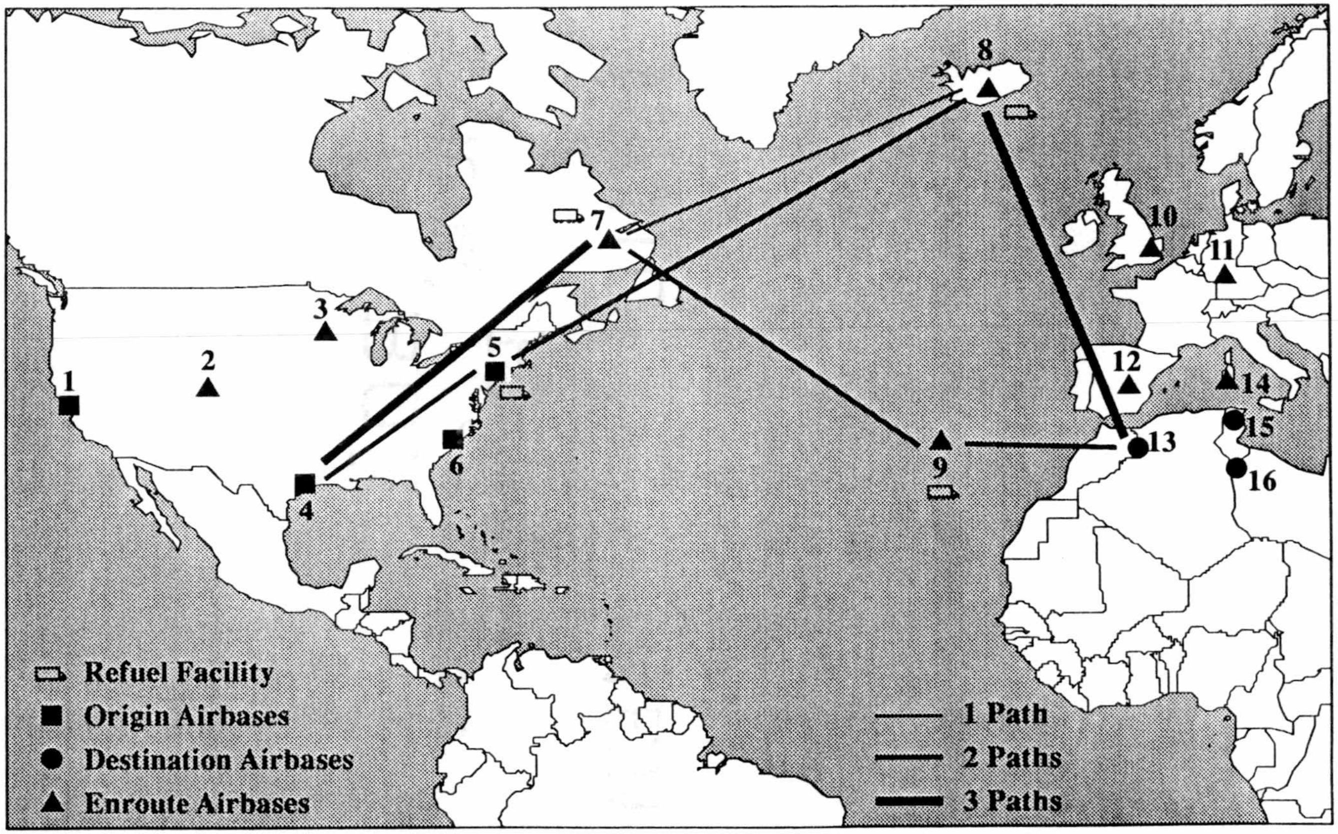


Figure 5.3

Test Problem 2: Solution By Channels

(D. Channel [4,13])

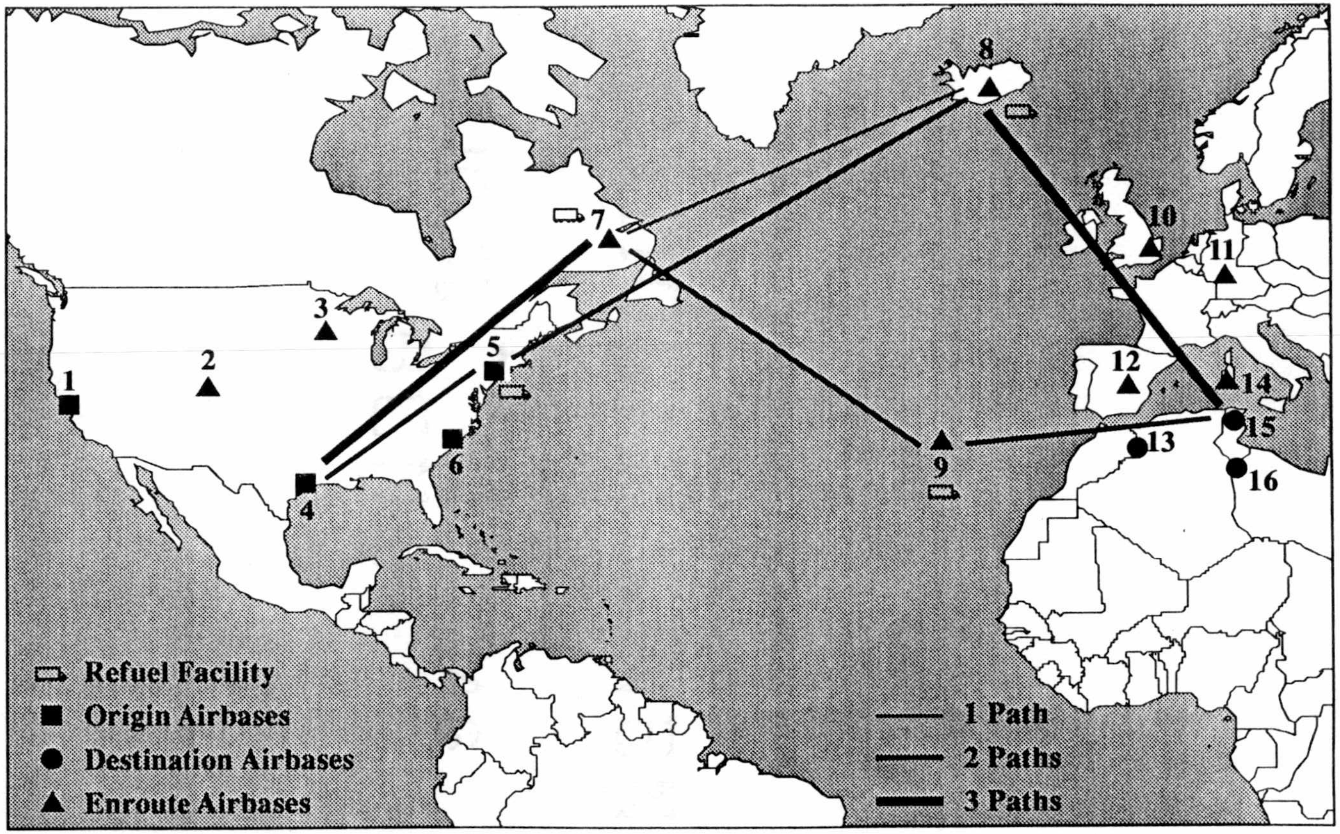


Figure 5.3

Test Problem 2: Solution By Channels

(E. Channel [4,15])

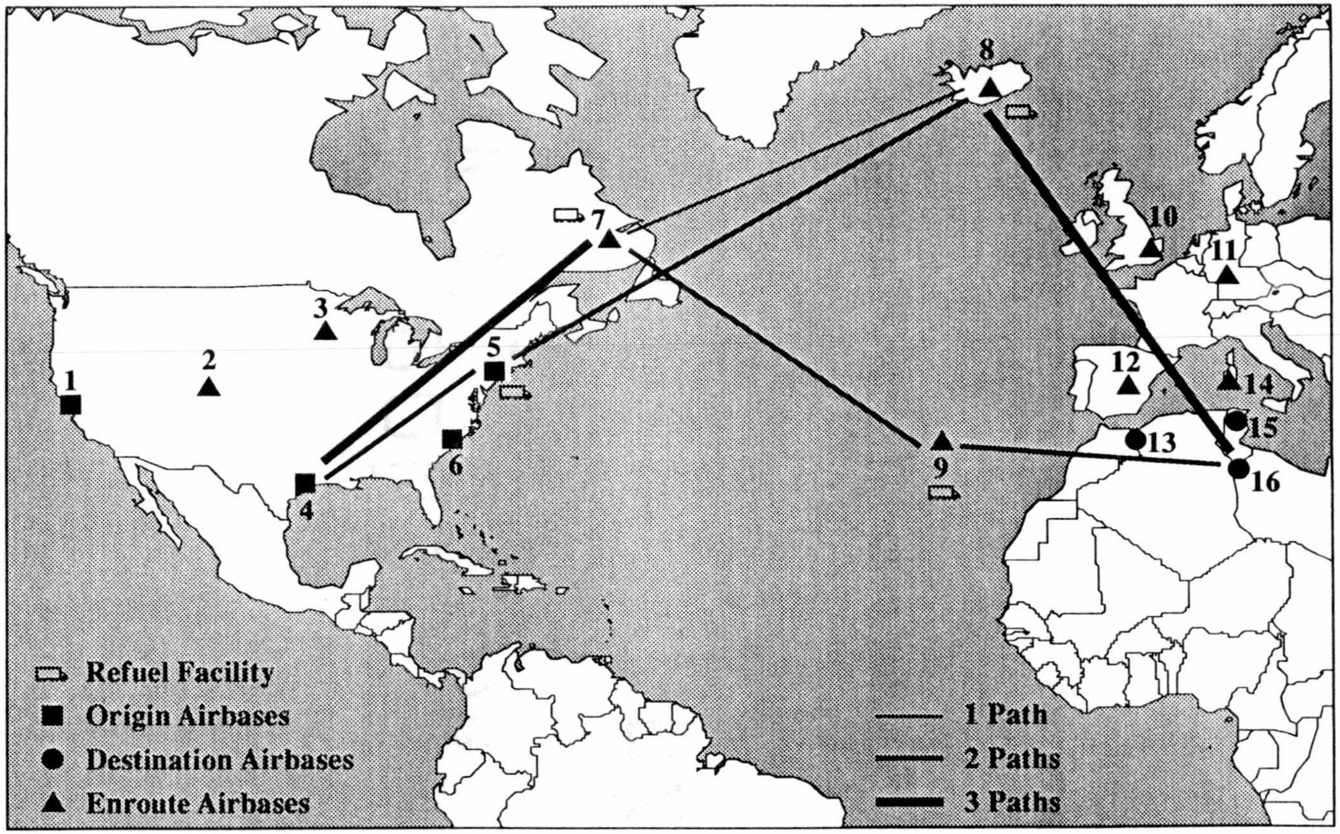


Figure 5.3

Test Problem 2: Solution By Channels

(F. Channel [4,16])

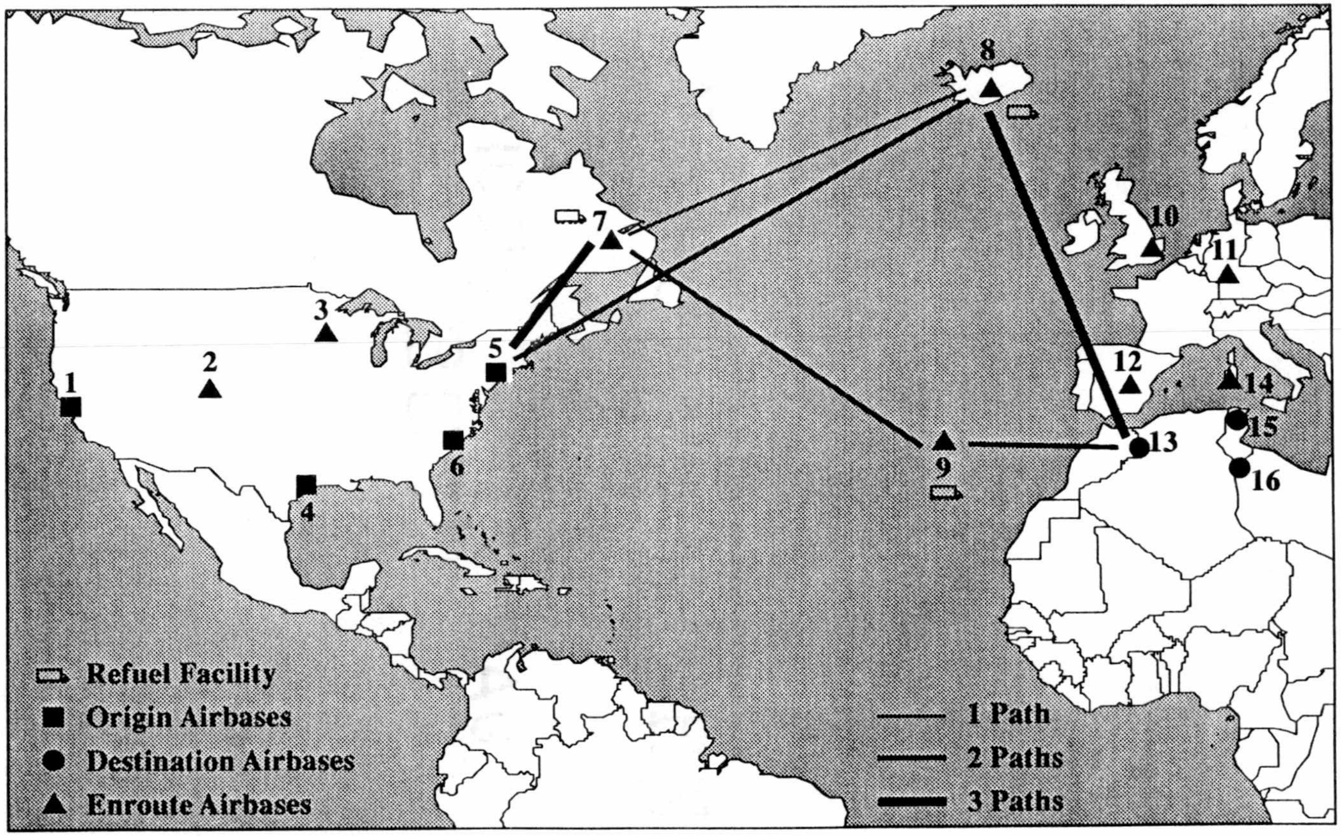


Figure 5.3

Test Problem 2: Solution By Channels

(G. Channel [5,13])

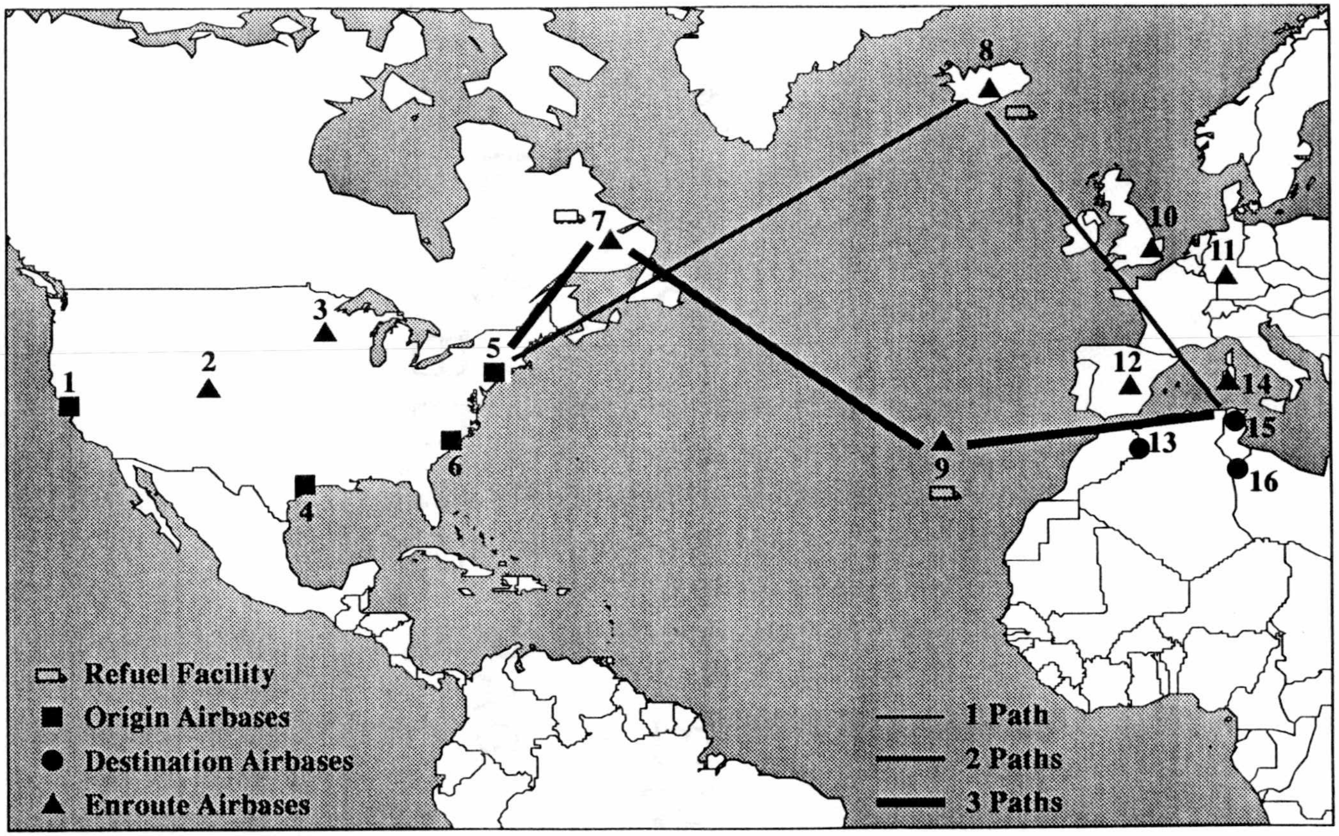


Figure 5.3

Test Problem 2: Solution By Channels

(H. Channel [5,15])

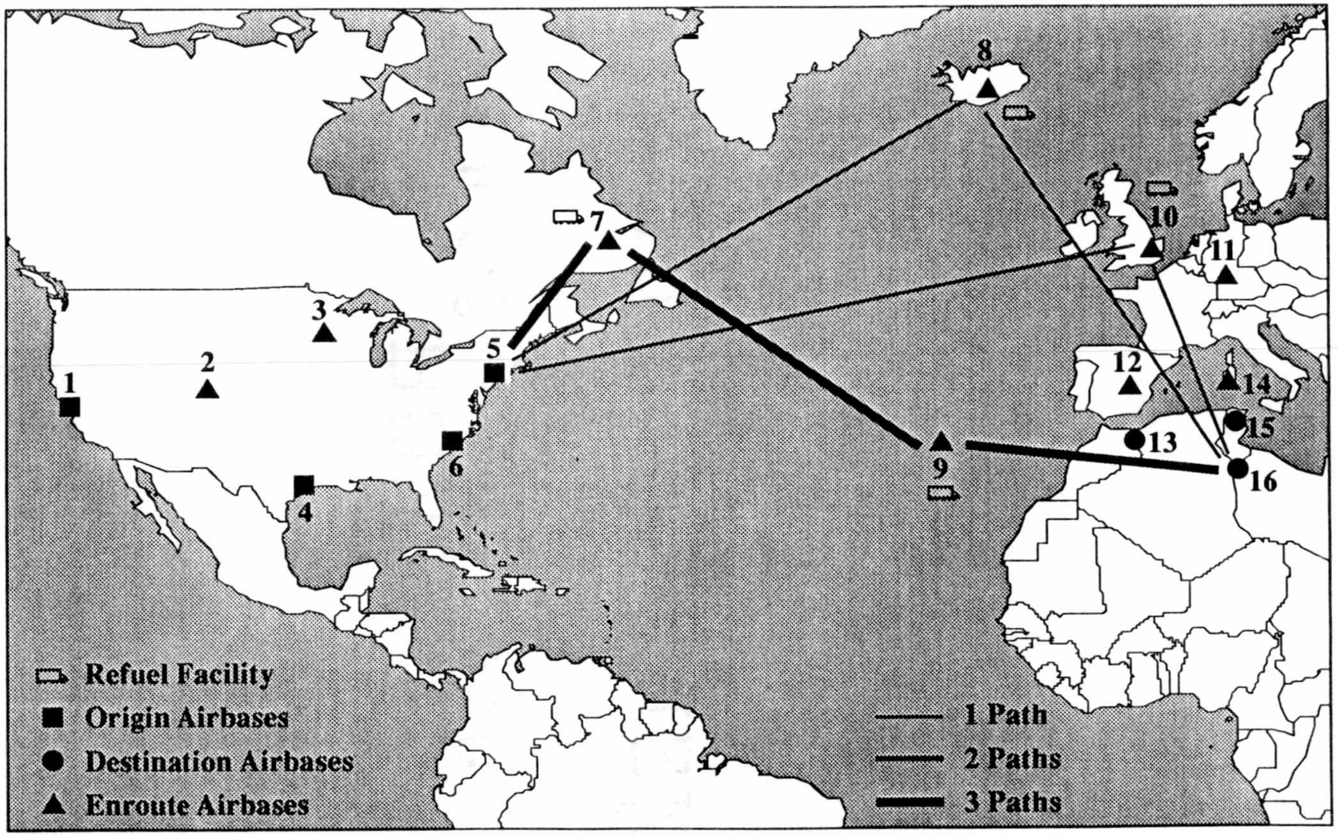


Figure 5.3

Test Problem 2: Solution By Channels

(I. Channel [5,16])

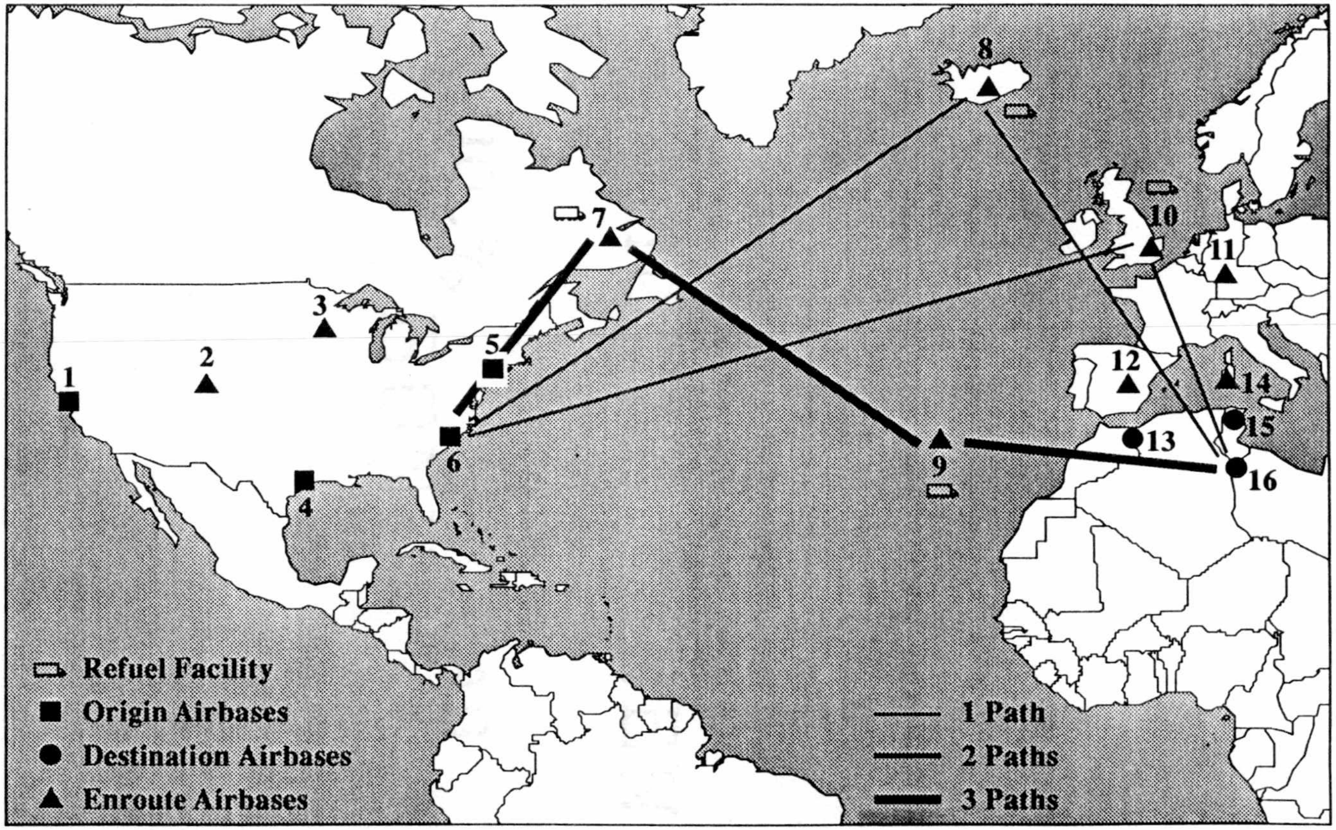


Figure 5.3

Test Problem 2: Solution By Channels

(J. Channel [6,16])

1974) to generate every path between the origin and destination for each channel[5]. Each path, p, was replicated η_p times where η_p is calculated as follows:

(5.1) $$\eta_p = \left\lceil \frac{W_{ij}}{C_p} \right\rceil$$

where W_{ij} is the channel demand, C_p is the path capability and p is a path from v_i to v_j. The optimal value[6] of the objective function for the LP relaxation of (2.3) was found to be 99120.5 dollars. Thus, the solution to test problem 2, as generated by Algorithm II, is shown to be within 1.13% of the optimal integer solution. Algorithm II appears to perform well and generate near-to optimal solutions for the test problem. Moreover, Algorithm II required only 0.2 cpu seconds to achieve this solution with a δ value of 10 whereas the time taken to generate the shortest paths and create and solve the full statement of the LP was over ten minutes. The next section discusses the calibration of δ and its effect on the quality of solutions generated.

5.3. Calibrating δ

The parameter δ is used to approximate the dual costs associated with the last optimal restricted master problem solved. As the true dual solution is unavailable (Section 3.2.1), δ is used to ensure that the surrogate costs associated with each vertex cause the path submodel to behave in a predictable manner.

---

[5] Shier's algorithm was compiled using Sun FORTRAN V2.0 and run on a Sun Sparc 4/360.
[6] The LP relaxation was solved on a Sun Sparc 4/360 using Sun/XA. The LP matrix consisted of 37 rows and 11,479 columns and required 3.2 megabytes of core space to solve it.

As the FRFLP is a maximization problem and the coefficients of the variables associated with the opening of facilities are costs, the dual costs of these variables are expected to be nonpositive. The relationship between the dual costs and the surrogate costs as follows. By adding one more path through the facilities located at some vertex k, one of three cases will occur:

The last facility allocated at v_k has excess capacity and can accommodate an extra path. In this case the dual cost for the facility will be zero since the fixed-charge for opening the facility has already been incurred. Hence, the surrogate cost will also be zero.

The facility has no excess capacity but another facility, which is not currently allocated, could be located at v_k. In this case, the dual cost is expected to be equal to the fixed-charge, Y_k, of opening a facility at v_k. Hence the surrogate cost will also be equal to Y_k;.

No excess capacity exists and no further facilities can be located at v_k. In this case the dual costs are unknown, and the surrogate cost must be estimated.

It is in cases of the third type that the parameter δ is used. Consider the problem of constructing a path through a vertex k which has no excess capacity and no further facilities can be located at v_k. There are two reasons why no further facilities can be located at v_k:

The number of facilities already located at v_k, L_k, is at its upper bound, M_k.

All facilities have been located.

The first case implies that any new path added to the solution will require the removal of a path already in the solution which utilizes a facility at vertex k. In this case, the new path must have a net benefit at least as large as the path utilizing a facility at v_k which has the smallest net benefit. In the second case, for the new path to enter the solution there are two alternatives: (1) an existing path using vertex k must be removed from the solution as in the former case above; or (2) a currently open facility at some vertex $v_{k'}$ must be closed and relocated to vertex k. This means that all paths utilizing that facility at $v_{k'}$ must be removed from the solution. Since each such path has a positive net benefit, the new path must have a net benefit at least as large as the sum of the benefits of all such paths removed.

The parameter δ then acts as a penalty to be incurred when the path submodel attempts to generate a path which would to perturb the set of paths currently in the solution. In essence, when either of the two cases outlined above occurs, the path submodel incurs a cost over and above the fixed-charge for opening a facility. The penalty, levied as a multiple of the fixed-charge, Y_k, attempts to force the path submodel to consider alternative paths through the network.

As δ increases from 1, a progressively higher penalty is incurred. Table 5.4 and Figure 5.4 show the effect of varying δ from a value of 1.0 to 50.0 in increments of 0.25. The data used are the 16-base problem (Test Problem 2) described earlier in this Chapter,

Table 5.4

Objective Function Values For Varying Values Of δ

ITERATION	$1.0 \le \delta < 3.0$	$3.0 \le \delta < 4.0$	$4.0 \le \delta < 4.25$	$4.25 \le \delta < 5.25$	$5.25 \le \delta < 50.0$
1	21300.0	21300.0	21300.0	21300.0	21300.0
2	43000.0	43000.0	43000.0	43000.0	43000.0
3	52000.0	52000.0	52000.0	52000.0	52000.0
4	52000.0	57600.0	63400.0	63400.0	63400.0
5		63300.0	70500.0	70500.0	81100.0
6		70400.0	77600.0	87600.0	94900.0
7		75700.0	84800.0	94900.0	98000.0
8		81100.0	89600.0	98000.0	98000.0
9		86300.0	92800.0	98000.0	
10		88300.0	94700.0		
11		92300.0	94700.0		
12		95500.0			
13		97500.0			
14		97500.0			

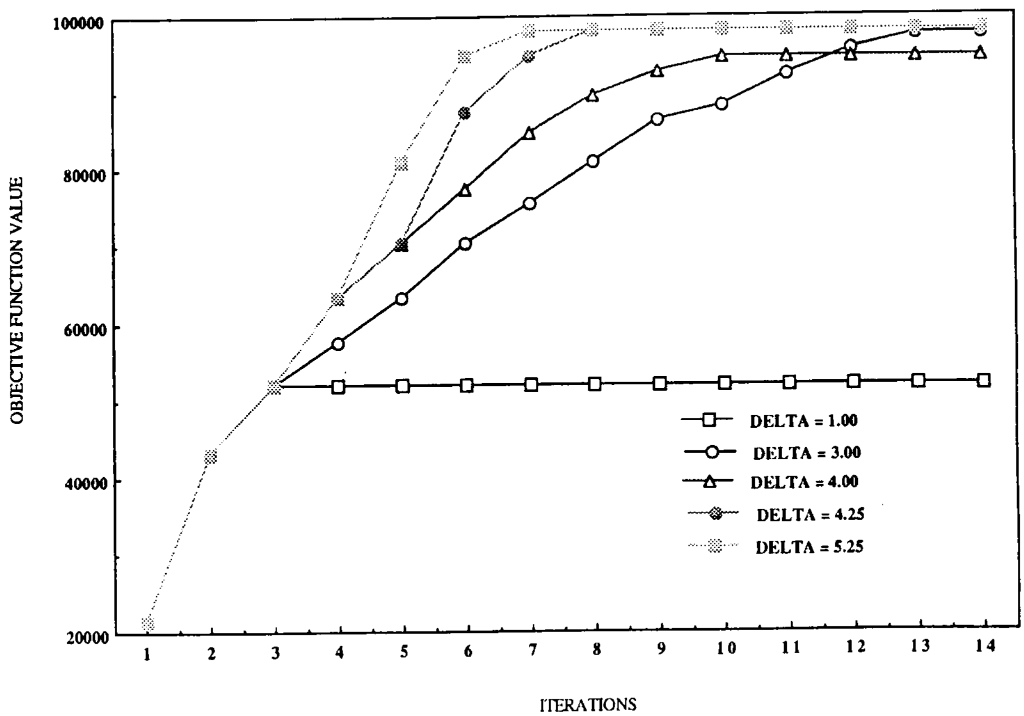


Figure 5.4

Objective Function Values For Varying Values Of δ

and the FRFLP was solved using Algorithm II. At values between 1.0 and 3.0, the path submodel had difficulty in identifying alternative paths in the network. Consequently, the algorithm converged after only three iterations with a final objective value of 52000.0 dollars. With a value of δ between to 3.0 and 4.0, Algorithm II converged at iteration 13 with an objective value of 97500.0 dollars. At values between 4.0 and 4.25, the algorithm converged after ten iterations with an objective value of 94700.0 dollars. At values between 4.25 and 5.25, the algorithm converged after eight iterations with a maximum value of 98000.0 dollars. The best performance was found with a value of δ at or above 5.25. With this value, the algorithm converged after seven iterations with a maximum value of 98000.0 dollars.

In general, it appears that as the value of δ increases, the speed of convergence and the quality of the solutions generated also increase. This occurs since the higher penalties incurred at capacitated facilities force the path submodel to generate alternative paths earlier, thus diversifying the distribution of facilities across the network. However, such is not always the case. Counterintuitively, a value of δ between 3.0 and 4.0 gave a better value of the objective function than a value of δ between 4.0 and 4.25. It is not entirely clear why this is so, but it certainly appears to be linked to the problem of generating alternative paths rather than forcing new paths through already capacitated facilities. To this end, the parameter δ should be calibrated by rerunning the algorithm (Algorithm I or II) with increasing values of δ.

5.4. Summary Of Chapter 5

This chapter used Algorithm II to solve Test Problem 2, a large FRFLP consisting of sixteen airbases and ten channels. The problem was derived from a real-world military airlift problem where the flow-resource facilities being located were aircraft refuelers. Algorithm II was shown to generate a solution that was within 1.13% of the optimal integer solution in reasonable time. A discussion about calibrating the parameter δ followed. The parameter was shown to affect the speed of convergence of the column generation algorithms, as well as the quality of solutions being generated. The next chapter discusses the conclusions drawn from the research contained within this dissertation and examines future extensions of the basic FRFLP.

CHAPTER 6

Conclusions And Final Comments

This chapter concludes the research contained within this dissertation. Section 6.1 contains the conclusions drawn from the study of the FRFLP and section 6.2 examines possible extensions and future work. Section 6.3 contains some concluding comments.

6.1. Conclusions

This section discusses the contribution of the FRFLP model to the field of locational analysis and examines the appropriateness of the two algorithms designed to solve FRFLP models.

6.1.1. The FRFLP As A Design Tool

The FRFLP was designed to address a locational problem faced daily by the U.S. Military Airlift Command. During the design of large-scale operational plans, the location of, and demand for, particular types of airlift resources, such as aircraft refuelers has to be determined. The location of these resources can affect the success of an OPLAN design, since badly located facilities could induce severe bottlenecks at certain airbases in the network. To this end, the FRFLP described in this dissertation is an attempt to mathematically model the network and use location theory to resolve the spatial competition for resources in the network. As stated in Chapter 1, the military currently has no analytical means for solving this problem.

The test problems, and particularly the problem described and solved in Chapter 5, show that the FRFLP model is capable of addressing a combinatorial problem that is difficult to solve. Based on the results obtained from the test problems, the FRFLP appears to be an appropriate design tool for the military OPLAN design and other similar problems. The advantages of the FRFLP model over 'rule-of-thumb' approaches are three-fold:

The solution to the FRFLP describes both the spatial distribution of flow-resource facilities as well as the expected demand for the use of those facilities.

An estimate of the efficiency of a solution to the FRFLP can be achieved by analyzing the residual demand for capacity in the network. Thus, curves depicting the trade-off between the cost (and number) of flow-resource facilities to be located and the net-benefit realized by the carrier attributed to a given value of network capacity can be determined. Such curves would provide carriers with a more complete understanding of the particular flow-resource location problems being addressed.

The impact of changes in a carrier's network can be quickly and easily addressed. For example, changes in the pattern of channel demands, changes in network costs, the impact of a new vehicle type, changes in the characteristics of the flow-resource facilities, and changes in costs and benefits can be readily analyzed.

The discussions and illustrative examples of certain concepts defined in this dissertation have focused on the airlift resource problem, specifically, the distribution of aircraft refuelers. The applicability of the FRFLP is not, however, confined to this one example. Other examples of flow-resource facilities are identifiable, among them: routers in computer networks, lifting stations in sewage systems, and pump stations in pipelines. In general, a distribution system which experiences a decrease in path capacity associated with increasing link length, but for which flow in the network can be boosted by the judicious location of flow-resource facilities can be described as a FRFLP.

6.1.2. Column Generation As An Appropriate Solution Technique

As stated in Chapter 3, column generating algorithms are best applied to linear systems where the number of variables is much greater than the number of constraints. The efficiency of the column generation technique is predicated on its ability to solve the submodel effectively and efficiently, and to achieve convergence with relatively few iterations.

From the formulation of the FLRP as an MIP in Chapter 2 and the discussion following the model description, it is clear that the FRFLP formulation has several features which make solving the problem by direct means virtually impossible for all but trivial problems.

The number of elements of the set of paths, P, in the network is large for even small networks. The discussion in Chapter 2 showed that the cardinality of the set P is exponential in the number of vertices in the network. Consequently, alternative means have to be sought for the solution of large-scale FRFLPs.

Algorithms to generate all non-decreasing ranked paths between two distinct vertices of a network are typically of order $O(n^3)$ where n is the number of vertices in the network (Shier 1974). Thus, the time and storage required to generate the set P of paths in G increases rapidly with problem size.

As each path in P could be used more than once, general integer variables must be used to model each element of the set P. Conventional techniques to solve problems containing such variables involve creating vectors of binary variables for each general integer variable in the formulation. This technique increases the number of discrete variables in the MIP, thus making the MIP more difficult to solve.

The column generation algorithms presented in Chapters 3 and 4 explicitly address the three problems outlined above.

The column generating submodel constructs only those paths which are potential candidates for inclusion into the solution. Thus, many paths which are inferior and would not be used may never be generated. It is possible that paths initially generated are replaced in the solution at a later iteration. Generally, however, the number of paths produced by the column generating algorithm will be less than the complete enumeration.

The labeling algorithm used to solve the path submodel is fast and easily coded. Thus, the subproblem algorithm can generate new paths quickly and efficiently. The order of the labeling algorithm upon which the path submodel is based is $O(n^2)$, which is considerably faster than routines to construct all-shortest-paths. The worst case performance of the column generation algorithm is to generate every path between the origin and destination of every channel. Thus, the set P for the worst case of the column generating algorithm is the same as for the original definition of the set P in Chapter 2. The average performance, however, of the column generation algorithm is expected to be much better than this.

If such a strategy is beneficial, the column generating submodel will replicate paths already generated. Each path generated by the submodel can be modeled using a single binary variable. This removes the need to generate vectors of binary variables for each path in the set P and helps reduce the size of the restricted master problem.

The drawbacks of the column generation approach are as follows:

The column generation algorithm is shown to be optimal for continuous linear models only. As the true dual costs are unavailable from the MIP formulation of the FRFLP, surrogate costs are used to approximate the dual solution. To this end, the algorithms described in Chapters 3 and 4 are restricted to providing heuristic rather than optimal solutions to FRFLPs.

The path submodel used to generate new columns during the column generation algorithm is also shown to be heuristic. An optimal algorithm for multi-criteria path problems of the type required to solve the path submodel does exist (Carraway *et al.* 1990). This algorithm, however, is a generalized dynamic program, and as such, tends to be applicable only for small problems.

The use of surrogate costs to approximate the dual solution to the restricted master problem necessitates the use of a parameter, δ, which must be calibrated for each particular FRFLP. The calibration of δ requires running the algorithm with varying values of δ until no further improvements in the generated solutions occur. Making small changes to the problem definition might necessitate a recalibration of δ. This is more of a problem in Algorithm I where the restricted master problem is solved using branch-bound techniques. A large amount of computer resources may be necessary to solve a single FRFLP in this instance.

In summary, the column generation approach for solving FRFLPs tends to provide good heuristic solutions to a highly combinatorial problem. Optimal solutions to small, trivial problems can be found by solving the complete MIP formulation. This necessitates enumerating all paths in the network and using highly computer intensive branch-bound algorithms. However, no known algorithm exists for optimally solving the FRFLP models presented in this dissertation. To this end, the column generating algorithms of Chapters 3 and 4 are effective and do provide good feasible solutions with relatively little computational effort.

6.1.3. A Comparison of Algorithm I And Algorithm II

Algorithm I and Algorithm II use alternative approaches to solve the restricted master problem. Algorithm I treats the restricted master problem as an MIP with binary variables. The solution strategy is to solve the MIP using conventional branch-bound techniques such as those employed by commercially available solvers. This solution procedure is viable for small problems where the maximum number of binary variables is small (*i.e.* less than 200). However, as the column generation algorithm progresses, the number of binary variables increases at each iteration. Hence, the time required to solve each further iteration is expected to increase. Moreover, the number of iterations and the final number of binary variables is usually unknown for a given FRFLP. Thus, a problem being solved using Algorithm I might become intractable if the size of the MIP exceeds the capacity of the branch-bound solver. The branch-bound solver, however, is guaranteed to produce an optimal solution to the restricted master problem.

In contrast, Algorithm II solves the restricted master problem using a specialized heuristic. The advantages of this approach over that of Algorithm I are as follows:

The heuristic is typically faster than the branch-bound solvers. Test Problem 1 was solved using 5.8 cpu seconds with Algorithm 1 against 0.2 cpu seconds for Algorithm II.

The heuristic maintains an integer-feasible solution at each step. Early termination of the restricted master problem would not, therefore, pass infeasible solutions back to the column generation algorithm.

The greedy and substitution algorithms require a constant amount of storage for a given set of paths. Therefore, if the number of paths in the set P can be held in core, then the restricted master problem can be solved. In contrast, the branch-bound solvers construct a tree structure of unknown breadth and depth which must be held in core. Thus, if the storage required to hold the branch-bound tree exceeds the storage capability of the computer, the restricted master problem would suddenly become intractable and Algorithm I would prematurely terminate.

The major disadvantage of Algorithm II over Algorithm I is that a heuristic solution to the restricted master problem is produced. As both algorithms are heuristic anyway, the problems of sub-optimality while solving the restricted master problem are likely to be outweighed by the speed and size performance of Algorithm II over Algorithm I. To this end, Algorithm II is chosen as the algorithm of choice to solve FRFLPs. The next section discusses possible extensions to the FRFLP model.

6.2. Extensions And Future Work

The MIP formulation of the FRFLP has several features which need to be extended in future work. One of the fundamental assumptions of the FRFLP is that there is an implied benefit to be gained by channeling flow in the network through facilities at centralized locations. For example, it is assumed that a facility which can service six paths, is more beneficial than one which can service at most one path. The former implies a median-type solution, that is, large facilities placed at central locations. The latter implies a more ubiquitous distribution of small facilities. In the real world, however, there are usually benefits associated with increasing economies of scale which are achieved by centralizing facilities. For example, it may be less costly to service many facilities at a single location than to service many locations each with a single facility, since less travel is involved. The formulation of FRFLP in Chapter 2 has no means of incorporating the issue of economies of scale.

The formulation of the FRFLP in Chapter 2 assumes that the flow-resource facilities provide a unit value of support for each path, and similarly, that each path extracts a unit value of service from each facility used. To return to the example of aircraft refuelers, it is assumed that each refueler can service up to Q_k aircraft in some given time interval. That is, each flow-resource facility provides a unit value of service to each aircraft. Consider the alternative problem of locating fuel supplies rather than refuelers. Fuel can be regarded as a continuous flow-resource since the amount of fuel required by an aircraft is a function of the length of the next arc in the path. This problem cannot be addressed by the present formulation.

Lastly, the FRFLP formulation assumes that the commodities or services traversing the network from origins to destinations are moved using a homogeneous transportation medium. In the example of airlift, it is assumed that all the aircraft have similar characteristics. Similarly, in the computer network example it is assumed that all computers are connected, and data is transmitted using the same type of transmission lines. A possible extension to the FRFLP model presented herein would be to allow multiple modes of transportation. For example, two or more types of aircraft could be used in the airlift refueler problem. In such a case, the FRFLP should determine not only the location and demand of the flow-resource facilities but also the demand for vehicle capacity by modal type.

6.3. Final Comments

The human and economic effects of the spatial organization of economic activities has long been a primary focus within geography. A proliferation of theories and models designed to provide insight into the issues associated with spatial decisions have been developed within the last twenty years. The apparent diversity of both the scale of the spatial decisions and the type of activities being located can be attributed to the identification of space as a critical factor in the organization of social and economic activity. As our understanding of the effects of space in decision making processes increases, so can our ability to develop theories and models to predict the human and/or economic consequences of spatial decisions. A visible effect of this increasing insight into the role of space can be seen by the adoption of locational models and theories in areas of social and economic activity which were not previously associated with spatial decision making.

The research contained in this dissertation is an attempt to develop a locational model designed to provide insight into the effects of spatial decision making. The Flow-Resource Facility Location Problem attempts to achieve an efficient solution to a combined location and routing problem where the distribution of flow-resource facilities is dependent on a pattern of path flows in a network. By the same token, the pattern of path flows is dependent on the distribution of flow-resource facilities. The development of the FRFLP model and a solution procedure is a step toward furthering the insight and understanding of the spatial consequences of decision making. However, as the discussions presented in this chapter indicate, many interesting and important extensions to the basic FRFLP model are yet to be addressed.

LIST OF REFERENCES

LIST OF REFERENCES

Balinski, M. L., (1965) "Integer Programming: Methods, Uses, Computation", Management Science, **12**, 253-313.

Berman, O., D. Einav, and G. Handler, (1990) "The Constrained Bottleneck Problem In Networks", *Operations Research*, **38**, 178-181.

Bradley, S. P., A. C. Hax, and T. L. Magnanti (1977) *Mathematical Programming*, Addison-Wesley, Reading, Massachusetts.

Carraway, R. L., T. L. Morin, and H. Moskowitz, (1990) "Generalized Dynamic Programming for Multicriteria Optimization", *European Journal of Operational Research*, **44**, 95-104.

Clarke, G., and J. W. Wright, (1964) "Scheduling Of Vehicles from A Central Depot To A Number Of Delivery Points", *Operations Research*, **12**, 568-581.

Cullen, F. H., J. J. Jarvis, and H. D. Ratliff, (1981) "Set Partitioning Based Heuristics For Interactive Routing", *Networks*, **11**, 125-143.

Desrosiers, J., F. Soumis, and M. Desrochers, (1984) "Routing With Time Windows By Column Generation", *Networks*, **14**, 545-565.

Dijkstra, E. W., (1959) "A Note On Two Problems In Connexion With Graphs", *Numerische Mathematik*, **1**, 269-271.

Feldman, E., F. A. Lehrer, and T. L. Ray, (1966) "Warehouse Location Under Continuous Economies Of Scale", *Management Science*, **12**, 670-684.

Garfinkel, R. S., A. W. Neebe, and M. R. Rao, (1977) "The M-Center Problem: Minmax Facility Location", *Management Science*, **23**, 1133-1142.

Geoffrion, A. M., and G. W. Graves, (1974) "Multicommodity Distribution System Design By Benders Decomposition", *Management Science*, **20**, 822-844.

Gilmore, P. C., and R. E. Gomory, (1963) "A Linear Programming Approach To The Cutting Stock Problem: Part II", *Operations Research*, **11**, 863-888.

Goldman, A. J., (1969) "Optimum Locations For Centers In A Network", *Transportation Science*, **3**, 352-360.

Gomory, R., (1958) "Outline Of An Algorithm For Integer Solutions To Linear Programs", *Bulletin of the American Mathematical Society*, **64**, 275-278.

Hakimi, S. L., (1964) "Optimum Location Of Switching Centers And The Absolute Centers And Medians Of A Graph", *Operations Research*, **12**, 450-459.

Hakimi, S. L., (1965) "Optimum Distribution Of Switching Centers In A Communications Network And Some Related Graph-Theoretic Problems", *Operations Research*, **13**, 462-475.

Hakimi, S. L., and S. N. Maheshwari, (1972) "Optimum Location Of Centers In Networks", *Operations Research*, **20**, 967-973.

Halpern, J., (1978) "Finding Minimal Center-Median Convex Combination (Cent-Dian) Of A Graph", *Management Science*, **24**, 535-544.

Handler, G. Y., and P. B. Mirchandani, (1979) *Location On Networks: Theory And Algorithms*, The MIT Press, Cambridge, Massachusetts.

Hodgson, M. J., (1981) "The Location Of Public Facilities Intermediate To The Journey To Work", *European Journal Of Operational Research*, **6**, 199-204.

Hooker, J. N., R. S. Garfinkel, and C. K. Chen, (1988) "Finite Dominating Sets For Network Location Problems", *Working Paper*, Graduate School of Industrial Administration, Carnegie Mellon University, Pittsburgh, PA.

Jacobsen, S. K., and O. B. G. Madsen, (1980) "A Comparative Study Of Heuristics For A Two-Level Routing-Location Problem", *European Journal of Operational Research*, **5**, 378-387.

Laporte, G., and Y. Nobert, (1981) "An Exact Algorithm For Minimizing Routing And Operating Costs In Depot Location", *European Journal of Operational Research*, **6**, 224-226.

Laporte, G., Y. Nobert, and D. Arpin, (1984) Capacitated Location-Routing Problems, Paper Presented March 1984.

Laporte, G., Y. Nobert, and P. Pelletier, (1983) "Hamiltonian Location Problems", *European Journal of Operational Research*, **12**, 82-89.

Lawrence, R. M., and P. J. Pengilly, (1969) "The Number and Location Of Depots Required For Handling Products For Distribution To Retail Stores In South-East England", *Operational Research Quarterly*, **20**, 23-32.

Levy, L. S., (1987) "The Walking Line Of Travel Problem: An Application Of Arc Routing And Partitioning", Ph.D Dissertation, Management Science and Statistics Department, University Of Maryland.

Lin S., and B. Kernighan, (1973) "An Effective Heuristic Algorithm For The Traveling Salesman Problem", *Operations Research*, **21**, 498-516.

Marsten, R., (1988) *User's Manual For ZOOM/XMP*, XMP Software, Inc., Tucson, Arizona, 1988.

Minieka, E., (1970) "The M-Center Problem", *SIAM Review*, **12**, 138-139.

Minoux, M., (1986) *Mathematical Programming: Theory And Algorithms*, John Wiley & Sons, Chichester, New York.

Montreuil, B., and H. D. Ratliff, (1988) "Optimizing The Location Of Input/Output Stations Within Facilities Layout", *Engineering Costs and Production Economics*, **14**, 177-187.

Murty, K. G., (Forthcoming) *Network Programming*, Prentice-Hall, Inc, Englewood Cliffs, New Jersey.

Nemhauser, G. L., and L. A. Wolsey, (1988) *Integer And Combinatorial Optimization*, John Wiley and Sons, New York, 127-139.

Perl, J., and M. S. Daskin, (1985) "A Warehouse Location-Routing Problem", *Transportation Research B*, **19B**, 381-396.

Revelle, C., and R. Swain, (1970) "Central Facilities Location", *Geographical Analysis*, **2**, 30-42.

Shier, D. R., (1974) "Computational Experience With An Algorithm For Finding The k Shortest Paths In A Network", *Journal of Research of the National Bureau of Standards - B. Mathematical Sciences*, **78B**, 139-165.

Tansel, B. C., R. L. Francis, and T. J. Lowe, (1983) "Location On Networks: A Survey. Part I: The P-Center and P-Median Problems", *Management Science*, **29**, 482-497.

Tansel, B. C., R. L. Francis, and T. J. Lowe, (1983) "Location On Networks: A Survey. Part II: Exploiting Tree Network Structure", *Management Science*, **29**, 498-511.

Tietz, M., and P. Bart, (1968) "Heuristic Methods for Estimating The Generalized Vertex Median Of A Weighted Graph", *Operations Research*, **16**, 955-961.

APPENDIX

APPENDIX

Origin	Destination	Cargo Cap.	Origin	Destination	Cargo Cap	Origin	Destination	Cargo Cap.	Origin	Destination	Cargo Cap.
1	2	25.0	5	7	25.0	8	11	24.0	11	12	25.0
1	3	25.0	5	8	18.0	8	12	22.0	11	13	25.0
1	4	25.0	5	9	20.0	8	13	21.0	11	14	25.0
1	5	25.0	5	10	17.0	8	14	18.0	11	15	25.0
1	6	25.0	6	7	25.0	8	15	19.0	11	16	25.0
1	7	24.0	6	8	18.0	8	16	19.0	12	13	25.0
2	3	25.0	6	9	20.0	9	10	22.0	12	14	25.0
2	4	25.0	6	10	16.0	9	11	21.0	12	15	25.0
2	5	25.0	7	8	20.0	9	12	25.0	12	16	25.0
2	6	25.0	7	9	22.0	9	13	25.0	13	14	25.0
2	7	25.0	7	10	19.0	9	14	25.0	13	15	25.0
3	5	25.0	7	11	18.0	9	15	22.0	13	16	25.0
3	6	25.0	7	12	17.0	9	16	22.0			
3	7	25.0	7	13	19.0	10	11	25.0			
3	9	22.0	7	14	17.0	10	12	25.0			
4	5	25.0	7	15	17.0	10	13	25.0			
4	6	25.0	7	16	17.0	10	14	25.0			
4	7	24.0	8	9	21.0	10	15	25.0			
4	9	21.0	8	10	25.0	10	16	25.0			

VITA

Julian Ray was born in north west England on November 12th, 1962. He attended Culcheth High School in Cheshire, and graduated in June 1981. In September 1981 he entered Reading University and graduated from the university in June 1984 with a B.S. (Honors) in Human Geography. After graduation he took a position with the Greater London Council. In September 1985, he entered The Graduate School of The University of Tennessee. He received a M.S. in Geography in August 1987 and the Doctor of Philosophy degree in Geography in August 1990.

During graduate school, the author has been employed as a research associate with the University of Tennessee Transportation Center and as a subcontractor to the Operations Research Group with the Transportation Center at Oak Ridge National Laboratory. During the course of his graduate work, Mr. Ray has been involved with developing decision models for the U.S. Agency for International Development (U.S. AID), the World Bank, the United States Transportation Command (U.S. TRANSCOM), the United States Military Air Command (U.S. MAC), the Knoxville School Board, and the Federal Emergency Management Agency (FEMA). Following graduation, the author will continue to be employed by the Transportation Center at the University of Tennessee.