To the Graduate Council:

I am submitting herewith a dissertation written by Charles Harold Aikens, III entitled "The Optimal Design of a Physical Distribution System on a Multicommodity Multi-Echelon Network." 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 Management Science.

Richard E. Rosenthal

Richard E. Rosenthal, Major Professor

We have read this dissertation
and recommend its acceptance:

Oscar S. Fowler

Robert S. Garfinkel

Wayne Claycombe

Robert A. McLean

Accepted for the Council:

L. Evans Roth

Vice Chancellor
Graduate Studies and Research

THE OPTIMAL DESIGN OF A PHYSICAL DISTRIBUTION SYSTEM ON A MULTICOMMODITY MULTI-ECHELON NETWORK

A Dissertation

Presented for the

Doctor of Philosophy

Degree

The University of Tennessee, Knoxville

Charles Harold Aikens, III

December 1982

DEDICATED

To My Wife, Helen

ACKNOWLEDGEMENTS

I wish to thank the faculty of the Management Science program for their individual contributions to my education. A special debt of gratitude is owed to Rick Rosenthal who has guided this research from its inception, and who dedicated many tireless hours to the meticulous editing of the dissertation draft.

I would also like to express sincere gratitude to the Walter Melville Bonham Endowment for providing support during the dissertation phase of my graduate studies.

To all the personnel of the Computer Services Unit, I would like to express a hearty thank you.

A special note of gratitude is extended to Paul Wheeler of Kingsport Press, Inc.; Dave Holmes and Steve Cliff of Computer Concepts; and Sir Lionel Macrae, without whose enthusiastic and complete cooperation the completion of the case study material for the research would not have been possible.

I would like to express appreciation to Lindsey Barker, Ian Langdon, and all the members of the Governing Council, Darling Downs Institute of Advanced Education, for their continuous support since the commencement of my doctoral program.

I am indebted to Art Geoffrion, UCLA, for providing motivation and inspiration to get the project started, and to Jerry Brown, Navy Postgraduate School, for his generosity in permitting the use of the GNET and XNET subroutines and for many helpful comments.

My sincere appreciation is extended to Georgia Bunn for a superb typing effort on a very difficult manuscript and for her interest and dedication in meeting some tight deadlines.

Finally, I will be forever indebted to my wife, Helen, and children, Christopher and Jocelyn, who never lost faith in my ability to succeed and without whose love, tolerance, patience, and understanding I would have soon lost the resolve and persistence necessary for completion.

ABSTRACT

Strategic planning in physical distribution management requires decisions as to where intermediate distribution centers should be located, what sizes they should be, and how commodities should be routed from sources of production through the distribution centers and on to customers. The literature concerning the location of distribution facilities is extensive, covering a wide range of modeling options. This work has developed an optimization procedure for solving a multi-commodity multi-echelon distribution/location problem with upper and lower bounds imposed on distribution center throughput.

An implicit enumeration algorithm is presented which uses a heuristic to generate an initial throughput-feasible solution. The algorithm employs a new hybrid price-directive/resource-directive scheme for efficiently solving the multicommodity capacitated transshipment problems which result from temporarily fixing the network configuration at each node of the enumeration tree. A number of branching and screening options are used in order to keep the tree search small. These have proven to be very effective in practice.

A software package, to implement the algorithm, is described and computational results are reported on a large-scale practical application involving national distribution of ice cream products in Australia.

TABLE OF CONTENTS

CHAPTER PAGE

LIST OF FIGURES

INTRODUCTION

Physical Distribution Overview

Physical Distribution Management (PDM) is defined by the National Council of Physical Distribution Management (NCPDM) [62] as

> . . . the term describing the integration of two or more activities for the purpose of planning, implementing and controlling the efficient flow of raw materials, in-process inventory and finished goods from point of origin to point of consumption. These activities may include, but are not limited to, customer service, demand forecasting, distribution communications, inventory control, materials handling, order processing, parts and service support, plant and warehouse site selection, procurement, packaging, return goods handling, salvage and scrap disposal, traffic and transportation and warehousing and storage.

This view embraces a kaleidoscope of activity, ranging from the procurement and receipt of raw materials, to the storage of materials and work-in-process, to the delivery of finished products. Financing the distribution infra-structure can represent a major drain on resource budgets and, ironically, many firms have little idea of the true magnitude of these costs. Sawdy [125] explains:

> There are two reasons for the lack of information. One is that conventional accounting systems do not isolate the costs of distribution. . . . The second reason is that, even if distribution costs are classified separately, they often do not provide information of economic significance.

Notwithstanding the lack of hard data at the company level, one can turn to the findings of several recent studies for an awareness of what proportion of the nation's economy is preoccupied with the movement of goods from point of production to point of consumption.

Estimates of distribution costs in Australia, based on 1974 data, and compiled by the Productivity Promotion Council of Australia (PPCA),

account for an alarming 15% of the Gross Domestic Product [118]. These costs have been broken down as follows:

	$ Million
Transport (17.9%)	1,100
Inventory (25.7%)	1,580
Order Processing (14.9%)	920
Warehousing (12.7%)	780
Packaging/Storage (11.9%)	730
Receiving and Dispatch (10.0%)	610
Administration (7.0%)	430
Total	6,150

Firth et al. [29] report similar results for Canada and, provide the following breakdown of costs as reflected by data collected from more than 250 firms:

Industry Type	Annual Costs $ Million	Distribution Costs $ Million	Distribution Costs as % of Annual Costs
Copper Refinery	100	55	55%
Plumbing and Heating Products	30	5.8	20%
Electrical Hardware	60	8.4	14%
Packaged Foods	120	34.5	28%
Brewery	45	12	27%
Bakery	45	16	36%
Pharmaceuticals	16	2.4	15%
Petroleum Products	45	13	29%

In a U. S. study, sponsored by the NCPDM, cost estimates were put at 20% of the Gross National Product [110]. The authors of this report courageously predict that $40 billion dollars in annual savings (or 10% of total distribution expenditures) could be recouped by U. S. industry through more judicious management of the physical distribution process.

This economic backdrop is the driving motivation behind the present work. Better management processes will spawn improved managerial performance. The evolution of management process is strongly linked to and dependent upon the tools of analysis available to the decision makers.

Tools of Analysis

Over a period of several decades management scientists have firmly established a niche for providing answers to problems which are "hard" to solve--hard by virtue of complex intervariable relationships, stochastic behavior, and/or sheer size. This latter characteristic is of particular relevance to distribution modeling. The typical distribution/location problem requires solution approaches which can accommodate thousands of decision variables. Where decision makers once had to rely heavily on intuition and experience alone to solve these problems, the use of exact or near-exact mathematical programming models has become widely accepted as decision aids which supplement this judgment. In a recent edition of a general management textbook, Koontz, O'Donnell, and Weihrich selected a case study in distribution planning as a scenario to illustrate how management science tools can provide valuable practical assistance in complex decision making [93].

To introduce the present work and see where it fits in the context of physical distribution planning, it is useful to reflect on the various types of decisions which distribution planners make. Eilon [20] classifies the problems in PDM three ways: strategic, tactical, and operational. At the strategic level planning is for the long

term. Such issues as market evaluation, profitability analysis, pricing policy, diversification vs. specialization, inventory strategy, and the use of computers are all strategic in nature. This thesis is concerned with the design of distribution configurations, that is, determining the size, number, and location of distribution centers (DC's). This is a strategic issue, and it is not something which a firm is likely to do often.

Tactical decisions are for a shorter term and include such activities as warehouse layout, vehicle replacement, the design of materials handling systems, manpower scheduling, formulation of maintenance procedures, and planning vehicle routes. The operational level is concerned with day-to-day activities, such as dispatching, repairs, handling queries and complaints, invoicing, expediting, and order processing.

Effective distribution planning involves developing an overall strategy which results in the minimum total cost of distribution--that is, the sum of the fixed costs which largely result from strategic decisions and the variable costs which are, to a large degree, influenced by tactical and daily operational decisions. It is important to note that, whilst the issues which this work seeks to resolve are strategic, the objective function is a mixture of cost elements arising from all the levels of decision making. In some contexts the concept of "sole-sourcing" (that is, the enforcement of a rule requiring the assignment of each customer to a single and fixed point of supply for all of a "bundle" of commodities [56] can also impinge on the cost function.

Meidan [104] has characterized solution approaches to location problems two ways:

> (a) The Infinite Set--assumes that the warehouse can be positioned anywhere on the map, obvious slight adjustments can be made later to allow for rivers, mountains etc. A main assumption with this set is that transport costs are directly proportional to distance (crow fly) and this is questionable in many situations.
>
> (b) The Feasible Set--assumes a finite number of possible locations and these costs can be calculated with a high degree of precision, both costs of buildings and haulage. The obvious problem with this method is one of cost and time employed in collating the necessary data but the expected benefits of an accurate analysis can easily offset this initial outlay.

Additionally, location models may be static or dynamic depending on whether they deal with a fixed point in time or a planning horizon. Most of the work in distribution/location modeling has concentrated on methodologies which are static and which fit in Meidan's Feasible Set category (that is, a pre-determined finite set of candidate warehouse locations is known). Some limited work in dynamic modelling has been successful (see for example [90, 113]). Broadly, decision tools may be classified in the following ways:

- Simulation
  - Static
  - Dynamic
- Heuristics
- Mathematical Programming
  - Static
  - Dynamic

Simulation offers a kind of real world experimentation which is attractive when more exacting techniques are either unavailable or otherwise infeasible. The complexities and interdependencies which

characterize physical distribution system behavior have motivated the development of some commercial simulators designed to handle distribution decision making in an integrated fashion and in a dynamic context. Powerful industrial logistics simulators, such as the Long Range Environmental Planning Simulator (LREPS) developed by Systems Research Incorporated [9] can be used to model the daily operating activities of the total distribution system. With the LREPS simulator it is possible to test and evaluate alternate order processing, warehousing, inventory management, and production strategies as they affect prescribed performance criteria. LREPS is incapable of answering strategic location questions.

Simulation can undoubtedly provide valuable management insights. However two major disadvantages limit their usefulness in some situations:

(1) Cost--simulation models are expensive to implement. Since simulation is based on statistical sampling techniques sufficient replications must be run in order that the sample size yield acceptable conclusions.

(2) Error--simulation results always contain sampling error.

Geoffrion and Powers [60] assert that, largely, industrial distribution planning is achieved through the use of heuristics owing to the high combinatorial complexity of realistic optimization models. Like simulation, heuristics also contain errors since such techniques often have shaky, if any, theoretical underpinnings. Surprisingly, the major shortcoming of either a heuristic or a simulator as a planning tool does not lie so much in its inability to provide accurate

and optimum solutions, but in the practical limitations of such tools to assist with sequential decision making--that is, reaching valid conclusions from a sequence of computer runs with parameter and input variations. Such inter-run comparisons are essential, particularly in strategic decision-making. It is well understood that the use of optimization modeling provides insights which assist the decision maker's judgment, not replace it [45]. Management must be able to pose "What If?" questions and have the model respond in an adaptive way. It is the parametric analysis of alternative policies that arms the decision maker with clearer insights and understanding, not the objective function value itself, optimum or not. Even when using an optimization method the solution may not be the true optimum since solution accuracy is vulnerable to data inaccuracies.

As Geoffrion and Powers [60] articulate:

> The problem is not that optimization capability is needed to cope with the staggering number of alternatives for distribution system design, although this is important. It is not that optimization capability is needed to resolve the cost tradeoffs inherent in distribution planning, although this too is important. It is not even that managers would rather have the best answers possible from their planning support systems, although certainly this is compelling. Rather, the crux of the matter is that optimization capability is needed to permit reliable comparisons between different runs of the model. If the solver is heuristic in character, such comparisons will be very unreliable because comparing two error-prone solutions greatly magnifies the relative error. Erroneous conclusions are likely to result.

In another source Geoffrion and Van Roy [61] demonstrate that heuristics may not only fail to achieve the best answer, but may often fail badly if the algorithm stops prematurely.

With such pitfalls, why have heuristics been so widely preferred to optimization techniques? The answer possibly lies in the historical inability of computer technology and management science theory to respond to real world problems in a timely and cost effective way. Certainly heuristics have in many cases overcome the computational problems that would have been encountered through the use of an optimization capability. It would appear that due to the combinatorial complexity of optimization models, the state-of-the-art has been slow in catching up with the needs of practitioners. However, today strong alliances between computer specialists and management scientists, armed with a powerful new computer technology, have led to new breakthroughs. As a result, optimization is now a serious contender in many new applications. This thesis reports one of these success stories. It is hoped this work will lead to others.

Thesis Organization

Chapter I of this thesis introduces a case study of a major Australian firm engaged in the manufacture and marketing of ice cream and frozen food products. The case is presented early because it provides a frame of reference for all the ensuing chapters. The case was a vital ingredient throughout the research project, not only for inspiration, but also as a source of realistic data for field testing the model proposed.

Chapter II reviews the relevant literature concerning distribution/location modeling.

Chapter III provides an explicit definition of the problem which is the subject of the present work and explains how this version of the problem differs from and/or extends the work of others. The present work was preoccupied mainly with extracting a solution for the totally linear model. It is asserted that nonlinearities can be introduced utilizing a piece-wise linearization variant of the software reported in this thesis. Nonlinear aspects are discussed fully in Chapter VIII.

Chapter IV explores possible solution approaches for solving the Fixed Charge Multicommodity Flow Problem, the class name which applies to the linear version of the problem at hand. Benders Decomposition and Branch and Bound are the two approaches considered. Both are looked at in depth together with possible methods for achieving tight upper and lower bounds.

Chapter V presents a new algorithm called the multi-echelon distribution optimization system (MEDOS). This algorithm employs a rather complicated heuristic (also the product of the present work) to achieve an initial tight upper bound. Then, a branch and bound implicit enumeration scheme is used which calls upon a number of fathoming rules to try to keep the enumeration tree small. Bounding is achieved through the use of a hybrid price directive/resource directive approach. A pricing loop drives an infeasible solution towards feasibility. Upon termination of the pricing loop, an infeasible, but near-feasible, solution enters a resource allocation loop which achieves feasibility and drives toward optimality.

Chapter VI describes the MEDOS software package, a major contribution of the present work.

Chapter VII revisits the Australian case and reports on the computational results achieved by applying MEDOS to this very large scale problem.

The final chapter (Chapter IX) offers suggestions for extending the work presented in this thesis.

CHAPTER I

AN AUSTRALIAN CASE STUDY: NATIONAL DISTRIBUTION OF ICE CREAM PRODUCTS

During the period May 1980-August 1981 a study was conducted of a major Australian firm which manufactures and distributes a variety of frozen food products. This case has served both as a model for the present work, and as a source for inspiration and data.

1.1 Background

This company is a principal manufacturer of ice cream and dairy products and, in addition, acts as an agency in the distribution of a range of non-ice cream frozen foods for other manufacturers. At the time of the investigation, management interest in physical distribution was particularly keen, brought about partly by several recent and major policy changes:

1. A shift from conventional to highly automated high rise cold store warehousing.
2. A merger with another large Australian firm which doubled the size of the national distribution network.
3. A shift in the marketing strategy for small customers. A system of taking pre-orders by telephone and processing those orders by computer had replaced from-the-van selling by drivers in large metropolitan areas.

The pertinent data is provided in detail in the Appendix. Using the MEDOS software described in Chapter VI, this data was run on the

DEC-10 computer system at The University of Tennessee, Knoxville. These results are reported in Chapter VII. The next four sections of this chapter will describe the major components of the distribution network: factories, commodities, distribution center locations, and customers. Figure 1.1 places this system in a geographical perspective.

1.2 Factory Network

After the merger the network consisted of the following seven ice cream plants:

Queensland: Brisbane, Townsville

New South Wales: Grafton

Victoria: Melbourne, Preston

Tasmania: Hobart

Western Australia: Perth

1.3 Commodities

The inventory for manufactured items of ice cream products represented over 100 distinct products; the non-manufactured items elevated the total range to approximately 500. This number takes into account not only distinct products, but flavor variations and pack sizes. For distribution management purposes it was possible to group large numbers of line items together so that the total number of product groupings could be reduced to thirteen according to the following broad classifications:

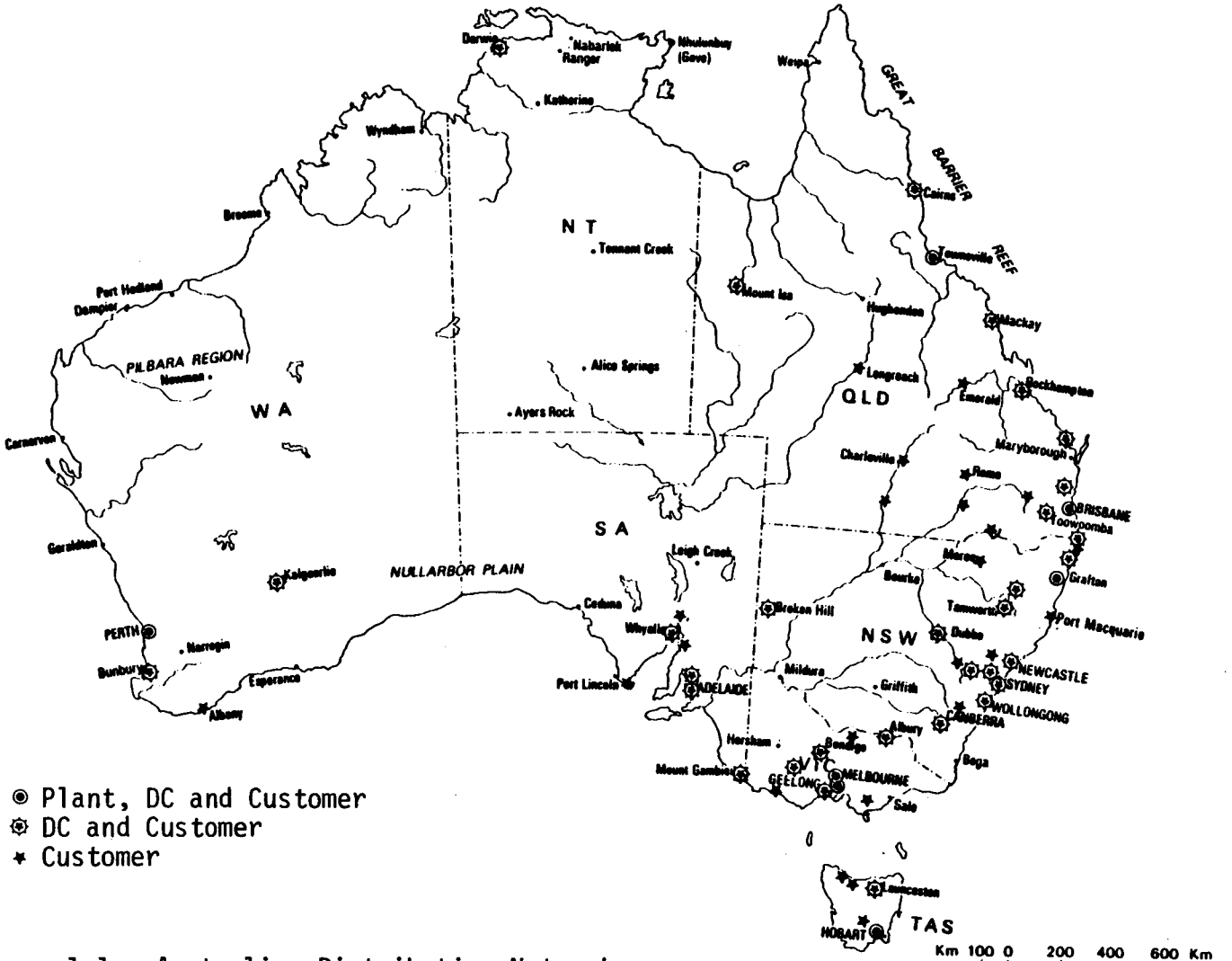

Figure 1.1. Australian Distribution Network.

Manufactured Products

1. Bulk Ice Cream and Confectionaries
2. Take Home Ice Cream
3. House Brand
4. Loose Pack Stick/Novelty Items
5. Take Home Stick/Novelty Items

Non-manufactured Products

6. Vegetables
7. Fish
8. Poultry
9. Sara Lee Products
10. Cultured Dairy Products
11. Cones
12. Frisco Food Products
13. Prepared Foods

The present work was only concerned in designing a network for the manufactured products. The omission of the other product groupings is reflected in the capacities and demand data used. It may be noted that the MEDOS algorithm can handle the non ice-cream segment by fixing the production capacities (upper and lower limits) at infinity and zero levels respectively for these products.

1.4 Distribution Center Location

A total of forty-three candidate DC locations (refer to Figure 1.1), including existing facilities, were selected in consultation with top management. The existing centers have been marked with an asterisk

in the following list. Additionally a distinction has been made between major and minor DC's--the principal difference being in size and the fact that major DC's may receive supplies from any production facility, whilst minor DC's may or may not be permitted to receive supplies direct from a factory, and, if so, then only from a very small subset of the total production facilities. That is, minor DC's rely primarily on major DC's for supply.

List of DC Candidates

Location	Major	Minor
Southern Queensland		
1. Brisbane*	x	
2. Toowoomba*	x	
3. Tweed Heads	x	
4. Nambour	x	
5. Bundaberg*	x	
Central Queensland		
6. Townsville*	x	
7. Rockhampton*		x
8. Mackay*		x
West Queensland		
9. Mt. Isa*		x
North Queensland		
10. Cairns*		x
Northern Territory		
11. Darwin*		x

Location	Major	Minor
Northern New South Wales		
12. Lismore		x
13. Grafton*		x
14. Armidale		x
15. Tamworth		x
Central New South Wales		
16. Newcastle*	x	
17. Bathurst		x
18. Wollongong	x	
19. Parramatta	x	
20. North Sydney	x	
21. East Sydney*	x	
22. West Sydney	x	
23. South Sydney	x	
West New South Wales		
24. Broken Hill*		x
25. Dubbo		x
South New South Wales		
26. Albury	x	
Australian Capital Territory		
27. Canberra*	x	
South Victoria		
28. Geelong*	x	
29. North Melbourne*	x	
30. East Melbourne	x	

Location	Major	Minor
South Victoria (continued)		
31. West Melbourne	x	
32. South Melbourne	x	
Central Victoria		
33. Ballarat*	x	
34. Bendigo	x	
South Australia		
35. Adelaide*	x	
36. Whyalla		x
37. Mt. Gambier		x
38. Elizabeth		x
Western Australia		
39. Perth*	x	
40. Kalgoorlie		x
41. Bunbury		x
South Tasmania		
42. Hobart*	x	
North Tasmania		
43. Launceston*		x

1.5 Customers

Customers fall in one of seven different classifications:

1. Major Retailer Warehouses. Bulk shipments are made to the warehouses of several major grocer chains.
2. Contract Warehousing. The warehousing function is performed for certain retail chains. Orders are taken by telephone

and shipments made direct to the retail stores.

3. Metropolitan Small Shops. Customers in this group are primarily sole proprietorships and include ice cream bars, delicatessens, corner shops--in essence, "mom and dad stores." In the Australian economy, such stores are numerous; for example, in Brisbane, there are approximately 2300 customers in this group. Sales in this group have, in the past, been essentially for ice cream products. With the advent of telephone selling and the preorder concept, a market is emerging for the non-ice cream lines. Deliveries are made using a fleet of company owned vans.
4. Catering and Food Service Areas. Customers in this group exist within the areas served by major DC's. Orders are filled on a preorder basis and deliveries are made by a fleet of small company owned trucks.
5. Export--Papua New Guinea and Pacific Islands. Shipments are in container loads or smaller quantities by air or sea.
6. Small Orders. Customers in this grouping include small shops, schools, organizations, etc. that cannot be serviced by the normal distribution network (e.g., located in an isolated or remote area). Orders are packed in dry ice in special cartons and consigned to the customer by bus, rail, or truck.
7. Staff Sales. Employee stores are operated in certain locations.

For the purposes of this investigation, the export market, which represents a very small percentage of total sales, was ignored. The

remaining salesmarkets were aggregated and grouped according to the seventy-four geographical regions, shown in Figure 1.1, and listed below.

Customer Listing

1. North Brisbane	23. Darwin	45. Albury
2. East Brisbane	24. Lismore	46. Canberra
3. South Brisbane	25. Grafton	47. Geelong
4. West Brisbane	26. Armidale	48. North Melbourne
5. Toowoomba	27. Tamworth	49. East Melbourne
6. Tweed Heads	28. Murwillumbah	50. West Melbourne
7. Nambour	29. Port Macquarie	51. South Melbourne
8. Bundaberg	30. Moree	52. Lilydale
9. Dalby	31. Newcastle	53. Warnambool
10. Maryborough	32. Bathurst	54. Morwell
11. St. George	33. Wollongong	55. Moe
12. Goondiwindi	34. Parramatta	56. Ballarat
13. Townsville	35. North Sydney	57. Bendigo
14. Rockhampton	36. East Sydney	58. Shepparton
15. Mackay	37. West Sydney	59. Adelaide
16. Mt. Isa	38. South Sydney	60. Whyalla
17. Charleville	39. Liverpool	61. Mt. Gambier
18. Cunnamulla	40. Penrith	62. Elizabeth
19. Longreach	41. Goulburn	63. Port Pirie
20. Emerald	42. Orange	64. Port Augusta
21. Roma	43. Broken Hill	65. Port Lincoln
22. Cairns	44. Dubbo	66. Perth

67. Kalgoorlie	70. Hobart	73. Burnie
68. Bunbury	71. New Norfolk	74. Devonport
69. Albany	72. Launceston	

1.6 Stock Keeping Unit

Due to the differences in the size and type of customer served, warehouse systems have had to be designed to enable the selection of a single item or bulk items. Consequently, the definition of a single stock keeping unit (SKU) was difficult. For the bulk customer, it makes sense to talk in terms of wraps or pallet loads. For the mom and dad shop, single cartons are more appropriate, and single items are often desired in catering and food service sales. Consultation with management produced an agreement on the "liter" as the best least common denominator volume unit of measurement. Since the loose pack and take home stick and novelty items (product groups 4 and 5) are counted by the dozen, appropriate constants had to be used to convert the volume for these products to liters.

1.7 Transportation

Road transport is the principal mode of transportation used, as trucks provide door-to-door service thereby minimizing the risk of product spoilage. Rails are used occasionally, but the savings in freight costs are generally not felt by management to justify the risk caused by delays and multiple handling. Sea is the mode usually employed for the export market. Transportation costs used in this study are based on over-the-road transport charter rates, with full loads, and in most cases, trailers which are block stacked (without

pallets). However, some customers require palletized shipments and the transportation costs used reflect this. Depending on the link, and the contract carrier, trucks quoted ranged from 18 foot trailers with 6000 kg capacity to 40 foot with 18,000 kg capacity. An 18,000 kg load is equivalent to approximately 20 pallets or 14,000 liters of product.

1.8 A Distribution/Location Problem

Top management interest in the physical distribution function was acute, since expenditures related to transportation and warehousing activities represented a significant slice of the total operational budget (approximately $30 million per year). A consulting firm, under contract, had made some cost effective recommendations concerning distribution and warehousing procedures, and many of these suggested improvements had been implemented. However, at the time of this study, no formal investigation into the network topology had taken place.

The location, size, and number of intermediate distribution centers, as well as the procedure utilized for the dispatch and routing of commodities over the distribution network is a critical management issue, not only for this firm, but for many other Australian firms as well. The importance of moving consumer goods efficiently and effectively nationwide can be better appreciated when placed in the context of Australia's rather unique demography. Australia is largely an urban society with 60% of the total population residing in the six capital cities. The remaining 40% are scattered over the inhabited regions of the country. Many of these residents are farmers or ranchers who live on large properties in very sparsely populated

areas. In comparative terms a population approximately one-twentieth the size of the United States is dispersed over a land area similar in size. For several decades, the Australian people have enjoyed a quality of life comparable to any nation of the free world. As technology has produced new products for labor or leisure, Australian industry has been quick in responding to the local demand for these new commodities. The maintenance of such a high standard, which, with few exceptions, has been available to everyone--even those in remote areas, has placed a special burden on the private sector in terms of distribution management.

The ice cream and frozen foods industry is an excellent example of how Australian enterprise has risen to the task of supplying the needs of its populous. Many parts of Australia are arid and cannot support the local production of agricultural and dairy products. Even if the geography of such areas could sustain agrarian production, the sparsity of demand would render commercial ventures uneconomical. The problem, then, has been how to feed the residents of such remote locations at a reasonable consumer cost.

The location/distribution model proposed in this thesis is designed to assist in solving this problem. The expected output from processing this firm's data through the model is an optimal solution which specifically will provide:

1. a list of intermediate distribution centers which should be established. The implication of this output is that any existing DC on the list should be retained, any DC not on the list should be closed and the asset liquidated, and

any DC on the list, but not in the existing configuration, should be established.

2. the commodity routings over the network as defined by the optimal configuration. These routings provide information on annual aggregate production schedules which may influence plant layouts and organization, order consolidation and dispatch procedures, and warehouse organization and design.
3. the total minimum distribution system cost realizable from the implementation of the optimal configuration and commodity routings.

Customer service, which is vital to the ice cream and frozen foods industry, has not specifically been addressed by the model. Penalties related to response times can be built into the transportation rates, but this alone may not be an adequate safeguard to insure that non-quantifiable customer service objectives are met. It cannot be over-emphasized that MEDOS is a decision tool designed to extend the capabilities of the decision maker, not preempt them. It is clearly understood that management will always retain its right to override, modify, or veto the model-produced solution. In this spirit the model can provide the rationale for such management actions, as computer replications provide an ever increasing understanding of the problem as inter-run comparisons are made, each time posing relevant "What If?" questions (e.g., "What if the DC in Toowoomba were locked open permanently?").

The major anticipated benefits to be derived from applying MEDOS to the ice cream and frozen foods scenario are:

1. to provide, for strategic planning purposes, a physical distribution configuration blueprint. This network in effect will constitute a plan which can be accepted outright, rejected outright, or accepted with modification. Therefore, the model solution provides at best the "ideal" topology and at least an advanced starting point for planning discussions.
2. to provide an "order-of-magnitude" estimate of the potential savings which could be realized through a change from the existing configuration.
3. through the preparation of model input and the examination and evaluation of model output, to obtain a greater depth of understanding of the underlying cost structure and the sensitivity of total costs to certain management policy decisions (e.g., centralization vs. de-centralization, mode of shipment trade-offs, etc.).

It is important to note that the procedures developed in the research work reported in this dissertation are applicable to a wide range of industrial types (e.g., imported consumer products, manufactured goods, raw materials, primary production, etc.).

CHAPTER II

LITERATURE REVIEW

The warehouse location problem with its conjugate distribution management issues is not new to the science of operations research. A rich and colorful branch of the literature has established a heritage which spans well over two decades. As with most classes of problems in management science, the distribution/location family has many members. Models in this class range in complexity from very simple single commodity linear deterministic formulations to multicommodity nonlinear stochastic versions. It is the purpose of this chapter to review some of the more significant work which has contributed to the present state of knowledge. The sections have been organized with the general intent of progressing from the simplest through the more complex in increasing order of solution complexity. This will not always follow and, the order presented may be debated by some, since the more complex models have individual salient characteristics that render solution processes challenging. This chapter is not intended to prioritize these differences. Conversely, the objective is to illustrate the taxonomic evolution that has taken place and the ever-increasing role of management science in the distribution planning field.

Models in the distribution/location class can be broadly classified according to,

1. Whether the distribution network is capacitated or uncapacitated (arcs and/or nodes).

2. The number of echelons (zero, single, or multiple) of transshipment points existing between supply nodes and demand nodes.
3. The number of commodities (single or multiple).
4. Whether costs are linear or nonlinear (arc and/or node costs).
5. Whether the planning horizon is fixed (static) or permitted to vary (dynamic).

2.1 Uncapacitated Simple Facility Location Model

The simplest location model, which for convenience will be called the Uncapacitated Facility Location Model (UFL) has the following formulation:

$$\text{(UF)} \quad \text{Minimize} \quad \sum_{i\in I} \sum_{j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i z_i$$

Subject to

$$\sum_{i\in I} x_{ij} = 1 \qquad j\in J \qquad (2.1)$$

$$z_i - x_{ij} \geq 0 \qquad i\in I,\ j\in J \qquad (2.2)$$

$$x_{ij} \geq 0 \qquad (2.3)$$

$$z_i \in \{0,1\} \qquad (2.4)$$

where

x_{ij} = the proportion of customer j's demand is satisfied by facility i.

z_i = $\begin{cases} 1 \text{ if facility i is established.} \\ 0 \text{ otherwise.} \end{cases}$

c_{ij} = the total production and distribution costs for supplying all of customer j's demand from facility i.

f_i = fixed cost of establishing facility i.

I,J = the sets of candidate facility sites and customer zones respectively.

(UFL) is a single commodity, uncapacitated, zero-echelon, linear model. Constraints (2.1) insure that each customer's demand is satisfied exactly, and constraints (2.2) insure that customers receive product only from open facilities. The model is zero-echelon as there are no transshipment points. The facilities to be located are the supply points (either plants or warehouses), and how the supply reached these facilities is of no concern to the model.

Numerous approaches have been proposed for solving (UFL). The earliest attempts were through the use of heuristics. The Kuehn and Hamburger [94] pairwise interchange, or "bump and shift," routine, although almost two decades old, is contemporary in the sense that the Kuehn and Hamburger battery of twelve test problems has been adopted as a generic standard against which algorithmic designers compete for computational efficiencies. Kuehn and Hamburger utilize the following three heuristic concepts:

1. The best candidate locations will be near demand concentrations.
2. Near optimum results can be achieved by opening those warehouses one-at-a-time which produce the greatest cost savings for the entire system.
3. Only a small subset of all candidate locations need to be investigated in order to determine the next warehouse to open.

A main program locates warehouses one-at-a-time until no more warehouses can be opened without increasing total system costs. Then a "bump and shift" routine investigates configuration modifications,

specifically evaluating the profit implications of closing or relocating open warehouses.

Efroymson and Ray [19] were the first to propose branch and bound as a technique for solving (UFL). By reformulating the problem, Efroymson and Ray were able to express the linear programming bounding problem in a form that can be solved by inspection. Specifically,

Let F_j = the set of plant indices that can supply customer j.

C_i = the set of customer indices that can be supplied from plant i.

n_i = the cardinality of C_i.

K_0 = the set of binary variables set to zero.

K_1 = the set of binary variables set to one.

K_2 = the set of binary variables which are uncommitted.

The problem becomes

$$\text{(UFL')} \quad \text{Minimize} \quad \sum_{i\in I} \sum_{j\in C_i} c_{ij}x_{ij} + \sum_{i\in I} f_i z_i$$

Subject to

$$0 \le \sum_{j\in C_i} x_{ij} \le n_i z_i \qquad i\in I$$

and (2.4).

If constraints (2.4) are relaxed, (UFL') is a linear programming problem which has an optimal solution given by

$$x_{ij} = \begin{cases} 1 \text{ if } c_{ij} + (g_i/n_i) = \underset{k\in K_1 \cup K_2}{\text{Min}} \; [c_{kj} + (g_k/n_k)] \\ 0 \text{ otherwise} \end{cases}$$

$$z_i = (1/n_i) \sum_{j\in C_i} x_{ij} \qquad i\in K_2$$

$$g_k = \begin{cases} f_k & k \epsilon K_2 \\ 0 & k \epsilon K_1 \end{cases}$$

The obvious advantage to this formulation is the speed with which bounding problems can be solved. The authors report computation times for linear test problems with 50 plants and 200 customers, of ten minutes on the IBM 7094. These results are impressive even on an early generation computer.

Spielberg [127] approached the problem from a different perspective. In Spielberg's implicit enumeration scheme, all facilities are either initially opened or closed. At each node, two solutions can be generated which will always be feasible. One solution (υ') is obtained by dropping the fixed charges for any facility not used in the subproblem solution (flow of zero). A second feasible solution (υ'') is obtained by solving a linear programming problem with all free variables open and the integer restrictions on the x variables relaxed, then rounding up all fractional values on the x's. If min (υ',υ'') < υ^* where υ^* is the incumbent, set $\upsilon^* \leftarrow$ min (υ',υ''). Spielberg reported computational results on a range of problem sizes. Albeit a much simpler problem an example is reported similar in order of magnitude to the Chapter I case study. This example concerned 60 candidate plant locations and 80 customers. The reported results for three problems of this size were: 150 CPU minutes on an IBM 360/50, halted short of optimality at 12,200 iterations from a cold start; 60 CPU minutes on an IBM 360/50, solved to optimality with 9975 iterations, starting with an initial incumbent; and 60 CPU minutes on an IBM 7094, halted short of optimality after 103,000 iterations.

Khumawala [87] has also been a notable contributor to this class of models, principally in the development of efficient branching rules for branch and bound. Khumawala [87] has proposed four criteria of branch selection, each embracing a pair of rules. These will be briefly described.

· Delta Rules

1. Largest Delta--select the free warehouse which has the largest delta from the set of warehouses having a negative delta.
2. Smallest Delta--select the free warehouse which has the smallest delta from the set of warehouses having a negative delta.

where, using the notation defined previously,

$$\Delta_i = \sum_{j \in C_i} \nabla_{ij} - f_i$$

and

$$\nabla_{ij} = \min_{k \in F_j \cap (K_1 \cup K_2); k \neq i} [\max(c_{kj} - c_{ij}, 0)]$$

∇_{ij} represents the difference in cost between opening free warehouse k and some other free closed warehouse.

If $\Delta_i \geq 0$, $z_i = 1$ for all completions.

If $\Delta_i < 0$, a large delta implies the warehouse is more likely to be open at the terminal node. A small delta implies the warehouse is less likely to be open at the terminal node.

· Omega Rules

1. Largest Omega--select the free warehouse which has the largest omega from those having positive omega.
2. Smallest Omega--select the free warehouse which has the smallest omega from those having positive omega.

where

$$\Omega_i = \sum_{j \in C_i} w_{ij} - f_i$$

and

$$w_{ij} = \min_{k \in F_j \cap K_l} [\max(c_{kj} - c_{ij}, 0)]$$

w_{ij} is similar to ∇_{ij}, only the cost difference is computed just with respect to those warehouses which are fixed open.

· Z Rules*

1. Largest Z--select the free warehouse with the largest z from the set having fractional z;
2. Smallest Z--select the free warehouse with the largest z from the set having fractional z;

where z is the binary variable.

· Demand Rules

1. Largest Demand--select the free warehouse which can supply the largest total demand.
2. Smallest Demand--select the free warehouse which can supply the smallest total demand.

---

*Khumawala actually referred to these as Y Rules to be consistent with his notation of y for the binary variables.

Khumawala tested the rules on a CDC 6500 computer and found the Largest Omega rule to perform the best and the Smallest Omega to be the poorest. The Demand rules were generally poor performers, Largest Z worked better than Smallest Z, and both Delta rules generally did not perform well.

Erlenkotter [21] has reported very impressive computational success using a multiplier adjustment method. Instead of solving (UFL) directly, Erlenkotter solves a condensed dual in which the dual of (UFL) is reduced to a form involving only the multipliers corresponding to constraints (2.1). The dual ascent procedure starts with some initial dual solution and adjusts the multipliers incrementally in such a way that complementary slackness violations are reduced. The algorithm terminates when no further adjustments are possible (that is, any further increases in the dual variables will violate one or more of the dual constraints). Erlenkotter's method has worked very well for problems with 100 plants and 100 customers; reported solution times are well under 5 seconds on the IBM 360/91 computer, including output but excluding input time.

Gizelis and Samoulidis [63] offer an interesting variant of Efroymson and Ray's algorithm in which the fixed charges are allowed to vary. Khumawala and Whybark [90] use branch and bound to solve a dynamic version of (UFL). Other recent work on the simple facility location model is reported by Laporte and Nobert [96], Morris [105], and Negelhout and Thompson [113].

2.2 Uncapacitated Plant and Warehouse Location Model

Kaufman, Eede, and Hansen [81] propose an algorithm which solves, using branch and bound, a more general two-level distribution system requiring the simultaneous location of plants and warehouses, or warehouses of different sizes. The mathematical formulation for this Simple Uncapacitated Plant and Warehouse Location Model (UPW) is the following:

$$\text{(UPW)} \quad \text{Minimize} \quad \sum_{i\in I}\sum_{j\in J}\sum_{k\in K} c_{ijk}x_{ijk} + \sum_{i\in I} f_i z_i + \sum_{j\in J} g_j y_j$$

Subject to

$$\sum_{i\in I}\sum_{j\in J} x_{ijk} = 1 \qquad k\in K \qquad (2.5)$$

$$\sum_{j\in J} x_{ijk} \le z_i \qquad i\in I,\ k\in K \qquad (2.6)$$

$$\sum_{i\in I} x_{ijk} \le y_j \qquad j\in J,\ k\in K \qquad (2.7)$$

$$z_i \le y_i \qquad i\in I \qquad (2.8)$$

$$x_{ijk} \ge 0 \qquad i\in I,\ j\in J,\ k\in K \qquad (2.9)$$

$$y_j, z_i \in \{0,1\} \qquad i\in I,\ j\in J \qquad (2.10)$$

where i,j,k index plants, warehouses, and customers respectively, and I,J,K are the corresponding sets.

This model fundamentally differs from (UFL) in the triple subscripting and the double set of binary variables. Constraints (2.5) are analogous to (2.1) requiring the satisfaction of demand. Constraints (2.6) and (2.7) correspond to (2.2), assuring in the former case that shipments only originate from plants which are open and in the latter case are only shipped through open warehouses. The triple subscripting

qualifies (UPW) as a single-echelon model. It is important to note that constraints (2.8) require that a warehouse be located wherever a plant is located.

Kaufman, Eede, and Hansen's [81] algorithm, which is a generalization of the work of Efroymson and Ray [19], computes the cost reductions which would occur, relative to a particular configuration, if each of the free facilities were opened. Since facilities with positive reduced costs cannot possibly lead to an improved solution, such facilities can be locked closed in any "next-lower-level" completion. The ability to compute net changes in cost which will occur for all completion actions immediately leads to a lower bound on completions (that is, the most optimistic outcome of opening a free facility is the cost of the current configuration less the maximum savings which could accrue from any completion). If the lower bound exceeds the incumbent, the node may be fathomed. Kaufman et al. [81] designed a fictitious problem in order to test the algorithm. A series of cases were run, varying the number of warehouses, plant locations, and fixed costs. A typical problem which consisted of five plants, 30 warehouse sites, and 50 customers, required approximately 26 CPU seconds on a CDC 6500 computer, excluding input and output time.

2.3 Multicommodity Uncapacitated Plant Location Model

Warszawski [132] was one of the first to address multicommodity aspects. Warszawski's contribution was a generalization of (UFL) in that not only locations, but commodities could be differentiated as well. This Multicommodity Uncapacitated Facility Location Model (MUF) has the following formulation:

(MUF) Minimize $\sum_{i\in I}\sum_{j\in J}\sum_{p\in P} c_{ijp}x_{ijp} + \sum_{i\in I}\sum_{p\in P} f_{ip}z_{ip}$

Subject to

$$\sum_{i\in I} x_{ijp} = 1 \text{ if } D_{jp} > 0 \qquad j\in J,\ p\in P \qquad (2.11)$$

$$\sum_{p\in P} z_{ip} \leq 1 \qquad i\in I \qquad (2.12)$$

$$\sum_{j\in J} x_{ijp} \leq \sum_{j\in J} z_{ip} \qquad i\in I,\ p\in P \qquad (2.13)$$

$$x_{ijp} \geq 0 \qquad i\in I,\ j\in J,\ p\in P \qquad (2.14)$$

$$z_{ip}\in\{0,1\} \qquad i\in I,\ p\in P \qquad (2.15)$$

where p indexes commodities, and i and j index plants and customers respectively. D_{jp} is the demand for commodity p by customer j. x_{ijp} and c_{ijp} are defined as with (UFL) and (UPW) except for the addition of the commodity subscript.

Constraints (2.11) require that demand be satisfied and constraints (2.13) insure that customers can only receive shipments from open facilities. It is noted that (MUF) is not in reality a multicommodity problem since constraints (2.12) limit each facility to a single commodity. This formulation was motivated by a large construction project which required the three major commodities: concrete, building blocks, and reinforcing steel. It was necessary to locate manufacturing or fabrication plants for each of the three commodities in such a way that 38 customers (destinations) could be serviced most efficiently.

Warszawski's paper [132] includes a discussion of a branch and bound procedure and also a heuristic. No computational results are provided for the branch and bound algorithm as Warszawski concludes that

large problems would consume excessive computer time. Results are reported on the heuristic. The problem described above was solved on an IBM 360/50 computer in 138 CPU seconds including compilation, and it is assumed, also input/output.

Karkazis and Boffey [80], combined some of the concepts of Bilde and Krarup [8] and Erlenkotter [21], to develop two dual-based approaches for solving (MUF). These approaches appear to surpass Warszawski's algorithm, computationally. Khumawala and Neebe [89] have devised tighter lower bounds for Warszawski's algorithm and Neebe and Khumawala [112] have suggested an efficient branch and bound algorithm for solving this problem.

2.4 Simple Multistage Plant Location Model

Warszawski [132] also proposed some solution strategies for a dynamic version of (UFL) in which supply node locations could change over time. This Multistage Location Problem, like (MUF), was motivated by the large construction project application cited above. Sites for concrete mixing, the manufacture of building blocks, and the cutting, bending, and storage of reinforcing steel can change as the requirements for these materials differ during various stages of construction. The multistage single-commodity plant location model has the following formulation:

$$\text{(MSL)} \quad \text{Minimize} \quad \sum_{i\epsilon I} \sum_{j\epsilon J} \sum_{s\epsilon S} c_{ijs}x_{ijs} + \sum_{i\epsilon I} \sum_{s\epsilon S} b_{is}z_{is}$$

$$+ \sum_{i\epsilon I} a_{i1}z_{i1} + \sum_{i\epsilon I} \sum_{s\epsilon S'} a_{is}z_{is}(1-z_{i,s-1})$$

Subject to

$$\sum_{i\epsilon I} x_{ijs} = 1 \text{ if } D_{js} > 0 \qquad j\epsilon J,\ s\epsilon S \qquad (2.16)$$

$$\sum_{j\epsilon J} x_{ijs} \leq \sum_{j\epsilon J} z_{is} \qquad i\epsilon I,\ s\epsilon S \qquad (2.17)$$

$$x_{ijs} \geq 0 \qquad i\epsilon I,\ j\epsilon J,\ s\epsilon S \qquad (2.18)$$

$$z_{is}\epsilon\{0,1\} \qquad i\epsilon I,\ s\epsilon S \qquad (2.19)$$

where

a_{is} = the installation or setup cost incurred in establishing a supply source at location i in stage s.

b_{is} = the fixed cost associated with supply source i which is independent of relocations (e.g., capital and maintenance costs).

and

s indexes the stage, S = {s}, and S' = S - {1}.

A distinction must be made between the two cost components defined above. The cost b_{is} is always incurred for an established source, whilst a_{is} is only incurred if the source is relocated from stage (s-1) to stage s.

The objective function of (MSL) contains the total transportation costs over all stages plus the fixed charges which are location independent summed over all stages and all open facilities, plus the initial setup costs for all facilities, plus the sum of the setup costs incurred due to facilities changing location from one stage to the subsequent stage.

For fixed s, the constraint set (2.16)-(2.19) will be recognized as being nearly identical to that of (UFL), the only difference being the substitution of the weaker constraint (2.17) for the tighter formulation utilizing constraint (2.2). Warszawski tested two methods for solving (MSL); an approximate dynamic programming recursion, and a procedure based on highest marginal savings. Typical run times on an IBM 360/50 computer for four stages, nine locations, and 16 destinations were 146.64 seconds using the dynamic programming formulation and 38.84 seconds using the highest marginal savings method.

2.5 Capacitated Facility Location Model

Problem (MUF) can be converted into a capacitated model by placing upper limits (capacities) on the supplies. The resulting problem, (CFL), has the following form:

$$\text{(CFL)}\quad \text{Minimize} \quad \sum_{i\in I}\sum_{j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i z_i$$

Subject to

$$\sum_{j\in J} D_j x_{ij} \leq S_i z_i \qquad i\in I \qquad (2.20)$$

$$\sum_{i\in I} x_{ij} = 1 \qquad j\in J \qquad (2.21)$$

$$x_{ij}\in[0,1] \qquad i\in I,\ j\in J \qquad (2.22)$$

$$z_i\in\{0,1\} \qquad i\in I \qquad (2.23)$$

Up to this point all formulations have treated the continuous flow variables (the x's) as measures of the proportion of total demand satisfied by the respective facilities under consideration. An important alternative formulation treats the continuous variables as units of

flow. For example, redefining the continuous variables in (CFL) in terms of units of flow and adjusting the cost coefficients appropriately, (CFL) can be transformed into the equivalent form:

$$\text{(CFL)'} \quad \text{Minimize} \quad \sum_{i\in I} \sum_{j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i z_i$$

Subject to

$$\sum_{j\in J} x_{ij} \leq S_i z_i \qquad i\in I \qquad (2.24)$$

$$\sum_{i\in I} x_{ij} = D_j \qquad j\in J \qquad (2.25)$$

$$x_{ij} \geq 0 \qquad i\in I,\ j\in J \qquad (2.26)$$

$$z_i \in \{0,1\} \qquad i\in I \qquad (2.27)$$

Constraints (2.24) prevent upper bound violations of supply for open facilities, and constraints (2.25) require that demand is satisfied. For fixed z, (CFL') reduces to the classical transportation problem and is easily solved.

Akinc and Khumawala [1] have generalized Khumawala's [87] bounding rules to the capacitated case and, additionally, have proposed a hybrid node selection rule. The node selection rule makes use of two parameters, α and β. Specifically, when a node is fathomed, the next node evaluated is selected to be the one with the least lower bound (LLB). This procedure will eventually result in a large number of non-terminal nodes in the enumeration tree. A last-in-first-out (LIFO) method generally leaves few non-terminal nodes since the node selection priority favors any completion of lower level nodes, if indicated, before backtracking to higher level nodes. Akinc and Khumawala's

procedure implements an LLB scheme and continues until the number of non-terminal nodes reaches the level β at which time a switch is made to LIFO to "clean up" some of the non-terminal nodes. When the number of non-terminal nodes reaches the level α, the procedure reverts to LLB.

Akinc and Khumawala [1] proposed the following eight branching rules:

- Integrality of Z
  1. Max Z--select the free warehouse which has the largest z from the set of warehouses having fractional z.
  2. Min Z--select the free warehouse which has the smallest z from the set of warehouses having fractional z.

  where

  $$z_k = 1 - x^*_{k,m+1}/S_k,\ k\epsilon K_2 \text{ where}$$

  $x^*_{k,m+1}$ is the unused capacity of warehouse k in the solution to the bounding problem.

- Penalty Functions
  1. Largest Omega--select the free warehouse having the largest omega. Select z_k: $\Omega_k - F_k = \underset{i\epsilon K_2}{\text{Max}} \{\Omega_i - F_i\}$
  2. Smallest Omega--select the free warehouse having the smallest omega. Select z_k: $\Omega_k - F_k = \underset{i\epsilon K_2}{\text{Min}} \{\Omega_i - F_i\}$

where

$$\Omega_i = \underset{0\le x_{ij}\le D_{ij}}{\text{Max}} \left\{ \sum_j w_{ij}x_{ij} \;\middle|\; \sum_j x_{ij} \le S_i \right\}$$

and

$$w_{ij} = v_j - c_{ij}$$

and

v_j is the dual variable corresponding to the demand constraint for customer j.

w_{ij} can be thought of as the marginal decrease in the transportation cost brought about by making x_{ij} basic.

3. Largest Delta--select the free warehouse having the largest delta. Select z_k: $\Delta_k - F_k = \underset{i \in K_2}{\text{Max}} \{\Delta_i - F_i\}$

4. Smallest Delta--select the free warehouse having the smallest delta. Select z_k: $\Delta_k - F_k = \underset{i \in K_2}{\text{Min}} \{\Delta_i - F_i\}$

where Δ_k is defined to be the difference between the optimal transportation costs with an augmented configuration derived by opening all free warehouses and adding to the configuration a set of dummy warehouses with the capacity constraints relaxed, and the optimal transportation costs of the augmented configuration, but with warehouse k closed.

· Feasibility

1. Largest Capacity--select the free warehouse with the largest capacity. Select z_k: $\tilde{C}_k = \underset{i \in K_2}{\text{Max}} \tilde{C}_i$

2. Smallest Capacity--select the free warehouse with the smallest capacity. Select z_k: $\tilde{C}_k = \underset{i \in K_2}{\text{Min}} \tilde{C}_i$

where

$$\tilde{C}_i = \text{Min} \{S_i, \sum_{j \in J_i} D_j\} \; i \in K_2, \; J_i = \{j \mid w_{ij} > 0\}$$

It was found that the branch and bound algorithm was significantly affected by how soon nodes became capacity feasible.

The bounding rules which performed best were the Largest Omega, Largest Z, Largest Capacity, and Smallest Delta.

Dearing and Newruck [17] have reported the use of an implicit enumeration scheme to solve a bottleneck version of (CFL) in which it is desired to minimize the maximum transportation costs subject to an upper bound on fixed charges. In this variant, specification of the binary variables generates subproblems which are bottleneck transportation problems for which efficient solution procedures are known. See, for example, Garfinkel and Rao [39].

Nauss [111] has recently proposed a branch and bound algorithm which appears to computationally outperform that of Akinc and Khumawala [1]. Nauss invokes "pegging rules" which significantly reduce the number of branching operations by enabling, through the use of tight lower bounds and penalties derived from Lagrangean relaxation, certain facilities to be "pegged" open or closed. The relaxation, a mixed integer program, is solved efficiently by decomposition methods. Specifically, for a fixed configuration, solving a continuous knapsack problem for each facility yields an optimal flow vector. Then, using the solution values, solving a 0-1 knapsack problem provides an optimal configuration (for fixed x) and a solution to the relaxation.

2.6 Generalized Capacitated Facility Location Model

Geoffrion and McBride [59] have applied Lagrangean relaxation to a generalized version of (CFL). This Generalized Capacitated Facility Location formulation, (CFLG) is:

(CFLG) Minimize $\sum_{i\in I} \sum_{j\in J} c_{ij}x_{ij} + \sum_{i\in I} f_i z_i$

Subject to

$$\underline{V}_i z_i \leq \sum_{j\in J} D_j x_{ij} \leq \overline{V}_i z_i \qquad i\in I \qquad (2.28)$$

$$\sum_{j\in J} a_{ij}x_{ij} + \sum_{j\in J} b_{ij}z_{ij} = r_i \qquad i\in I \qquad (2.29)$$

and (2.21)-(2.23).

The generalization consists of permitting lower as well as upper bounds on the volume constraints at each location, enforced by constraints (2.28), and allowing for an arbitrary set of linear constraints (2.29) to be imposed on the x and z variables.

The Geoffrion and McBride [59] results are convincing. A series of test runs were conducted on an IBM 370/158 computer to compare the performance of a branch and bound algorithm with and without Lagrangean relaxation as the bounding device. The bounding device used for comparative purposes was a linear programming relaxation with penalties computed from the Driebeek-Tomlin formula [131]. A typical test problem was characterized by 24 potential facilities, 102 customers, 298 transportation links, and 76 linear side constraints. Computational results for this problem were 243.8 CPU seconds using the conventional Driebeek-Tomlin penalties and 51.8 CPU seconds using Lagrangean relaxation.

2.7 Stochastic Transportation-Location Model

Some limited research has dealt with a single-commodity version of the problem in which demands are random variables. This Stochastic Transportation Location Problem (STLP) can be explicitly represented by the following formulation:

$$\text{(STLP)}\quad \text{Minimize} \sum_{i\in I\cup I'} \sum_{j\in J} c_{ij}x_{ij} + \sum_{j\in J}\left[h_j \int_0^{y_j} (y_j - v)\,\phi_j(v)dv + p_j \int_{y_j}^{\infty} (v - y_j)\,\phi_j(v)dv\right] + \sum_{i\in I} f_i z_i$$

Subject to

$$y_j = \sum_{i\in I} x_{ij} \qquad j\in J \qquad (2.30)$$

$$\sum_{j\in J} x_{ij} \le S_i \qquad i\in I\cup I' \qquad (2.31)$$

$$y_j \ge 0,\ z_i\in\{0,1\} \qquad j\in J,\ i\in I' \qquad (2.32)$$

$$z_i = 1 \qquad i\in I$$

and (2.26)-(2.27)

where

$\phi_j(v)$ is the density function for the demand for customer j.

h_j is the unit holding cost at location j.

p_j is the unit shortage cost at location j.

y_j is the total amount shipped into location j from all sources.

I is the set of all existing supply nodes.

I' is the set of all proposed new supply nodes.

and the remaining notation is as previously defined.

Constraints (2.30) are definitional and (2.31) enforce upper bounds on supply.

Balachandran and Jain [3] have proposed a branch and bound algorithm and LeBlanc [99] has proposed a heuristic for solving (STLP).

2.8 Multicommodity Single-Echelon Distribution System Model

It appears that Geoffrion and Graves [55] were the first to solve the multicommodity location problem as a model which simultaneously deals with location, commodity flows, and customer assignment, while permitting commodities to pass through an intermediate distribution center (DC) enroute from plants to customers. In the Geoffrion-Graves model, sole-sourcing of customers is mandatory, and transportation costs are determined by the total plant-to-customer route and the distance traveled.

The original model proposed in [55] has the following formulation:

(MDS) Minimize $$\sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{p\in P} c_{ijkp}x_{ijkp} + \sum_{j\in J}\left[f_j z_j + v_j \sum_{k\in K}\sum_{p\in P} D_{kp}y_{jk}\right]$$

Subject to

$$\sum_{j\in J}\sum_{k\in K} x_{ijkp} \le S_{ip} \qquad i\in I,\ p\in P \qquad (2.33)$$

$$\sum_{i\in I} x_{ijkp} = D_{kp}y_{jk} \qquad p\in P,\ j\in J,\ k\in K \qquad (2.34)$$

$$\underline{V}_j z_j \le \sum_{k\in K}\sum_{p\in P} D_{kp}y_{jk} \le \overline{V}_j z_j \qquad j\in J \qquad (2.35)$$

$$x_{ijkp} \ge 0 \qquad i\in I,\ j\in J,\ k\in K,\ p\in P \qquad (2.36)$$

$$y_{jk}, z_j \in \{0,1\} \qquad j\in J,\ k\in K \qquad (2.37)$$

where p,i,j,k index commodities, plants, DC's, and customers respectively; v_j is the unit variable cost of throughput for a DC located at

j; $\underline{V}_j$ ($\overline{V}_j$) is the minimum (maximum) allowable throughput for a DC at site j, and y_{jk} is a binary variable which represents the assignment of customer k to a DC at site j. Constraints (2.33) place an upper bound on supply, by commodity, at each source. Constraints (2.34) not only require that demand be satisfied, but that it be satisfied by the DC to which the customer has been assigned. Constraints (2.34) together with (2.35) insure that demand will only be satisfied by open DC's. Constraints (2.35) require that the total DC throughput not exceed the upper and lower bounds for DC's which are open.

Geoffrion, Graves, and Lee [56,57] refined (MDS) to a form more amenable to practical application. In the revised version, sole-sourcing is only imposed on a "bundle" of similar items, not the totality of demand for all items. Upper and lower limits on DC throughput are not strictly enforced and violation is allowed at a penalty cost. Lower as well as upper limits are imposed on plant capacity to enable some control over economies of scale. Throughput is computed as a weighted sum of items shipped through a DC with each commodity having a distinct weight. And, finally, the refinement permits the unit variable cost of throughput to differ by commodity. The Geoffrion-Graves-Lee refinement, (MDSR), has the following formulation:

$$\text{(MDSR)}\quad \text{Minimize} \sum_{i\in I}\sum_{j\in J}\sum_{k\in K}\sum_{p\in P} c_{ijkp}x_{ijkp} + \sum_{j\in J}\Big[f_j z_j + \sum_{k\in K}\sum_{b\in B_k}\sum_{p\in \bar{P}_k(b)} v_{jp}D_{kp}y_{jkb}\Big] + \sum_{j\in J}\big[\underline{P}_j\underline{\nu}_j + \overline{P}_j\overline{\nu}_j\big]$$

Subject to

$$\underline{S}_{ip} \leq \sum_{j\in J} \sum_{k\in K} x_{ijkp} \leq \overline{S}_{ip} \qquad p\in P,\ i\in I \qquad (2.38)$$

$$\sum_{i\in I} x_{ijkp} = D_{kp} y_{jkb} \qquad j\in J,\ b\in B_k,\ p\in P_k(b) \qquad (2.39)$$

$$\sum_{j\in J} y_{jkb} = 1 \qquad k\in K,\ b\in B_k \qquad (2.40)$$

$$\underline{V}_j z_j - \underline{\nu}_j \leq \sum_{k} \sum_{b\in B_k} \sum_{p\in P_k(b)} {}^{\beta P} D_{kp} y_{jkb} \leq \overline{V}_j z_j + \overline{\nu}_j \qquad j\in J \qquad (2.41)$$

$$\underline{\nu}_j,\ \overline{\nu}_j \geq 0 \qquad j\in J \qquad (2.42)$$

and (2.36)-(2.37)

where

B_k denotes the set of bundle indices for customer k.

$P_k(b)$ denotes the set of commodity indices corresponding to bundle b for customer k.

β^P = burden factor for commodity p used in calculating DC throughput.

$\underline{P}_j, \overline{P}_j$ = penalty rates for violating lower (upper) throughput limits for DC j.

$\underline{S}_{ip}(\overline{S}_{ip})$ = lower (upper) limits on plant capacity for plant i.

$\underline{\nu}_j(\nu_j)$ = amount of underflow (overflow) at DC j.

The Geoffrion-Graves-Lee model has benefitted from many man-years of effort and numerous applications, and has evolved into a user-oriented complete management support system known as ODS [57]. ODS is a fully integrated system tailored to the specific needs of managerial decision-making, and consisting of four major components:

1. A modeling framework--the conceptual context and menu of options and solvers with which models can be formulated to represent problem specific operational systems.
2. A data file structure--a set of data specifications for the model including how the data must be organized and the required form of input.
3. An optimizer--the software package that implements the model.
4. Interface facilities--the vital communication link to the user. The primary function of this component is communication both incoming and outgoing; incoming in the sense of creating and maintaining necessary data files, outgoing in the sense of generating meaningful management reports as computer results are obtained.

The modeling framework is of principal interest in the present work. The solution technique employed is based on the decomposition theory developed by Benders [7]. This procedure, which was first applied to distribution/location problems by Balinski [4], and which is discussed in detail in Chapter IV, is based on separating a difficult problem into two simpler problems. Problem (MDSR) is a large-scale integer programming problem and, for most practical applications, computationally intractable using conventional branch and bound techniques unless some very efficient bounding devices and branching criteria can be found. Benders decomposition reduces the level of difficulty factor in the following way.

Referring to (MDSR), an equivalent formulation is

$$\text{Minimize} \left\{ \sum_{j\epsilon J} [f_j z_j + \sum_{k\epsilon K} \sum_{b\epsilon B_k} \sum_{p\epsilon P_k(b)} v_{jp} D_{kp} y_{jkb}] \right.$$

$$+ \sum_{j\epsilon J} [\underline{P}_j \underline{\nu}_j + \overline{P}_j \overline{\nu}_j]$$

$$\left. + \text{Min} \left\{ \sum_{i\epsilon I} \sum_{j\epsilon J} \sum_{k\epsilon K} \sum_{p\epsilon P} c_{ijkp} x_{ijkp} \right\} \right\} \qquad (2.43)$$

Subject to (2.38)-(2.41).

If $z, y, \underline{\nu}$, and $\overline{\nu}$ are held temporarily fixed, (2.43) together with (2.36) and (2.38)-(2.39) defines a classical multicommodity transportation problem which decomposes on commodity. If a solution x to this transportation problem is held fixed temporarily and the vectors $z, y, \underline{\nu}$, and $\overline{\nu}$ are permitted to vary, (2.43) together with (2.37) and (2.39)-(2.42) defines a mixed integer programming problem.

The Benders algorithm is based on the convergence of upper and lower bounds obtained from the oscillatory solutions to the two problems. With the binary variables held fixed (that is, a fixed configuration) the commodity-independent transportation subproblems are solved for optimal transportation costs and flows. These subproblems constitute restrictions on (MDSR) since not all of the variables are free to vary. Consequently any subproblem solution is an upper bound on (MDSR). The iterative solution of these subproblems with different configurations generates a sequence (non-monotonic) of such upper bounds. Each time a subproblem is solved, the optimal solution is used to solve the master problem for a new configuration. Specifically the master problem is (2.43) with "Min $\sum_{ijp} c_{ijp} x_{ijp}$" replaced by "Max $\sigma(h)$" and subject to (2.37) and (2.39)-

(2.41), where $\sigma(h)$ is the value of the optimal objective function of the h^{th} transportation subproblem solved. The sense of the optimization is "maximize" instead of "minimize" since it is actually the transportation duals which are solved rather than the primals. The reason for this is that the solution space of the transportation dual is configuration-independent. Such is not the case with the primal. The master problem is a relaxation of (MDSR) and each additional transportation subproblem solved contributes a new constraint of the form $\sigma \geq \sigma(h)$ to the master. These constraints are called Benders cuts and because each new cut reduces the size of the solution space of the master problem, successive solutions constitute a monotonic increasing sequence of lower bounds on (MDSR). The master problem is solved for a new configuration and the procedure repeats. Termination occurs when the upper and lower bounds converge to an ε-gap. Computational experience with the Benders decomposition algorithm is discussed in the next section.

2.9 Applications

The literature abounds with examples of successful model implementations of problem-specific variants of this general class of location problems. Highlighting some selected scenarios is a fitting way to close this chapter.

Geoffrion [47] reports the results of a large-scale distribution warehouse location analysis for Hunt-Wesson Foods, Inc., a firm which produces several hundred distinct food products at fourteen plants, and distributes these products to customers nationally through a network of twelve intermediate distribution centers. The solution

technique used for the analysis was Benders decomposition as described above. Computational results on an IBM 360/91 computer ranged from just over 16 CPU seconds in the case of a fixed configuration of 16 distribution centers, and solving for 249 assignment variables only; to 191 CPU seconds in the case of 30 distribution center candidates, and 513 binary variables. In the former case only three master problems had to be solved; in the latter case, five solutions had to be generated. Cost savings were estimated to be in the low millions.

Kochman and McCallum [92] demonstrate how the formulation of a model can impact solution efficiency. These authors solved a planning problem concerned with the optimum placement of communication cables, and the routing of individual circuits between demand points (both satellite and cable) such that the total discounted cost over a T-period horizon is minimized. This is a multiperiod capacitated distribution/location problem with some complicating side constraints which specify, for reliability purposes, a minimum network diversity. Two formulations emerge: facility diversity and routing diversity. The routing diversity problem is significantly smaller than the facility diversity problem and can be more efficiently solved. In the application, both formulations contained 75 binary variables, but the facility diversity formulation contained 5400 continuous variables compared to 600 for the route diversity problem. Using MPSX and a branch and bound scheme, the route diversity formulation required 20-30 CPU minutes on an IBM 370/168. In comparison, the facility diversity model had a 5-10% gap between the best lower bound and the incumbent after 4-6 hours of

computing. The method actually used for solving the facility diversity model was based on relaxing some of the coupling constraints and using the efficient code developed by Chen and Saigal [12]. This procedure reduced the solution times for the facility diversity problem to about three minutes, and the first solutions produced were never more than 12% from the optimum.

Van Roy and Gelders [134] report impressive results in the design of a large-scale single commodity single-echelon problem for a company which distributes bottled product. The problem solved is a capacitated plant location problem with some special linear side constraints. The approach used was a variant of Erlenkotter's dual ascent method [21] imbedded in a branch and bound scheme. All capacity constraints were relaxed and the dual ascent method was used to solve the relaxations. This procedure proved to be extremely fast. The problem consisted of 40 customers, 10 distribution centers, and 17 side constraints. A total of 30 parameter adjustment cases of the problem were run, and the average solution time was .3 CPU seconds on an IBM 3033. Natural integer solutions were obtained in 40% of the cases. Of the 60% remaining, at most three nodes had to be explored on the branch and bound tree.

Some other applications worthy of note are Nambier, Gelders, and Van Wassenhove [109] who used a heuristic to locate processing plants and collection stations, and solved the associated vehicle routing problem, for latex product in Malaysia; Jacobson and Madsen [78] who located transfer points and solved simultaneous vehicle routing

problems for a large newspaper firm; and Markland and Newett [101] who solved a large-scale multiperiod distribution planning model for a soybean processing firm.

CHAPTER III

THE MULTI-ECHELON MULTICOMMODITY DISTRIBUTION SYSTEM MODEL

3.1 Problem Motivation

The literature is conspicuously lean in contributions of models designed to solve large-scale capacitated distribution/location problems with multiple sources and sinks, multiple commodities, and throughput-capacitated transshipment nodes. ODS, the management support system of Geoffrion, Graves, and Lee [57] would appear to be the paradigm, since (1) its empirical performance has proven to be robust over a wide range of applications, and (2) no models have been proposed in the literature to challenge it. In the present work, therefore, the ODS optimizer model was the pattern for study with a view toward finding beneficial ways in which modeling capability for this important class of problems could be expanded. The result has been an extended capability in one very germane aspect (the inclusion of multiple echelons) and the foundations have been planted for generalizing the model in another (the incorporation of nonlinear costs).

Geoffrion, who for years has been an active participant in industrial strategic planning observed the need for a modeling capability which will handle multiple echelons and nonlinear throughput costs [54], neither of which can be effectively dealt with using existing methods.

Multiple echelons. ODS is limited to distribution system designs in which all products pass through a single DC enroute from plant to customer. For some industries it may be possible to realize considerable efficiencies by permitting multiple echelons of distribution between plants and customers. The present work allows complete generality in the number of intermediate stocking points between any plant and customer pair. For example, the same product leaving the same plant may reach customer A via three echelons of DC's, customer B via a single DC, and customer C via direct plant-to-customer transfer.

Nonlinear throughput costs. The mathematical model of physical distribution underlying ODS assumes that the variable costs of operating a DC depend linearly on throughput volume. The present work investigated the incorporation of nonlinear cost functions with particular attention to concave or S-curves which typify many inventory systems. Whilst this aspect has been addressed and solution strategies considered, a major drawback which discounts the prospects for global implementations in the immediate future is the unavailability of realistic distribution cost information. It appears that firms simply do not know enough about the cost response surfaces as a function of major perturbations in volume. Due to the practical time limitations available for the research reported herein, cost modeling has necessarily had to be left as a topic for future interest.

The practical impetus, coupled with the compelling economic incentives elaborated in the Introduction, has fueled this research from the start and will continue to stimulate those who follow.

3.2 Problem Definition

Consider the following hierarchy of problems:

- How many distribution centers (DC's) and of what sizes should exist to serve as intermediaries between production facilities and customers?
- Given an answer to the first question, where should these DC's be located?
- Given answers to the first and second questions, that is, given the system configuration, how should commodities be routed from plants to DC's and then on to customers?

Now, let this hierarchy of questions also permit the existence of a multi-echelon network structure. The term "multi-echelon" is used to describe any system in which transactions are allowed to occur between any two DC's. It will be assumed that, whilst the first question has yet to be answered, there does exist a finite list of DC candidates for which the costs of operation and establishment are known. Additionally, upper and lower throughput limits for each DC candidate have been set--that is, practical maximum and minimum throughput levels for each facility are known. These throughput bounds represent a shared volume for all commodities. Next, let there exist upper and lower bounds for each commodity at each plant (the lower bound may be zero). And, finally, assume that demands are known with certainty for each commodity, and that customers can be grouped into a set of finite identifiable zones.

Costs may be assumed to be linear throughout, or the model can be expanded to include general cost structures. The principal thrust of the work documented herein has been the development of computational machinery to handle the linear case, machinery which could potentially be applied to nonlinear problems. A guiding principle throughout the research has been that algorithmic design must not only lead to theoretical solutions but must also be computationally efficient.

3.3 Model Description

Consider a directed graph G = (N,A). G is characterized by multiple sources (plants), multiple sinks (customer zones), and a number of transshipment nodes corresponding to a set of proposed sites for the location of distribution centers (DC's). The set of proposed sites includes existing sites, if applicable. Multiple arcs, each of which represents a distinct commodity, or commodity grouping, connect each node pair belonging to N.

Supply of each commodity at each source is upper and lower bounded. Upper and lower bounds also apply to the total throughput, summed over all commodities, at each transshipment node. And, finally, demand for each commodity at each sink must be met.

In order to enforce the throughput bounds at the transshipment nodes, a simple trick due to Ford and Fulkerson [35, pages 23-25] is used. This device enlarges the original network by splitting each DC node into a node pair (inbound and outbound). The arc between the inbound node and the outbound node is assigned the node capacity and

cost. Hence, capacitated nodes are transformed into capacitated arcs. The following notation is relevant:

Notation: P denotes the set of commodities.

$F \subset N$ denotes the set of sources (plants).

$C \subset N$ denotes the set of sinks (customers)

and $F \cap C = \emptyset$.

$I = N - F \cup C$ denotes the set of candidate transshipment nodes.

Let, I^+ denote the set of inbound nodes.

I^- denote the set of outbound nodes.

$D \subset A$ denotes the set of arcs(k): $k \in I^+, k+1 \in I^{-*,\dagger}$

Cost coefficients:

c_{ijp} = unit cost of sending flow along arc(i,j) for commodity p. $(i,j) \in A-D$.

f_k = fixed cost of establishing DC k on the network.

$v_k(\cdot)$ = variable cost function of total DC throughput, represented by $(\cdot)$, for DC k : $k \in D$.

---

*For notational convenience, it will be assumed that the node numbering regimen assigns a number to the outbound node i^- which is precisely equal to the number assigned to the inbound node i^+ plus one. [In this thesis, the norm will be to refer to the special DC arcs by the single subscript k, which is taken to imply arc(k,k+1).] When these arcs are referred to as part of a larger set or the entire network, it will be clearer to revert to the (i,j) notation.

†Note that this definition restricts the set to those special arcs (which in general will be a very small subset of A) connecting the node-pair corresponding to the inbound and outbound nodes for each individual DC and does not include those arcs which may exist between the outbound node of one DC and the inbound node of another--that is arcs (k+1,j): $k+1 \in I^-, k, j \in I^+$ and $k \neq j$.

Variables:

x_{ijp} = continuous flow variables representing the quantity of commodity p shipped from node i to node j.

x_{kp} = continuous variables representing the quantity of commodity p shipped through DC k.

z_k = "open-shut" binary variable which equals unity if DC k has been established, zero otherwise.

Right-hand-sides:

$\overline{S}_{ip}$ = upper bound on supply of commodity p at plant i. $i \epsilon F$.

$\underline{S}_{ip}$ = lower bound on supply of commodity p at plant i. $i \epsilon F$.

D_{ip} = demand of commodity p by customer i. $i \epsilon C$.

$\overline{V}_k$ = upper bound on total flow (all commodities) for DC k.

$\underline{V}_k$ = lower bound on total flow for DC k.

3.4 Model Formulation

$$\text{(MDM)} \quad \text{Minimize} \sum_{p \epsilon P} \sum_{(i,j) \epsilon A-D} c_{ijp} x_{ijp} + \sum_{k \epsilon D} [f_k z_k + v_k(\phi_k)]$$

Subject to,

$$\phi_k = \sum_{p \epsilon P} x_{kp} \qquad k \epsilon D \qquad (3.1)$$

$$\underline{S}_{ip} \leq \sum_{j \epsilon N} x_{ijp} \leq \overline{S}_{ip} \qquad i \epsilon F,\ p \epsilon P \qquad (3.2)$$

$$\sum_{j \epsilon N} x_{jip} = D_{ip} \qquad i \epsilon C,\ p \epsilon P \qquad (3.3)$$

$$\underline{V}_k z_k \leq \phi_k \leq \overline{V}_k z_k \qquad k\epsilon D \qquad (3.4)$$

$$x_{ijp} \geq 0 \qquad (i,j)\epsilon A\text{-}D,\ p\epsilon P \qquad (3.5)$$

$$x_{kp} \geq 0 \qquad k\epsilon D,\ p\epsilon P \qquad (3.6)$$

$$z_k \epsilon \{0,1\} \qquad k\epsilon D \qquad (3.7)$$

Lower bounds on plant supply can be eliminated by a standard trick, which involves introducing upper bounded slack variables. These slack variables require that a dummy customer q be added to the set C. Therefore, the variable $x_{iqp}: i\epsilon F$ denotes slack plant capacity for commodity p for plant i. Further the set $\{(i,q): i\epsilon N\text{-}F\}$ is empty.

Using these transformations, (MDM) can be equivalently written,

$$(MDM_a) \quad \text{Minimize} \quad \sum_{p\epsilon P} \sum_{(i,j)\epsilon A\text{-}D} c_{ijp}\, x_{ijp} + \sum_{k\epsilon D} [f_k z_k + v_k(\phi_k)]$$

Subject to

$$\sum_{j\epsilon N} x_{ijp} = \overline{S}_{ip} \qquad i\epsilon F,\ p\epsilon P \qquad (3.8)$$

$$x_{ipq} \leq \overline{S}_{ip} - \underline{S}_{ip} \qquad i\epsilon F,\ p\epsilon P \qquad (3.9)$$

and (3.1), (3.3), (3.4), (3.5), (3.6), and (3.7).

The appearance of (MDM_a) can be simplified by combining constraints (3.3) and (3.8), and generalizing (3.9). It is noted at this point that constraints (3.1) are notational only.

An equivalent form of (MDM_a) is,

$$(MDM_b) \quad \text{Minimize} \quad \sum_{p\epsilon P} \sum_{(i,j)\epsilon A\text{-}D} c_{ijp}\, x_{ijp} + \sum_{k\epsilon D} [f_k z_k + v_k(\phi_k)]$$

Subject to

$$\sum_{j\epsilon N} x_{ijp} - \sum_{j\epsilon N} x_{jip} = b_{ip} \qquad i\epsilon N,\ p\epsilon P \qquad (3.10)$$

$$0 \leq x_{ijp} \leq u_{ijp} \qquad (i,j)\epsilon A \qquad (3.11)$$

and (3.1), (3.4), and (3.7)

where

$$b_{ip} = \begin{cases} \overline{S}_{ip} & i\epsilon F \\ -D_{ip} & i\epsilon C \\ 0 & \text{otherwise} \end{cases}$$

$$u_{ijp} = \begin{cases} \overline{S}_{ip} - \underline{S}_{ip} & (i,j): i\epsilon F,\ j = q,\ p\epsilon P \\ \alpha_{kp}\,\overline{V}_k & k\epsilon D,\ p\epsilon P,\ \alpha_{kp}\epsilon(0,1]^* \\ \infty & \text{otherwise} \end{cases}$$

It may be noted that the model as proposed does not enforce sole-sourcing. In many applications it is in fact undesirable to rigidly enforce this concept, a belief expressed by Fisher [31], and given partial recognition in the Geoffrion-Graves-Lee model [56], which replaces an absolute sole sourcing policy with the more relaxed bundling structure.

(MDM_b) is a mixed nonlinear integer programming model and, it can also be observed that this model possesses a very special structure.

---

*The introduction of α_{kp} here is to provide some upper bounds on the DC x-variables, thereby accelerating convergence of the solution algorithm. A resource directive scheme will seek to solve for the optimum α's which sum to unity over each DC. Any collection of starting α's could be used, the sum of which are greater than one. In the MEDOS procedure, elaborated later in this thesis, all α's are initialized at one.

All coefficients on the x variables are either -1, 0, or +1. The coefficient matrix is a standard node-arc incident matrix with some additional rows for the DC throughput constraints. This observation implies the possible use of some streamlined network procedures for solving the problem. Not insignificantly, in most strategic planning applications, these problems tend to be extremely large. Conventional enumerative (tree search) approaches can produce slow convergence unless procedures can be devised to improve the quality of lower bounds and thereby accelerate fathoming of unsolved subproblems (that is, restricting the area of the search).

The present research has examined (MDM_b) in the context of relevant work reported in the literature for solving large-scale mathematical programming problems with particular interest in preserving and exploiting the special network structure of this problem.

If $(\cdot)$ in (MDM_b) is defined to be a multiplication operator, rather than functional notation, and v_k the applicable unit variable throughput cost (or v_{kp} if commodity-distinct costing is used), the resulting model is a variant of (MDS) in which a multi-echelon structure is incorporated and sole-sourcing is deleted. For the linear case, let $c_{ijp} = v_k$ (or v_{kp}) $k \in D$, $(i,j) : i \in I$. Then, (MDM_b) simplifies to the following formulation:

$$\text{(MDML)} \quad \text{Minimize} \sum_{p \in P} \sum_{(i,j) \in A} c_{ijp} x_{ijp} + \sum_{k \in D} f_k z_k$$

Subject to

$$\underline{V}_k z_k \leq \sum_{p \in P} x_{kp} \leq \overline{V}_k z_k \qquad k \in D \qquad (3.12)$$

and (3.7), (3.10), and (3.11).

(MDML) is a Fixed Charge Multi-commodity Flow Problem and this problem has been the focal point of the work documented herein.

CHAPTER IV

THE FIXED CHARGE MULTICOMMODITY FLOW PROBLEM: POSSIBLE SOLUTION STRATEGIES

Two alternative solution strategies for solving the Fixed Charge Multicommodity Flow problem have been investigated: branch and bound and Benders decomposition. Throughout large-scale integer programming literature branch and bound has gained universal acceptance. Refer Garfinkel [37]. Benders decomposition, though not as widely used, has been firmly established as a useful tool for solving one class of large-scale distribution problems across a wide range of industrial applications ([55, 56, 57]). Magnanti and Wong [100] have noted that the Benders algorithm has been extremely slow in converging in some other applications. The present work uses branch and bound, although some of the features of the algorithm proposed were inspired by the price directive decomposition principles engendered by the Benders approach. Section 4.1 of this chapter provides background for the techniques actually used in designing the algorithm described in Chapter V. Section 4.2 describes the theoretical basis of the Benders Decomposition approach. Even though this approach was not used in the present algorithm, the Benders algorithm should be seen as a serious alternative contender for solving the Fixed Charge Multicommodity Flow Problem. A valuable topic for further study would be the investigation of possible efficient methods for solving the Benders Master Problems (relaxations), specialized to this class of distribution/location

problems. Section 4.3 synopsizes the recipe of ingredients drawn from the two solution strategies discussed in Sections 4.1 and 4.2 and utilized in the current work.

4.1 Branch and Bound

Branch and bound is a term used to describe a method for solving mixed integer programming problems. A rooted tree structure is used to represent an enumeration of subproblems, each subproblem corresponding to a constraint imposed on the original problem. If all integer variables are binary (0-1), the enumeration tree is finite. If there are m of these variables, there are 2^m possible subproblems, or nodes, on the enumeration tree ($2^m - 1$ if one is not concerned with the instance in which all variables are equal to zero). Branch and bound specialized to the binary variable case is called implicit enumeration. Each edge, or branch, of the enumeration tree defines a value for one of the binary variables. Each vertex, or node, of the tree represents a subproblem in which the variables, as defined by the unique collection of branches between the node and the root, are fixed. Any variable not so defined is said to be free; and the fixing of a free variable is said to be a completion of that node.

Even though it is theoretically possible to extract an optimal solution through complete enumeration, since the number of subproblems is finite, the total number of such problems become prohibitively large for even a moderate size problem. It therefore is imperative that methods for legitimately skipping large numbers of subproblems be investigated. Any subproblems so skipped are said to have been

implicitly enumerated, and the process of eliminating these subproblems from further consideration is called fathoming. Fathoming can occur any time that it is clear that no completion of the present node can produce a solution which improves on the best known solution up to that point. When fathoming is not possible a new completion is constructed (that is, a new branch on the tree is drawn). The process of generating node completions is called branching, or separation. Good bounding techniques are critical to fathoming efficiency, which in turn is important to computational efficiency (that is, keeping the tree search small).

In designing a branch and bound scheme, four decisions are relevant:

1. how to obtain quality upper bounds (that is, how to generate good feasible solutions).
2. how to obtain quality lower bounds (that is, selecting a good relaxation strategy).
3. how to establish effective fathoming rules.
4. after a subproblem is solved, what should be done next (that is, how to select the best node to examine next when fathoming or branching).

The second issue will be dealt with first. Lagrangean relaxation is a technique which has become firmly established as a method for achieving tight lower bounds; bounds which can often be near-feasible solutions. See for example Fisher [30].

4.1.1 Lagrangean Relaxation

Lagrangean relaxation is a procedure in which "complicating constraints" are removed from the constraint set (i.e., the feasible region is enlarged) and those constraints are placed in the objective function with cost efficients which represent per unit penalties charged for noncompliance of the constraints. From Linear Programming theory it is known that, if an optimal solution exists to the original fully-constrained problem, there also exists a set of optimal penalty coefficients. If these coefficients were known in advance, solving the relaxed form of the problem would yield equivalent results to solving the unrelaxed version (i.e., optimality upon termination). To elaborate, what was originally a pure minimization problem has been transformed into a maximization. For any fixed set of penalties, the problem reverts to the pure minimization form, only with an adjusted objective function. If this problem is solved and constraint violations occur, the penalties will effect an increase in objection function value. As the penalties are increased, a minimization algorithm is less and less likely to permit such violations. Optimality is reached at precisely that point where penalties are just "large enough" to enforce constraint obedience. The Lagrangean problem therefore seeks to find the maximum of the infimums of the objective function values obtained over the range of all possible penalties. Since these optimal penalties are unfortunately never known they must either be estimated through some external pricing mechanism, or are in themselves a by-product of the solution procedure. These coefficients have a

variety of names (e.g., Lagrange multipliers, dual multipliers, simplex multipliers, dual variables, penalty coefficients). The act of relaxing a constraint and bringing it up into the objective function (forming a "Lagrangean function") is called "dualizing" the constraint. Any constraint can be dualized, but there must be a clear benefit in doing so (e.g., the relaxed problem is easier to solve, computer storage or computational requirements are reduced, the relaxed form fits some model for which standard algorithms and software are available).

Referring to (MDML) constraints (2) are complicating in the sense that these constraints couple commodities. Dualizing these constraints, the Lagrangean relaxation is:

$$(\text{MDML}_R) \quad \text{Minimize} \quad \sum_{p\in P} \sum_{(i,j)\in A} c_{ijp}\, x_{ijp} + \sum_{k\in D} f_k z_k$$

$$+ \sum_{k\in D} \left[\pi_k\left(\underline{V}_k z_k - \sum_{p\in P} x_{kp}\right) - \lambda_k\left(\overline{V}_k z_k - \sum_{p\in P} x_{kp}\right)\right]$$

Subject to

$$\sum_{j\in N} x_{ijp} - \sum_{j\in N} x_{jip} = b_{ip} \qquad i\in N,\ p\in P \qquad (4.1)$$

$$0 \le x_{ijp} \le u_{ijp} \qquad (i,j)\in A,\ p\in P \qquad (4.2)$$

$$z_k \in \{0,1\} \qquad k\in D \qquad (4.3)$$

which can equivalently be written

$$\text{Minimize} \quad \sum_{p\in P} \sum_{(i,j)\in A} (c_{ijp} + \lambda_{ij} - \pi_{ij})\, x_{ijp}$$

$$+ \sum_{k\in D} (f_k + \pi_k \underline{V}_k - \lambda_k \overline{V}_k) z_k$$

Subject to (4.1), (4.2), and (4.3)

where,

$$\lambda_{ij},\ \pi_{ij} \begin{cases} = 0 & (i,j)\in A\text{-}D \\ \geq 0 & (i,j)\in D \end{cases}$$

$$\lambda_k,\ \pi_k \quad \geq 0 \quad k\in D$$

For fixed z's, λ's, and π's, ($MDML_R$) reduces to a problem of the form,

$$(MDML_R(z,\lambda,\pi)) \quad \text{Minimize} \sum_{p\in P} \sum_{(i,j)\in A} \tilde{c}_{ijp}\, x_{ijp} + \sum_{k\in D} (f_k + \pi_k \underline{V}_k - \lambda_k \overline{V}_k) z_k$$

Subject to (4.1) and (4.2)

where

$$\tilde{c}_{ijp} = c_{ijp} + \lambda_{ij} - \pi_{ij} \qquad (i,j)\in A,\ p\in P$$

$(f_k + \pi_k \underline{V}_k - \lambda_k \overline{V}_k) z_k$ is a constant $k\in D$; λ, π, z fixed

Clearly, $MDML_R(z,\lambda,\pi)$ separates on commodity, so solving this problem is equivalent to solving independent single commodity capacitated transshipment problems of the form,

$(CTP)_p$ Minimize $\sum_{(i,j)\in A} \tilde{c}_{ijp}\, x_{ijp}$

Subject to

$$\sum_{j\in N} x_{ijp} - \sum_{j\in N} x_{jip} = b_{ip} \qquad i\in N \qquad (4.4)$$

$$0 \leq x_{ijp} \leq u_{ijp} \qquad (i,j)\in A \qquad (4.5)$$

The problem $(CTP)_p$ has a familiar and convenient form. It is a capacitated transshipment problem and streamlined solution procedures exist for its solution [10].

The solutions of ($MDML_R$) are lower bounds on the optimum solution to (MDML), since the feasible region of ($MDML_R$) constitutes an enlargement of the feasible region of (MDML). If the dualized constraints are equalities, a feasible solution to ($MDML_R$) which happens to also be feasible in (MDML), would be optimal in (MDML). Fisher [30] discusses this point and how often a near-feasible solution can be made feasible with a little "judicious tinkering." In general, the dualized constraints in ($MDML_R$) will be inequalities, a condition rendering mere feasibility as a sufficient test of optimality inadequate.

However, near-feasible solutions produce tight lower bounds and are especially significant if ($MDML_R$) is used as the bounding problem in a branch and bound algorithm. Sharp lower bounds are dependent on the quality of the λ's and π's selected. Clearly, the optimal λ's and π's will solve,

$$(DR) \quad \underset{\lambda,\pi \geq 0}{\text{Maximize}} \left\{ \text{Min} \left[\sum_{p \in P} \sum_{(i,j) \in A} \tilde{c}_{ijp} x_{ijp} + \sum_{k \in D} (f_k + \pi_k \underline{V}_k - \lambda_k \overline{V}_k) z_k \right] \right\}$$

Subject to (4.1), (4.2), and (4.3).

(DR) suggests that for any fixed λ and π vectors, the inner minimization will generate an optimal z-vector by the following rule:

$$\text{set } z_k = \begin{cases} 1 & k \in D : \overline{V}_k \lambda_k \geq f_k \text{ or } \pi_k > 0 \\ 0 & \text{when } \lambda_k > 0, \pi_k = 0 : \overline{V}_k \lambda_k < f_k \text{ or } \lambda_k = 0,\ \pi_k > 0 \end{cases}$$

This will not generally be the case since the λ's and π's appear as coefficients for the x-variables. However, the z-rule offers

interesting possibilities for unconventional node selection in branch and bound fathoming. When fathoming occurs, a branch and bound scheme backtracks in the direction of the root node to a new subproblem. Many conventional methods employ a last-in-first-out (LIFO) rule which requires backtracking to the node from which the last completion was constructed. The z-rule could be a useful non-conventional node selection tool to direct a tree search to the node represented by the following:

$$\text{Set } z_k = \begin{cases} 1 & k \in D : k \text{ is fixed, } \lambda_k > 0,\ \pi_k = 0, \text{ and } \overline{V}_k \lambda_k \geq f_k \\ 0 & k \in D : k \text{ is fixed, } \lambda_k > 0,\ \pi_k = 0, \text{ and } \overline{V}_k \lambda_k < f_k \\ & \text{or, } \pi_k > 0 \end{cases}$$

Record-keeping problems could arise using unconventional methods. Certainly, the simplicity of LIFO [40] would be lost. The present work does not employ unconventional node selection rules, but such is seen as a possible worthwhile extension which should be thoroughly investigated.

Fisher [31] discusses three methods for obtaining a good sequence of Lagrange multipliers: (1) subgradient methods, (2) the Boxstep Method, and (3) multiplier adjustment methods.

The use of multiplier adjustment methods has not been widely reported. Where such techniques have been employed the use appears to have been limited to small and specialized applications. Under the proper conditions, such methods tend to perform well, notably the dual adjustment technique of Erlenkotter [21].

Subgradient methods and the Boxstep procedure have a more universal applicability.

4.1.1.1 Subgradient methods. A subgradient of a convex function is a linear support of that function relative to a point on the surface of the function. If the function is differentiable at this point, the subgradient is unique and is equal to the gradient. Figure 4.1 illustrates this in E^2 space.

At nondifferentiable points the subgradient is not unique, as illustrated by Figure 4.2.

An explicit definition follows. Additional details may be found in the very excellent paper by Held, Wolfe, and Crowder [72].

Let f be some concave function defined on the convex set S, a proper subset of E^n. The n-vector u is a subgradient of f at the point $x \in S$ if

$$f(y) - f(x) \leq u \cdot (y - x) \text{ for all } y \in S.$$

The sense of the inequality is reversed if f is convex. The following proposition shows that a subgradient of $(MDML_R(z,\lambda,\pi))$ at any point $\psi = (\lambda, -\pi)$ for which x(h) solves $(MDML_R(z,\lambda,\pi))$ is a vector of the throughput bound violations relative to x(h).

Proposition: A subgradient of $(MDML_R(z,\lambda,\pi))$ at any point $\psi = (\lambda,-\pi)$ for which x(h) solves $(MDML_R(z,\lambda,\pi))$ is the vector $u = (\underline{T},\overline{T})$ where the k^{th} components of $\underline{T}$ and $\overline{T}$ are respectively defined by:

$$\underline{T}_k = \sum_{p \in P} x_{kp}(h) - \underline{V}_k z_k$$

$$\overline{T}_k - \sum_{p \in P} x_{kp}(h) - \overline{V}_k z_k$$

Proof: Let $g(\psi)$ be the optimal value of $(MDML_R)$ for fixed ψ. Select some point $\overline{\psi}$. Then

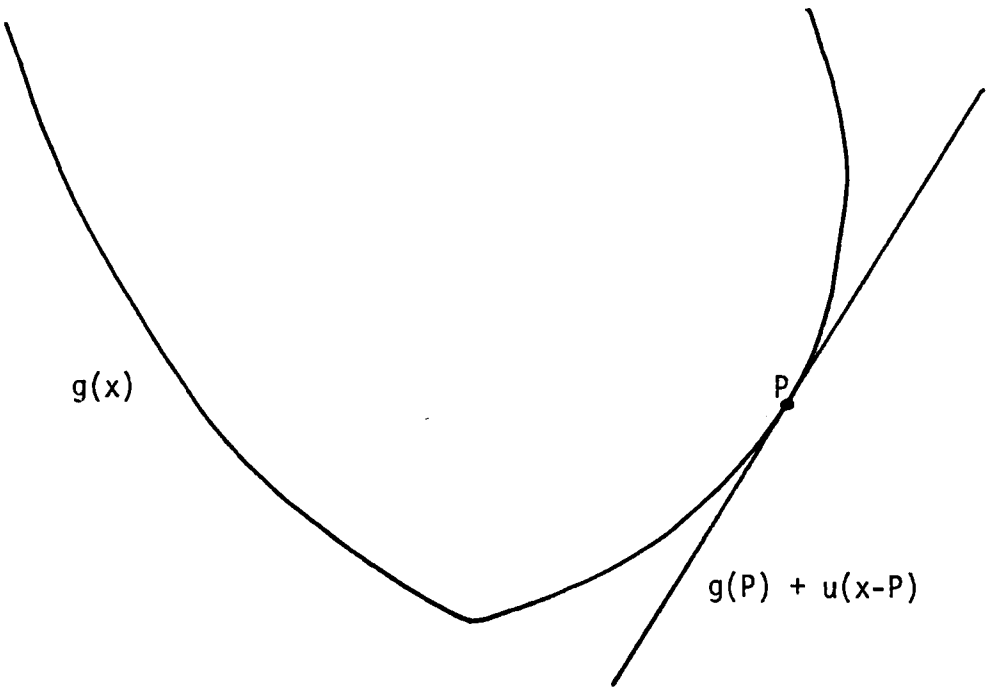

Figure 4.1. Illustration of Unique Subgradient u at Differentiable Point P.

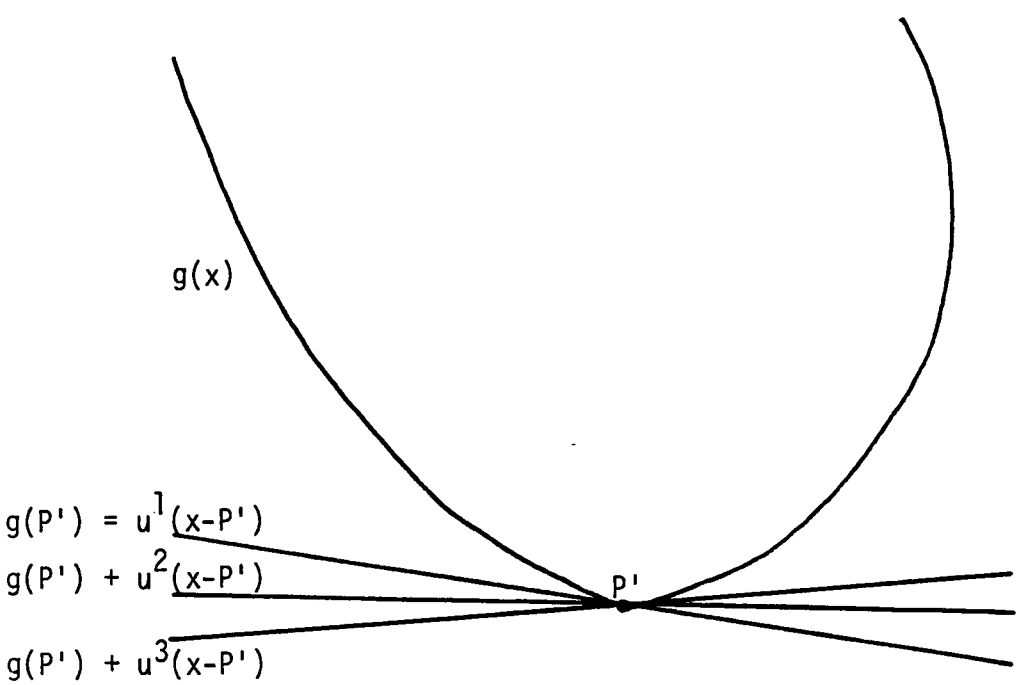

Figure 4.2. Illustration of Numerous Subgradients at Non-Differentiable Point P'.

$$g(\psi) - g(\bar{\psi}) = \sum_{p\epsilon P} c_p x_p + fz + u\cdot\psi - \sum_{p\epsilon P} c_p \bar{x}_p - f_z$$

$$- \bar{u}\cdot\bar{\psi} \leq \sum_{p\epsilon P} c_p \bar{x}_p + fz + \bar{u}\cdot\psi - \sum_{p\epsilon P} c_p \bar{x}_p$$

$$- fz - \bar{u}\cdot\bar{\psi} = \bar{u}\cdot(\psi-\bar{\psi})*$$

The subgradient can be used to establish a direction of improvement in λ and π. This is analogous to the way in which gradients are used in steepest descent methods in the case of differentiable functions.

A sequence of improving λ's and π's can be generated using the following iterative rules:

$$\lambda_k^{h+1} \leftarrow \text{Max} \quad \{0,\ \lambda_k^h + \Theta^h \left(\sum_{p\epsilon P} x_{kp}(h) - \bar{V}_k z_k \right)\}$$

$$\pi_k^{h+1} \leftarrow \text{Max} \quad \{0,\ \pi_k^h - \Theta^h \left(\sum_{p\epsilon P} x_{kp}(h) - \underline{V}_k z_k \right)\}$$

where h indexes the iteration and Θ^h is a positive scalar stepsize constant defined as follows [72]:

$$\Theta^h = \frac{\gamma_h \left(UB - \upsilon(MDML_R)\right)}{\|u\|^2}$$

where $\gamma_h \in (0,2]$, $\upsilon(MDML_R)$ is the incumbent value of the objective function of $(MDML_R)$, UB is an upper bound on (MDML), u is a subgradient, and $\|\cdot\|$ denotes the Euclidean norm.

Apparently any sequence of γ_h's will do provided the following conditions are satisfied:

---

*Throughout the thesis, unless otherwise noted, cost vectors are assumed to be row vectors and all other vectors column vectors. This convention is adopted for typing convenience. For example, the vector multiplication $c^T x$ can be typed as cx.

$$\gamma_h > 0 \qquad \text{all } h$$

$$\sum_h \gamma_h = \infty$$

$$\lim_{h\to\infty} \gamma_h = 0$$

Kennington and Helgason [84] suggest the following rule for generating a sequence of γ's:

$\gamma_h = \alpha$ for 2^ν iterations, after which

$\alpha \leftarrow \alpha/2$ and $\nu \leftarrow \nu - 1$

and initial values for ν and α are 6 and 2 respectively.

4.1.1.2 Boxstep method. It was previously noted that the best vectors λ and π are the ones which solve problem (DR). For a temporarily fixed z vector (DR) reduces to,

$$\text{(DRS)} \quad \underset{\lambda,\pi\geq 0}{\text{Max}} \left\{ \text{Min} \left[\sum_{p\in P} \sum_{(i,j)\in A} \tilde{c}_{ijp} x_{ijp} + \sum_{k\in D} (\pi_k \underline{V}_k - \lambda_k \overline{V}_k) z_k \right] \right\}$$

Subject to (4.1) and (4.2).

Let ψ be the partitioned vector $\left[\begin{array}{c}\lambda \\ \hline -\pi\end{array}\right]$. The Boxstep Method generates a sequence of ψ's, starting from an initial vector, and solving (DRS) with the additional constraint appended:

$$\|\psi - \psi^h\|_\infty \leq \beta$$

where β is a positive constant which represents one-half the dimension of a hyper-cube (box) centered at ψ^h. The idea is not new, but only recently has this concept been applied to mathematical programming, the credit for which belongs to Marsten, Hogan, and Blankenship [102].

Boxstep is based on a rather fundamental observation. If Y is some closed convex set over which some concave function v(y) is to be maximized, and B represents the hyper-cube for which $Y \cap B$ is non-empty, a point y* is globally optimal over Y if (1) y* is optimal over $Y \cap B$ and (2) y* is interior to the box. If y* lies on a boundary of B, the box is relocated to a new position such that B is centered at y* (or a more favorable position) and the algorithm repeats.

For each choice of ψ, and with z fixed, the inner bracketed term of (DRS) represents a point-wise minimum and is given by the sub-problem

$$\text{(SP)} \quad \text{Minimize} \quad \sum_{p \in P} \sum_{(i,j) \in A} \tilde{c}_{ijp} \, x_{ijp}$$

Subject to (4.1) and (4.2).

As indicated previously, (SP) separates into independent sub-problems of the form $(CTP)_p$.

Define Y^P as the convex hull for $(CTP)_p$. That is

$$Y^P = \{x_{ijp} : \sum_{j \in N} x_{ijp} - \sum_{j \in N} x_{jip} = b_{ip},\ 0 \leq x_{ijp} \leq u_{ijp};$$

$$i \in N,\ p \in P (i,j) \in A\}$$

Let $x_p(h)$ denote the h^{th} extreme point of Y^P. There is no need to be concerned about extreme rays and possible unboundedness since Y^P is bounded below.

An equivalent form of (DRS), then, is,

$$\underset{\lambda, \pi \geq 0}{\text{Max}} \left\{ \underset{h}{\text{Min}} \sum_{p \in P} \sum_{(i,j) \in A} \tilde{c}_{ijp} \, x_{ijp}(h) + \sum_{k \in D} (\pi_k \underline{V}_k - \lambda_k \overline{V}_k) \, z_k \right\}$$

This may equivalently be written in terms of hyper-planar cuts as follows:

$$\text{(DRB)} \quad \underset{\lambda,\pi \geq 0}{\text{Max}} \quad \phi(\lambda,\pi)$$

Subject to

$$\phi(\lambda,\pi) \leq \sum_{p\in P} \sum_{k\in D} [(x_{kp}(h) - \overline{V}_k z_k)\lambda_k - (x_{kp}(h) - \underline{V}_k z_k)\,\pi_k]$$

$$+ \sum_{p\in P} \sum_{(i,j)\in A} \tilde{c}_{ijp}\, x_{ijp}(h) \qquad h = 1, \ldots, H \qquad (4.6)$$

The Boxstep Method solves a relaxed version of (DRB) (that is, one which contains only a partial constraint set), and with the additional hyper-cube boundary constraints appended.

Therefore, the Boxstep Master Problem is as follows:

$$\text{(BSM)} \quad \underset{\lambda,\pi \geq 0}{\text{Max}} \quad \phi(\lambda,\pi)$$

Subject to

$$\lambda_k^h - \beta \leq \lambda_k \leq \beta + \lambda_k^h \qquad k\in D \qquad (4.7)$$

$$\pi_k \leq \beta - \pi_k^h \qquad k\in D \qquad (4.8)$$

and (4.6).

Let $\overline{\lambda}^h$, $\overline{\pi}^h$ solve (BSM). Marsten et al. [102] and Fisher [30] recommend as a preferred next position for the center of the box λ^{h+1}, π^{h+1} rather than the obvious choice $\overline{\lambda}^h$, $\overline{\pi}^h$, where

$$\lambda^{h+1} \leftarrow \lambda^h + \Xi^h\,(\overline{\lambda}^h - \lambda^h)$$

$$\pi^{h+1} \leftarrow \pi^h + \Xi^h\,(\overline{\pi}^h - \pi^h)$$

and, Ξ^h solves the following one-dimensional line search problem:

$$\text{Maximize}_{\Xi \geq 1} \quad \sum_{p\epsilon P} \sum_{(i,j)\epsilon A} (c_{ijp} + \lambda_{ij}^h - \pi_{ij}^h) + \sum_{k\epsilon D} (f_k - \overline{V}_k \lambda_k^h + \underline{V}_k \pi_k^h) z_k$$

$$+ \Xi \left[\sum_{p\epsilon P} \sum_{(i,j)\epsilon A} \left[(\overline{\lambda}_{ij}^h - \lambda_{ij}) - (\overline{\pi}_{ij} - \pi_{ij}) \right] x_{ijp}(h) \right.$$

$$\left. + \sum_{k\epsilon D} \left[(\overline{\pi}_k^h - \pi_k^h) \underline{V}_k - (\overline{\lambda}_k^h - \lambda_k^h) \overline{V}_k \right] z_k \right]$$

Marsten et al. [102] suggest an interesting variant to (BSM) which, in some cases, may accelerate convergence. The authors point out that the requirement that B be a hyper-cube may be too restrictive--in some cases it may be desirable to use a hyper-rectangle instead. This necessitates changing (4.7) and (4.8) in (BSM) to the following:

$$\lambda_k^h - \beta_k \leq \lambda_k \leq \beta_k + \lambda_k^h \qquad k\epsilon D$$

$$\pi_k \leq \beta_k - \pi_k^h \qquad k\epsilon D$$

Marsten et al. [102] also report some impressive results using a solution variant in which the Boxstep Method is used to complement the subgradient method. Evidently, subgradient optimization can very quickly reach the neighborhood of the optimum solution and the Boxstep Method works best if the first (not too large) box contains the optimum.

4.2 Benders Decomposition

Two decades ago Benders [7] pioneered a technique for solving separable mathematical programs, which divides a difficult problem into two simpler problems--one of which produces a sequence of upper bounds, the other a sequence of lower bounds. Successive solutions

of both problems generates a "homing" effect, in that the bounds converge on the optimum like a vice. This method, known as Benders Decomposition, has established itself squarely as a technique effective in solving some large scale distribution problems [55]. This approach has been investigated as a possible solution strategy for solving the Fixed Charge Multicommodity Flow Problem.

An equivalent form of (MDML) is

$$\text{(MDMLB)}\quad \text{Minimize } \{ \sum_{k\in D} f_k z_k + \text{Min} \sum_{p\in P} \sum_{(i,j)\in A} \Big| \sum_{j\in N} x_{ijp} - \sum_{j\in N} x_{ip}$$

$$= b_{ip},\ i\in N,\ p\in P;\ \underline{V}_k z_k \le \sum_{p\in P} x_{kp} \le \overline{V}_k z_k,\ k\in D;$$

$$0 \le x_{ijp} \le u_{ijp},\ (i,j)\in A,\ p\in P\}$$

Subject to (4.3).

For a temporarily fixed z-vector, the inner minimization is a subproblem called Benders Restriction.

$$\text{(RST)}\quad \text{Minimize} \sum_{p\in P} \sum_{(i,j)\in A} c_{ijp}\, x_{ijp}$$

Subject to

$$\underline{V}_k z_k \le \sum_{p\in P} x_{ijp} \le \overline{V}_k z_k \qquad k\in D \qquad (4.9)$$

and (4.1) and (4.2).

This is a lower-bounded multicommodity network flow problem. Since the feasible solution space of (RST) varies as the z vector is changed, it is more convenient to solve the dual of (RST) which has a fixed solution space:

$$\text{(RSTD)}\quad \text{Maximize} \sum_{p\in P} \sum_{i\in N} \omega_{ip} b_{ip} + \sum_{k\in D} (\underline{\rho}_k \underline{V}_k - \overline{\rho}_k \overline{V}_k)\, z_k$$

$$- \sum_{p\in P} \sum_{(i,j)\in A} \mu_{ijp}\, u_{ijp}$$

Subject to

$$\omega_{ip} - \omega_{jp} - \mu_{ijp} \leq c_{ijp} \quad (i,j)\in A\text{-}D,\ p\in P \tag{4.10}$$

$$\omega_{kp} - \omega_{k+1,p} + \underline{\rho}_k - \overline{\rho}_k - \mu_{k,k+1,p} \leq c_{k,k+1,p} \quad k\in D,\ p\in P \tag{4.11}$$

$$\underline{\rho}_k,\ \overline{\rho}_k \geq 0 \quad k\in D \tag{4.12}$$

$$\mu_{ijp} \geq 0 \quad (i,j)\in A,\ p\in P \tag{4.13}$$

$$\omega_{ip} \text{ unrestricted} \quad i\in N,\ p\in P \tag{4.14}$$

Substituting (RSTD) into (MDMLB) yields

$$\text{Minimize} \left\{ \sum_{k\in D} f_k z_k + \text{Max} \left[\sum_{p\in P} \sum_{i\in N} \omega_{ip} b_{ip} + \sum_{k\in D} (\underline{\rho}_k \underline{V}_k - \overline{\rho}_k \overline{V}_k)\, z_k \right.\right.$$

$$\left. - \sum_{p\in P} \sum_{(i,j)\in A} \mu_{ijp} u_{ijp} \,\middle|\, \omega_{ip}, \underline{\rho}_k, \overline{\rho}_k, \mu_{ijp} \in \Lambda^p, \right.$$

$$\left.\left. i\in N,\ k\in D,\ (i,j)\in A\text{-}D \right] \right\}$$

Subject to (4.3)

where

$$\Lambda^p = \{ \omega_{ip}, \underline{\rho}_k, \overline{\rho}_k, \mu_{ijp} : (4.9) - (4.13) \text{ are satisfied} \}$$

Now let $\{\omega_p(h),\ \underline{\rho}(h),\ \overline{\rho}(h),\ \mu_p(h)\}$ be the h^{th} extreme point of Λ^p. (MDMLB) can be expressed as

$$\text{Minimize} \sum_{k\in D} f_k z_k + \underset{h}{\text{supremum}} \sum_{p\in P} \sum_{i\in N} \omega_{ip}(h)\, b_{ip} + \sum_{k\in D} (\underline{\rho}_k(h) \underline{V}_k$$

$$- \overline{\rho}_k(h) \overline{V}_k)\, z_k - \sum_{p\in P} \sum_{(i,j)\in A} \mu_{ijp}(h)\, u_{ijp}$$

subject to (4.3).

Now let

$$\sigma(h) = \sum_{p\in P} \sum_{i\in N} \omega_{ip}(h)\, b_{ip} + \sum_{k\in D} (\underline{\rho}_k(h)\, \underline{V}_k - \overline{\rho}_k(h)\, \overline{V}_k)\, z_k$$

$$- \sum_{p\in P} \sum_{(i,j)\in A} \mu_{ijp}\, u_{ijp} \qquad h = 1, \ldots, H$$

Then, an equivalent form of (MDMLB) is

(BD) Minimize $\sum_{k\epsilon D} f_k z_k + \underset{h}{\text{supremum}}\ \sigma(h)$

or

Min $\sum_{k\epsilon D} f_k z_k + \sigma$

Subject to

$$\sigma \geq \sum_{p\epsilon P} \sum_{i\epsilon N} \omega_{ip}(h) b_{ip} + \sum_{k\epsilon D} (\underline{\rho}_k(h)\underline{V}_k - \overline{\rho}_k(h)\overline{V}_k) z_k$$

$$- \sum_{p\epsilon P} \sum_{(i,j)\epsilon A} \mu_{ijp}\, u_{ijp} \quad h = 1, \ldots, H \qquad (4.15)$$

and (4.3).

A difficulty with solving (BD) is that total enumeration of all extreme points is necessary. The Benders strategy solves a relaxed form of (BD) starting with no constraints and judiciously adding constraints one-at-a-time as extreme points are generated by the subproblems (RSTD).

This relaxed master problem, called Benders Relaxation, is as follows:

(BDR) Minimize $\sum_{k\epsilon D} f_k z_k + \sigma$

Subject to (4.3) and (4.15R).

where (4.15R) is a subset of (4.15) which only contains the first $\overline{H} \leq H$ of all the "Benders Cuts" in the set.

Since the solution to (BDR) involves solving a sequence of progressively tighter relaxations of (BD) the resulting solutions are a sequence of lower bounds on (MDML) which is monotone increasing.

Solution of (RSTD) produces a non-monotonic sequence of upper bounds on (MDML). Termination of the Benders algorithm is based on convergence of the tightest upper and lower bounds achieved.

The Benders Decomposition Algorithm is as follows:

Step 0 Initialize $h = 0$, $UB = +\infty$, $LB = -\infty$. Select $\varepsilon \geq 0$, z^0

Step 1 Solve (RSTD) using z^h fixed. Let $\upsilon(RSTD^h)$ denote the optimal objective function value and $\omega(h)$, $\underline{\rho}(h)$, $\overline{\rho}(h)$, and $\mu(h)$ the optimal vectors.

Set $UB \leftarrow \text{Min } \{UB, \upsilon(RSTD^h)\}$

Step 2 Solve BDR. Let $\overline{z}^h$ denote the optimal solution vector and scalar respectively for this problem, and let $\upsilon(BDR^h)$ denote the corresponding objective function value.

Set $LB \leftarrow \upsilon(BDR^h)$

Step 3 If $UB - LB \leq \varepsilon$ terminate. Current solution is ε-optimal.

Step 4 $z_k^{h+1} \leftarrow \overline{z}_k^h$

$h \leftarrow h + 1$

Go to Step 1.

Problem (BDR) is a mixed integer programming problem in variables σ and z. As such, it is difficult to solve. Geoffrion and Graves [55] suggest a pure integer variant which is "feasibility-seeking" only, and which apparently works well in practice, particularly when the number of Benders cuts is small. This variant, which is solved for a feasible solution only is as follows:

$$\text{(VAR)} \quad \text{Minimize} \sum_{k \in D} f_k z_k + \sum_{p \in P} \sum_{i \in N} \omega_{ip}(\overline{H}) b_{ip} + \sum_{k \in D} (\underline{\rho}_k(\overline{H}) \underline{V}_k - \overline{\rho}_k(\overline{H})(\overline{V}_k)\, z_k - \sum_{p \in P} \sum_{(i,j) \in A} \mu_{ijp}(\overline{H})\, u_{ijp}$$

Subject to

$$\sum_{k\in D} f_k z_k + \sum_{p\in P}\sum_{i\in N} \omega_{ip}(h) b_{ip} + \sum_{k\in D} (\underline{\rho}_k(h)\overline{V}_k - \overline{\rho}_k(h)\overline{V}_k)\, z_k$$

$$- \sum_{p\in P}\sum_{(i,j)\in A} \mu_{ijp}(h)\, u_{ijp} \leq UB - \varepsilon \quad h = 1, \ldots, \overline{H}$$

and (4.3).

It is noted that when this variant is used the mechanism for generating lower bounds no longer exists so this feature of the algorithm must be de-activated.

Kennington and Shalaby [85] have specialized Geoffrion's Resource-Directive approach [41] to the solution of the multicommodity network flow problem. Kennington and Helgason [84] propose a Price-Directive algorithm. These results are readily available for solving the sub-problems (RSTD).

The Price Directive and Resource Directive schemes are complementary duals. Price direction is concerned with finding the optimal vector of Lagrange multipliers (dual variables) associated with independent subproblems, separated by product, such that the solution to the subproblems also solves the original, constraint-coupled, master problem.

Resource direction is concerned with finding the optimal assignment vector of total arc capacities to each commodity. For any fixed assignment, the problem de-couples into commodity-independent subproblems. The objective is to find an assignment vector which is optimal in the subproblems and also solves the master problem.

Geoffrion [41] recommends Resource Directive approaches to Price Direction because the former, a primal method, starts with a feasible

solution and preserves feasibility, whilst the latter, a dual method does not produce feasibility until termination. In practice, if iterations are halted short of optimality, Resource Direction will assure at least a good feasible solution.

4.2.1 Resource Direction

For clarity, the primal form of Benders Restriction will be re-stated:

$$\text{(RST)} \quad \text{Minimize} \quad \sum_{p\in P} \sum_{(i,j)\in A} c_{ijp}\, x_{ijp}$$

Subject to (4.1), (4.2), and (4.9).

Let $\bar{y}_p$, $\underline{y}_p$ be allocation vectors of $\bar{V}, \underline{V}$ respectively to commodity p.

(RST) may be rewritten:

$$\text{(RSD)} \quad \underset{\bar{y},\underline{y}}{\text{Min}} \left\{ \text{Min} \left[\sum_{p\in P} \sum_{(i,j)\in A} c_{ijp}\, x_{ijp} \mid x_{ijp} \in \Delta \quad (i,j)\in A,\ p\in P \right] \right\}$$

Subject to

$$\sum_{p\in P} \bar{y}_{kp} = \bar{V}_k z_k \qquad k\in D \qquad (4.16)$$

$$\sum_{p\in P} \underline{y}_{kp} = \underline{V}_k z_k \qquad k\in D \qquad (4.17)$$

$$0 \le \bar{y}_{kp} \le u_{kp} \qquad k\in D,\ p\in P \qquad (4.18)$$

$$\underline{y}_{kp} \ge 0 \qquad k\in D,\ p\in P \qquad (4.19)$$

where

$$\Delta = \{x_{ijp} : \underline{y}_{kp} \le x_{kp} \le \bar{y}_{kp} \quad k\in D,\ p\in P;\ \text{and (4.1) and (4.2) are satisfied}\}$$

By duality theory (RST) is equivalent to

$$(\overline{RSD}) \quad \underset{\overline{y},\underline{y}}{Min} \left\{ Max \sum_{p\epsilon P} \left[\sum_{i\epsilon N} \omega_{ip} b_{ip} - \sum_{(i,j)\epsilon A-D} \mu_{ijp} u_{ijp} \right.\right.$$

$$\left. + \sum_{k\epsilon D} (\underline{\rho}_{kp}\underline{y}_{kp} - \overline{\rho}_{kp}\overline{y}_{kp}) \mid \omega_{ip}, \mu_{ijp}, \overline{\rho}_{kp}, \underline{\rho}_{kp} \epsilon \overline{\Delta}(i,j)\epsilon A \right.$$

$$\left.\left. p\epsilon P,\ k\epsilon D \right] \right\}$$

Subject to (4.16)-(4.18)

where

$$\overline{\Delta} = \{\omega_{ip}, \mu_{ijp}, \overline{\rho}_{k}, \underline{\rho}_{k} : \text{(4.10-4.14) are satisfied}\}$$

The subproblem denoted by the inner maximization is decomposable by commodity. Each commodity-independent subproblem is a capacitated transshipment problem $(CTP)_p$ of the following form:

$$(SB)_p \quad Max \sum_{i\epsilon N} \omega_{ip} b_{ip} - \sum_{(i,j)\epsilon A-D} \mu_{ijp} u_{ijp} + \sum_{k\epsilon D} (\underline{\rho}_{kp}\underline{y}_{kp} - \overline{\rho}_{kp}\overline{y}_{kp})$$

Subject to

$$\omega_{ip} - \omega_{jp} - \mu_{ijp} \leq c_{ijp} \qquad (i,j)\epsilon A-D \qquad (4.20)$$

$$\omega_{ip} - \omega_{jp} - \overline{\rho}_{kp} + \underline{\rho}_{kp} \leq c_{kp} \qquad (i,j) : i\epsilon D,\ k\epsilon D \qquad (4.21)$$

$$\mu_{ijp}, \underline{\rho}_{kp}, \overline{\rho}_{kp} \geq 0 \qquad \text{all } i, j, k$$

$$\omega_{ip} \text{ unrestricted} \qquad i\epsilon N,\ p\epsilon P$$

Let $\phi(\overline{y}_p, \underline{y}_p)$ = solution to $(SB)_p$ for given allocation vectors $\overline{y}_p, \underline{y}_p$.

Let $\overline{\Delta}_p \subset \overline{\Delta}$ denote the feasible subspace of $\overline{\Delta}$ which is specific to commodity p.

Let $(\omega_p(h), \mu_p(h), \underline{\rho}_p(h), \overline{\rho}_p(h))$ be the h^{th} extreme point of $\overline{\Delta}_p$. Then, $\overline{RSD}$ can be written:

$$(MST) \quad \underset{\overline{y},\underline{y}}{Min} \sum_{p\epsilon P} \phi(\overline{y}_p, \underline{y}_p)$$

Subject to

$$\phi(\bar{y}_p, \underline{y}_p) \geq \sum_{i \in N} \omega_{ip}(h) b_{ip} - \sum_{(i,j) \in A-D} \mu_{ijp}(h) u_{ijp}$$

$$+ \sum_{k \in D} (\underline{\rho}_{kp} \underline{y}_{kp} - \bar{\rho}_{kp} \bar{y}_{kp}) \qquad p \in P \quad h = 1, \ldots H$$

and (4.16)-(4.19).

In practice, a Benders approach is taken in which a relaxed form of (MST) is solved. That is, at any iteration the constraint set of RSD only contains a partial set of extreme points (Benders cuts).

4.2.1.1 Solution by tangential approximation. The method of tangential approximation calls for the iterative and successive solutions of: first, with $\bar{y}, \underline{y}$ fixed, the subproblems $(SB)_p$ for optimal ω_p, μ_p, $\bar{\rho}_p$, $\underline{\rho}_p$ vectors, and then, with ω_p, μ_p, $\bar{\rho}_p$, $\underline{\rho}_p$ fixed, the master problem (MST) for new $\bar{y}, \underline{y}$ vectors. The tangential approximation algorithm is as follows:

Step 0 Initialize. Set $h \leftarrow 0$, $UB \leftarrow \infty$, $LB \leftarrow -\infty$

Select $y^0, \underline{y}^0$ which are feasible in (RSD).

Select $\varepsilon \geq 0$.

Step 1 Temporarily fixing $\bar{y}$ and $\underline{y}$ at $\bar{y}^h$ and $\underline{y}^h$ respectively, solve $(SB)_p$ for optimal $\omega_p(h)$'s, $\mu_p(h)$'s, and $\bar{\rho}_p(h)$'s and $\underline{\rho}_p(h)$'s.

Let $\upsilon(SB)_p^h$ denote the optimal objective function value.

Set $UB \leftarrow \upsilon(SB)_p^h$.

Step 2 Temporarily fixing ω, μ, $\underline{\rho}$, and $\bar{\rho}$ at $\omega(h)$, $\mu(h)$, $\underline{\rho}(h)$, and $\bar{\rho}(h)$, solve (MST) for new $\bar{y}, \underline{y}$ vectors.

Let $\upsilon(MST)^h$ be the optimal objective function value.

Set $LB \leftarrow \upsilon(MST)^h$.

Set $\bar{y}^h \leftarrow \bar{y}$ $\quad \underline{y}^h \leftarrow \underline{y}$

Step 3 If UB - LB $\leq \varepsilon$, terminate. $x(h)$ is ε-optimal.

Set $h \leftarrow h+1$

Set $\overline{y}^h \leftarrow \overline{y}^{h-1}$

Set $\underline{y}^h \leftarrow \underline{y}^{h-1}$

Go to Step 1.

4.2.1.2 Solutions using subgradients. An alternate method for solving (RSD) involves the use of subgradients for generating an improving sequence of allocation vectors.

Let $\varepsilon = [- \overline{\underline{y}} -]$. The subgradient algorithm produces a new vector by projecting a point $\xi^h - \theta^h \eta^h$ onto the y solution space Λ_y. The projection operation locates the new vector ξ^{h+1} as the closest feasible point to $\xi^h - \theta^h \eta^h$ with respect to the Euclidean norm. This is illustrated in Figure 4.3, where θ^h is a scalar stepsize, η^h is a subgradient of $\phi(\xi)$ at ξ^h, and

$$\Lambda_y = \{\varepsilon : \sum_p \overline{y}_p = \overline{V},\ \sum_p \underline{y}_p = \underline{V},\ \overline{y},\ \underline{y} \geq 0\}$$

Kennington and Helgason [84] and Geoffrion [41] prove that for the upper bounded case a subgradient of $\phi(y) = \sum_p \phi(y_p)$ (that is, the objective of the master problem), for some fixed allocation $\overline{y}$, is the vector $-\rho$. This result can readily be extended to include the lower bounded case as well.

Let $\delta = [- \overline{\underline{\rho}} -]$. It can be shown that $\delta(h)$ is a subgradient of $\phi(\xi)$ at some point ξ^h.

$$\begin{aligned}
\phi(\xi) - \phi(\xi^h) &= b^T\omega - u^T\mu + \delta \cdot \xi - b^T\omega(h) \\
&\quad + u^T\mu(h) - \delta(h) \cdot \xi^h \geq b^T\omega(h) - u^T\mu(h) \\
&\quad + \delta(h) \cdot \xi - b^T\omega(h) + u^T\mu(h) - \delta(h) \cdot \xi^h \\
&= \delta(h) \cdot (\xi - \xi^h)
\end{aligned}$$

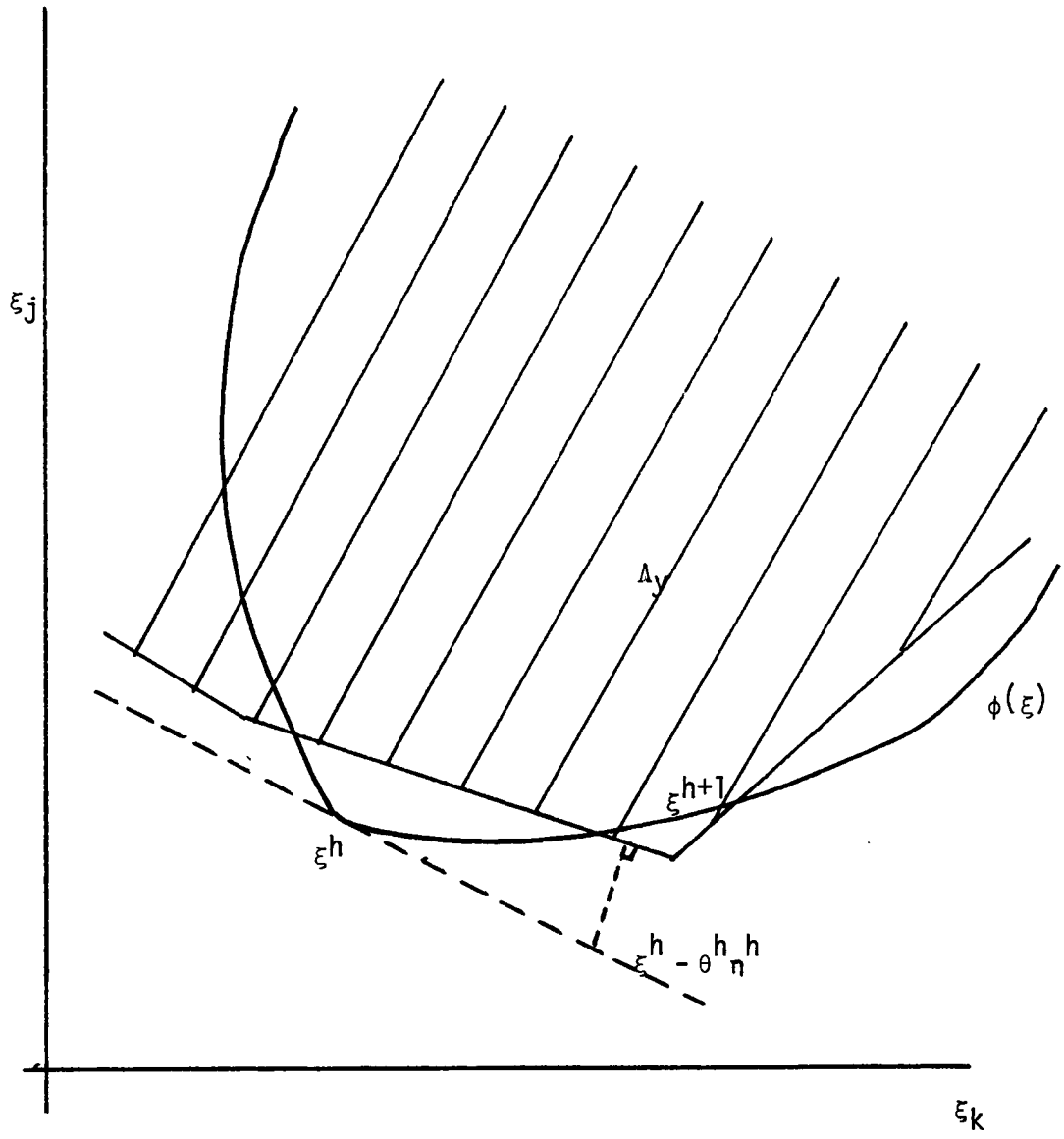

Figure 4.3. Illustration of Subgradient Projection Operation.

The projection operation can be summarized as follows:

Given some point $\xi^* \notin$ CTP-solution space Λ, the projection of ξ^* onto Λ, $P[\xi^*]$, requires solving the following quadratic programming problem:

$$(P) \quad \underset{\xi \in \Lambda}{\text{Min}} \{ \| \xi - \xi^* \| \mid \sum_p \overline{y}_{kp} = \overline{V}_k z_k, \ \sum_p \underline{y}_{kp} = \underline{V}_k z_k \quad k \in D;$$

$$0 \leq \overline{y}_{kp} \leq u_{kp}, \ \underline{y}_{kp} \geq 0, \ p \in P, \ k \in D \}$$

or, equivalently

$$\underset{\substack{\overline{y} \in \Lambda \\ \underline{y} \in \Lambda}}{\text{Min}} \Big\{ \Big(\sum_{p \in P} \Big(\sum_{k \in D} (\overline{y}_{kp} - \overline{y}_{kp}^*)^2 + \sum_{k \in D} (\underline{y}_{kp} - \underline{y}_{kp}^*)^2 \Big)^{1/2} \Big|$$

$$\sum_{p \in P} \overline{y}_{kp} = \overline{V}_k z_k, \ \sum_{p \in P} \underline{y}_{kp} = \underline{V}_k z_k \quad k \in D;$$

$$0 \leq \overline{y}_{kp} \leq u_{kp}, \ \underline{y}_{kp} \geq 0, \quad p \in P, \ k \in D \Big\}$$

which is also equivalent to solving

$$\underset{\overline{y} \in \Lambda}{\text{Min}} \sum_{p \in P} \sum_{k \in D} (\overline{y}_{kp} - \overline{y}_{kp}^*)^2 + \underset{\underline{y} \in \Lambda}{\text{Min}} \sum_{p \in P} \sum_{k \in D} (\underline{y}_{kp} - \underline{y}_{kp}^*)^2$$

subject to (4.16)-(4.19).

It is noted that (P) conveniently decomposes on k and also is separable on $\overline{y}$ and $\underline{y}$. (P) can therefore be solved by solving the simpler subproblems of the form $(\overline{P}_k)$ and $(\underline{P}_k)$:

$$(\overline{P}_k) \quad \underset{\overline{y} \in \Lambda}{\text{Min}} \sum_{p \in P} (\overline{y}_{kp} - \overline{y}_{kp}^*)^2 \quad \text{subject to (4.16) and (4.18)}$$

$$(\underline{P}_k) \quad \underset{\underline{y} \in \Lambda}{\text{Min}} \sum_{p \in P} (\underline{y}_{kp} - \underline{y}_{kp}^*)^2 \quad \text{subject to (4.17) and (4.19).}$$

Kennington and Helgason [84] have presented an efficient algorithm for solving $(\overline{P}_k)$ and $(\underline{P}_k)$, which is based on original results reported by Held, Wolfe, and Crowder [72].

The subgradient algorithm follows:

Step 0 Initialize Set $h \leftarrow 0$, $\gamma^0 \leftarrow 2$, $\nu \leftarrow 2^6$, $UB \leftarrow +\infty$.

Select $\overline{y}^{*0}$, $\underline{y}^{*0}$, $\varepsilon \geq 0$.

Select iteration count limit H.

Solve subproblems $(SB)_p$ for $\omega_p(0)$, $\mu_p(0)$, $\overline{\rho}_p(0)$, $\underline{\rho}_p(0)$.

Set $LB \leftarrow \sum_p \phi(\overline{y}_p^{*0}\ \underline{y}_p^{*0})$

Set $\overline{y}^0 \leftarrow P[\overline{y}^{*0}]$

Set $\underline{y}^0 \leftarrow P[\underline{y}^{*0}]$

Step 1 Find subgradient.

Solve subproblems $(SB)_p$ for $\omega_p(h)$, $\mu_p(h)$, $\overline{\rho}_p(h)$, $\underline{\rho}_p(h)$.

Set $UB \leftarrow \sum_p \phi(\overline{y}_p^h, \underline{y}_p^h)$.

Step 2 Is $UB - LB \leq \varepsilon$? If so, terminate with Step 1 solution.

Otherwise, continue.

Step 3 Is $h = H$? If so, terminate with Step 1 solution.

Otherwise, compute

$$\theta^h = \gamma^h(UB-LB)/\sum_{p\varepsilon P} (\overline{\rho}_p(h)\overline{\rho}_p(h) + \underline{\rho}_p(h)\underline{\rho}_p(h))$$

and

Set $\overline{y}_{kp}^{*h} \leftarrow \overline{y}_{kp}^h + \theta^h\ \overline{\rho}_{kp}(h)$

$\underline{y}_{kp}^{*h} \leftarrow \underline{y}_{kp}^h - \theta^h\ \underline{\rho}_{kp}(h)$

Step 4 Set $\overline{y}_{kp}^{h+1} \leftarrow P[\overline{y}_{kp}^{*h}]$

Set $\underline{y}_{kp}^{h+1} \leftarrow P[\underline{y}_{kp}^{*h}]$

Step 5 Set $h \leftarrow h+1$. Go to Step 1.

4.3. Ingredients for Algorithmic Design

An algorithm for solving the Fixed Charge Multicommodity Flow Problem has been an important product of the present research. The design of this algorithm, which is discussed fully in Chapter V of this thesis, follows a branch and bound framework with a heuristic to provide an advanced starting node position in the enumeration tree, and an efficient branch ordering from the root to that initial node.

4.3.1 Lagrangean Relaxation

Lagrangean relaxation has been used to solve the nodal sub-problems with a subgradient price improvement routine. An iteration limit is placed on this scheme, and if termination does not occur prior to reaching this limit, the best sub-optimal Lagrangean solution is input to a Resource Directive scheme (in which the objective function is adjusted back to its original form) and the solution process is completed.

The Lagrangean relaxation is also used as a bounding problem to determine if any completions of a current node could possibly produce a new incumbent. The multipliers used in these problems are the most recent set generated by the subgradient scheme.

4.3.2 Resource Direction

Although Benders Decomposition is not used in the proposed algorithm, the knowledge of this technique has influenced the design process. With a fixed configuration at each node of the enumeration

tree, the subproblem at that node constitutes a Benders Restriction, which has been shown to be a multicommodity flow problem (MCTP). Resource Direction as a means for solving MCTP's has appeal, since premature termination will insure a feasible albeit sub-optimal solution. If a procedure is used which quickly converges on the neighborhood of the optimum, this feasible solution will be a near-optimal one. The algorithm described in Chapter V attempts to bridge the best of two concepts. An optimal dual solution is first sought through the Lagrangean price adjustment routine. This "Price Directive" loop drives the solution toward feasibility and therefore optimality. On termination of Price Direction these sub-optimal (often near-optimal) solutions are projected onto the solution space and the projected vector constitutes an initial allocation of resources across the commodities. Commodity-independent capacitated transshipment problems (CTP's) are solved and subgradient methods are used to locate improving allocation vectors.

4.3.3 Branching Rules Based on Benders Decomposition

The Benders Relaxation inspired some limited experimentation with some unconventional branching rules based on selecting that variable which is most likely to improve the value of the Benders Relaxation objective. The evidence is inconclusive, and more work needs to be done in this area. This topic is treated in more detail in Chapter V.

CHAPTER V

MULTI-ECHELON DISTRIBUTION OPTIMIZATION SYSTEM: A NEW ALGORITHM, MEDOS

When a branch and bound framework is used to solve the Fixed Charge Multicommodity Flow Problem, each node of the enumeration tree represents a simpler multicommodity network flow problem (with a fixed configuration of binary variables). Multicommodity flow problems are not easily solved, but the solution of many of the subproblems can be avoided by the administration of logical supply and demand tests, and by effective bounding. A bounding problem which has proven to be effective is a relaxation in which the multicommodity constraints are deleted from the constraint set of the multicommodity network flow problem. The relaxations are capacitated transshipment problems which separate on commodity and for which streamlined solution procedures exist [10]. It is the efficiency with which the bounding problems can be solved that renders branch and bound appealing as a solution approach.

This chapter presents a new algorithm which uses a hybrid price-directive/resource-directive scheme with subgradient optimization embedded within a branch and bound framework. Lagrangean relaxation is used to obtain lower bounds on the multicommodity network flow problem and the solutions generated from the relaxations from the basis for fixing initial simple bounds for the capacitated arcs. A sequence of restrictions, which are capacitated transshipment problems are solved, each solution forming the basis for adjusting the simple bounds.

The solutions to the restrictions are upper bounds on the configuration and on the fixed charge problem. These solutions are used to update the lower bounds (that is, solve a relaxation with adjusted prices) the solution to which can be used, if needed, to generate another vector of simple bounds for the restriction. Termination occurs when the best configuration upper bound is within some desired percentage (ε) of the lower bound.

In describing the proposed new technique, this chapter is organized in such a way that intrinsic detail is segregated from important fundamentals for the sake of clarity of exposition. Section 5.1 defines the problems of interest and certain additional required notation, and outlines the major steps of the algorithm. Section 5.2 describes in detail the fathoming, screening, and branching devices used by the algorithm. Section 5.3 discusses the combined use of price direction and resource direction as solution techniques. Section 5.4 details certain model enhancements, and finally Section 5.5 outlines the steps of the heuristic algorithm which is used to obtain an initial configuration and upper bound.

5.1 Description of the MEDOS Algorithm

Problems:

Fixed Charge Linear Multicommodity Flow Problem

(MDML) Minimize $\sum_{p \in P} c_p x_p + fz$ (5.1)

Subject to

$$Ex_p = b_p \qquad p\epsilon P \qquad (5.2)$$

$$\underline{V}_k z_k \leq \sum_{p\epsilon P} x_p \leq \overline{V}_k z_k \qquad (5.3)$$

$$0 \leq x_p \leq u_p \qquad p\epsilon P \qquad (5.4)$$

$$z\epsilon \{0,1\} \qquad (5.5)$$

Multicommodity Flow Problem with Configuration Fixed

(MCTP(z)) Minimize $\sum_{p\epsilon P} c_p x_p$ (5.6)

Subject to

$$\underline{V}_k \leq \sum_{p\epsilon P} x_{kp} \leq \overline{V}_k \qquad k\epsilon DC(z) \qquad (5.7)$$

$$x_{kp} = 0 \qquad p\epsilon P,\ k\notin DC(z) \qquad (5.8)$$

and (5.2) and (5.4).

Multicommodity Flow Restriction with Configuration and Allocation Fixed

(RMCTP(z,y)) Minimize $\sum_{p\epsilon P} c_p x_p + fz$ (5.9)

Subject to

$$\underline{y}_{kp} \leq x_{kp} \leq \overline{y}_{kp} \qquad p\epsilon P,\ k\epsilon DC(z) \qquad (5.10)$$

and (5.2) and (5.8)

where

$$y = (\underline{y},\overline{y}) : \sum_{p\epsilon P} \underline{y}_{kp} = \underline{V}_k$$

$$\sum_{p\epsilon P} \overline{y}_{kp} = \overline{V}_k$$

Commodity-Specific Multicommodity Flow Restriction: Capacitated Transshipment Problem (CTP)*

For a fixed z,

(RMCTP(z,y)) decomposes on commodity into problems of the form

$$(RMCTP_p(z,y)) \quad \text{Minimize } c_p x_p \tag{5.11}$$

Subject to

$$Ex_p = b_p \tag{5.12}$$

$$\underline{y}_{kp} \leq x_{kp} \leq \bar{y}_{kp} \qquad k \in DC(z) \tag{5.13}$$

$$0 \leq x_p \leq u_p \tag{5.14}$$

$$x_{kp} = 0 \qquad k \notin DC(z) \tag{5.15}$$

Multicommodity Flow Relaxation with Fixed Configuration and Prices

$$(LMCTP(z,\pi)) \quad \text{Minimize } \sum_{p \in P} \bar{c}_p \bar{x}_p + \bar{f}z \tag{5.16}$$

Subject to (5.2), (5.4), and (5.8)

where

$$\bar{c}_{kp} = c_{kp} - \underline{\pi}_k + \bar{\pi}_k \qquad k \in DC(z)$$

$$\bar{f}_k = f_k + \underline{\pi}_k \underline{V}_k - \bar{\pi}_k \bar{V}_k$$

and

$$\pi = (\underline{\pi}, \bar{\pi}) \geq 0$$

Commodity-Specific Multicommodity Flow Relaxation: Capacitated Transshipment Problem (CTP)*

For a fixed z,

(LMCTP(z,π)) decomposes on commodity into

---

*The solution technique used for solving the CTP's is Bradley, Brown, and Graves' GNET and XNET [10]. More details on the use of these algorithms by MEDOS are contained in Chapter VI.

$$(\text{LMCTP}_p(z,\pi))\ \text{Minimize}\ \bar{c}_p\bar{x}_p \tag{5.17}$$

Subject to (5.12), (5.14), and (5.15)

$$(\text{LBC}(\tilde{z},\pi))\ \text{Minimize}\ \sum_{p\in P}\bar{c}_p\bar{x}_p + \sum_{k\in DC(z)}\bar{f}_k z_k + \min_{k\in FR} f_k + \sum_{k\in DC(z)\cup FR}(\underline{\pi}_k\underline{V}_k - \bar{\pi}_k\bar{V}_k) \tag{5.18}$$

Subject to (5.2), (5.4), and (5.8)

where

$$\tilde{z}_k = 1 \text{ for } k\in DC(z)\cup FR$$

Additional Notation:

- $G=(N,A)$ - graph for distribution network.
- $DC(z)$ - denotes the set of DC's which are open, in fixed configuration z.
- FR - denotes the set of DC's which are currently free.
- E - node-arc incidence matrix corresponding to G.
- $x^*(z)$ - optimum solution to MCTP(z) with configuration fixed at z.
- $x^*(z,y)$ - optimum solution to RMCTP(z,y) with configuration fixed at z and resource allocation fixed at y.
- $x^*(z,\pi)$ - optimum solution to LMCTP(z,π) with configuration fixed at z and pricing vector fixed at π.
- $\bar{x}^*(\tilde{z},\pi)$ - optimum solution to LBC($\tilde{z}$,π) with configuration fixed at z and prices fixed at π.
- $\upsilon^*(\cdot)$ - optimum objective function value of $(\cdot)$.
- UB - value of the incumbent solution (UB on MCMDL).
- $UB(z)$ - upper bound on MCTP(z).
- $LB(z)$ - lower bound on MCTP(z).
- $LB(\tilde{z})$ - lower bound on completions of configuration z.

$y = (\underline{y},\overline{y})$ - an allocation of upper and lower throughputs to commodity-specific arcs.

$P[\cdot]$ - denotes projection of point $[\cdot]$ onto y-space.

ε - solution tolerance parameter.

L - the limit on the number of resource directive iterations between LB(z) updates.

K - the limit on the number of iterations performed in price directive loop.

Definitions:

Live Node - a node in the enumeration tree for which all completions have been fathomed.

Scanned Node - a node for which all completions have been explicitly or implicitly investigated.

Visited Node - a node which has been investigated for possible solution.

Infeasibility:

For the sake of clarity, a distinction is made between infeasibility arising from violation of conservation of flow equations and simple bounds (CTP-infeasibility) and that from violation of DC throughput constraints (throughput-infeasibility).

Specifically, a solution will be said to be

CTP - feasible if constraints (5.2) and (5.4) are obeyed.

TP - feasible if, additionally, constraints (5.3) are satisfied.

5.1.1. The MEDOS Algorithm (Refer to Figure 5.1)

Step 0 Initialization

Initialize z, DC(z), $\varepsilon > 0$

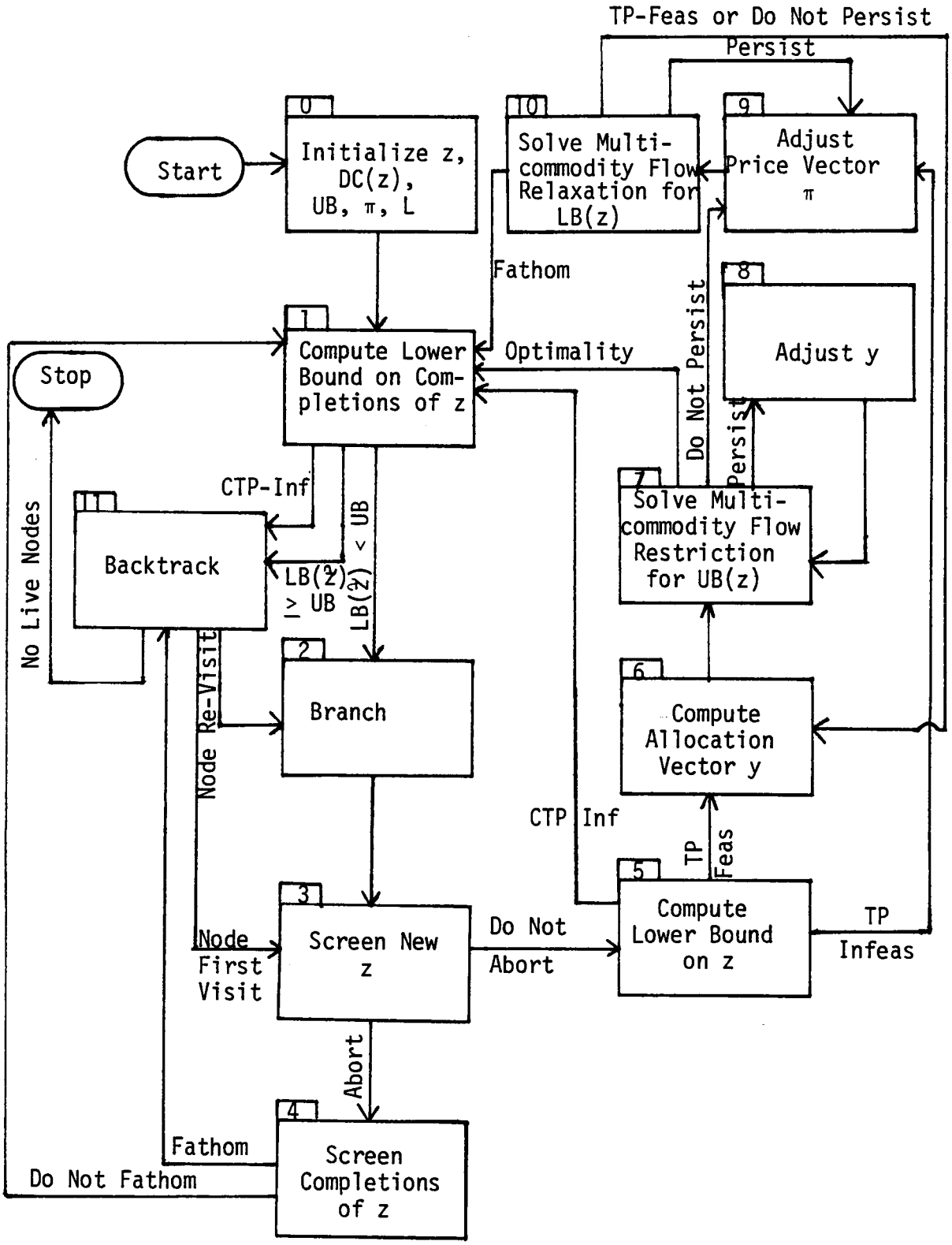

Figure 5.1. Flowchart for MEDOS Algorithm

Solve MCTP(z) for $x^*(z)$

Set UB $\leftarrow$ υ^*(MCTP(z))

Initialize π, L, K

Step 1 Compute Lower Bound on Completion

Set $\tilde{z}_k \leftarrow \begin{cases} 1 : k \in DC(z) \cup FR \\ 0 \quad \text{Otherwise} \end{cases}$

Solve LBC($\tilde{z}$,π) for $\bar{x}^*(\tilde{z},\pi)$

If LBC($\tilde{z}$,π) is CTP-infeasible, go to Step 11 (Backtrack)

Set LB($\tilde{z}$) $\leftarrow$ υ^*(LBC($\tilde{z}$,π))

If LB($\tilde{z}$) $\geq$ UB-ε, go to Step 11 (Backtrack)

Step 2 Branch

Select $k \in FR$ (details provided in Section 5.2.4)

Set DC(z) $\leftarrow$ DC(z)$\cup${k}

Set $z_k \leftarrow \begin{cases} 1 : k \in DC(z) \\ 0 \quad \text{Otherwise} \end{cases}$

Step 3 Perform Configuration Capacity Screening Tests (details provided in Section 5.2.1)

If z remains a contender, go to Step 5.

Step 4 Perform Completion Capacity Screening Tests (details provided in Section 5.2.1)

If z remains a contender, go to Step 1

Otherwise, go to Step 11 (Backtrack)

Step 5 Compute Lower Bound on Configuration

Set $\pi \leftarrow \tilde{\pi}$

Solve LMCTP(z,π) for $x^*(z,\pi)$

If $x^*(z,\pi)$ is CTP-infeasible, go to Step 1

Set LB$(z) \leftarrow \upsilon^*$(LMCTP$(z,\pi)$)

If $x^*(z,\pi)$ is TP-infeasible,

Set $i \leftarrow 0$, and go to Step 9a

Set UB $\leftarrow$ Min {UB, υ^*(LMCTP(z,π))}

Step 6 Compute an Allocation Vector y (details provided in Section 4.2.1.2)

If $x^*(z,\pi)$ is TP-feasible

Set $\overline{y}_{kp} \leftarrow t_{kp}\overline{V}_k$

$\underline{y}_{kp} \leftarrow t_{kp}\underline{V}_k$

where $t_{kp} = x^*_{kp}(z,\pi) / \sum_{p\varepsilon P} x^*_{kp}(z,\pi)$

Otherwise

Set $y^* \leftarrow x^*(z,\pi)$

$y \leftarrow P[y^*]$

Set $i \leftarrow 0$

Step 7 Solve Multicommodity Flow Restriction with Fixed z and y

Solve RMCTP(z,y) for $x^*(z,y)$

Set UB $\leftarrow$ Min {UB, υ^*(RMCTP(z,y))}

If υ^*(RMCTP(z,y)) $\leq$ LB$(z) + \varepsilon$, go to Step 1

If i=L, set $i \leftarrow 0$ and go to Step 9b.

Set $i \leftarrow i + 1$

Step 8 Adjust Resource Allocation (details provided in Section 4.2.1.2)

Set $y^* \leftarrow y - \theta\eta$

where η is a subgradient of (5.9) at y

and θ is a scalar stepsize

Step $y \leftarrow P[y^*]$

Go to Step 7

Step 9 Adjust Price Vector (details provided in Section 4.1.1.1)

a. Method used while within Price-Directive Loop

Set $\overline{\pi}_k \leftarrow \text{Max}\ \{\overline{\pi}_k - \delta\overline{u}, 0\}$

$\underline{\pi}_k \leftarrow \text{Max}\ \{\underline{\pi}_k - \delta\underline{u}, 0\}$

where $\overline{u}, \underline{u}$ are subgradients of (5.16) at $\overline{\pi}, \underline{\pi}$ respectively and δ is a scalar stepsize

Go to Step 10

b. Method used when transferring from Resource-Directive Loop

$$\text{Set } \overline{\pi}_k \leftarrow \begin{cases} 0 \text{ if } \sum_{p \in P} x_k < \overline{V}_k \\ \max_p \overline{\rho}_{kp} \text{ if } \sum_{p \in P} x_k = \overline{V}_k \end{cases}$$

$$\text{Set } \underline{\pi}_k \leftarrow \begin{cases} 0 \text{ if } \sum_{p \in P} x_k > \underline{V}_k \\ \min_p \underline{\rho}_{kp} \text{ if } \sum_{p \in P} x_{kp} = \underline{V}_k \end{cases}$$

Step 10 Solve Lagrangean Relaxation

Solve LMCTP(z,π) for $x^*(z,\pi)$

Set LB$(z) \leftarrow \text{Max}\ \{\text{LB}(z), \upsilon^*(\text{LMCTP}(z,\pi))\}$

If LB(z) $\geq$ UB, go to Step 1

If $x^*(z,\pi)$ is TP-infeasible and,

i = K, set $\tilde{\pi} \leftarrow \pi$ and go to Step 6.

Otherwise, set i $\leftarrow$ i+1 and go to Step 9.

If $x^*(z,\pi)$ is TP-feasible,

Set UB $\leftarrow$ Min {UB, υ^*(LMCTP(z,π))}

Set $\tilde{\pi} \leftarrow \pi$

Go to Step 6

Step 11 Backtrack

Close all DC's up to last live node (LIFO procedure)

If DC(z) = ϕ terminate

If node is unvisited and unscanned, go to Step 3

Otherwise, go to Step 2

5.2 Screening, Fathoming, and Branching Rules

5.2.1. Screening Tests

Whenever a new configuration is generated, two issues arise. First, is there any potential benefit in solving the multicommodity flow problem with the new configuration? And, secondly, is there any potential benefit in augmenting the new configuration by opening one of the free DC's? The first issue is dealt with by Steps 3 and 5 of the algorithm. Steps 1 and 4 cover the second issue.

Step 3 of the algorithm is called whenever a forward step, or branch action is taken, or whenever a backward step leads to a node

which has not previously been visited. The latter will be the case when the new node visited corresponds to a node which was on the stem of the initial configuration tree generated by the heuristic. Possible exits from Step 3 are Step 11 (backtrack) when z is fathomed, Step 5 when z remains a contender, and Step 4 when z ceases to be a contender, but its completions require investigation. A battery of three tests is performed in Step 3, as follows:

Test 3a. Reverse Star* Lower Bound Aggregate Test

If the aggregate lower throughput bounds of DC's which are open and which lie on customer reverse stars is greater than total aggregate demand, the configuration may be fathomed.

That is, if

$$\sum_{k \cup (DC(z) \cap RS(j))} \underline{V}_k > \sum_{j \in C} \sum_{p \in P} D_{jp} \qquad (5.19)$$

where $\underline{V}_k$ = lower throughput bound for DC k

D_{jp} = demand for product p by customer j

Go to Step 11 (Backtrack).

Otherwise administer Test 3b.

Test 3b. Reverse Star Configuration Capacity Test

If the demand for any customer is not covered by sufficient reverse star upper capacity, there is no benefit in solving the multicommodity flow problem for the present configuration. A new configuration should be generated. The completions of z require investigation

---

*The reverse star RS(j) of node j is the set of nodes $\{i: (i,j) \in A\}$. In the case of customer nodes, the reverse star includes all plants or DC's which are permitted to act as sources of supply.

(Steps 4 and 1 of the algorithm) in order to determine whether this configuration results from a backtrack or branch action. Formally, the test is,

If

$$\sum_{k\in DC(z)\cap RS(j)} \overline{V}_k < \sum_{p\in P} D_{jp} \text{ for any } j\in C \tag{5.20}$$

Go to step 4.

Otherwise administer Test 3c.

Test 3c. Configuration Aggregate Capacity Test

If the aggregate upper throughput capacity does not cover total demand the subproblem is capacity infeasible and the pursuit of a solution is not warranted. Specifically,

If

$$\sum_{k\in DC(z)} \overline{V}_k < \sum_{p\in P} \sum_{j\in C} D_{jp} \tag{5.21}$$

Go to Step 4 (Test 4b). Otherwise, go to Step 5.

Note that this may not be a strong test, since the multi-echelon structure enables a shipment to pass through more than one DC enroute to a customer. However, (5.21) is easy to check and if it holds, solution of the subproblem may certainly be aborted.

Step 4 of the algorithm screens the set of free DC's to determine if any completion of the current configuration is capacity feasible. Exits are Step 11 when no capacity feasible completion exists and Step 1 otherwise. The following two tests, analogous to Tests 3b and 3c respectively are performed.

Test 4a. Reverse Star Completion Capacity Test

If

$$\sum_{k\in DC(z)\cup[FR\cap RS(j)]} \overline{V}_k < \sum_{p\in P} D_{jp} \text{ for any } j\in C \qquad (5.22)$$

Go to Step 11 (Backtrack).

Otherwise administer Test 4b.

Test 4b. Aggregate Completion Capacity Test

If no completion exists for which aggregate upper bound throughput capacity covers aggregate total demand, the current configuration can be fathomed. That is,

If

$$\sum_{k\in DC(z)\cup FR} \overline{V}_k < \sum_{j\in C} \sum_{p\in P} D_{jp} \qquad (5.23)$$

Go to Step 11 (Backtrack). Otherwise go to Step 1.

Test 4b may be weak for the same reasons stated above under Test 3c. Generally, the aggregate capacity tests will be dominated by the reverse star capacity tests.

Computational experience has shown Tests 3b and 3c to be extremely effective in keeping the number of multicommodity flow problems (MCTP(z)) which must be solved very small (less than 1%) relative to the number of subproblems enumerated. An average of 71% of the total subproblems enumerated passed Test 3b and an additional 20% passed Test 3c. This means that 91% of the subproblems were aborted without having to solve the bounding problem (LMCTP(z,π)). The completion tests proved to be effective fathoming tools, again without the need for computing bounds. On the average 10% of the nodes enumerated were fathomed as a result of Test 4a, 25% as a result of Test 4b.

5.2.2 Bounding Problems

In Step 5 (LMCTP(z,π)) is solved to obtain a lower bound on the current configuration z. In Step 1 (LBC($\tilde{z},\pi$)) is solved for a lower bound on the completions of z. The main issue for either of these problems is the determination of π. The theory of Lagrangean relaxation assures valid lower bounds with any choice of non-negative π's. The π's actually used in the bounding problems (LMCTP(z,π)) and LBC($\tilde{z},\pi$)) are the most recent π's generated by the price-directive mechanism.

In Step 1, two cases must be considered, depending on whether the current configuration covers the reverse star demand for all customers.

Case 1: There exists a customer $j\varepsilon C$ for which (5.28) holds. In this case, any capacity-feasible completion must include at least one free DC in RS(j). Two bounds can be computed. The first bound is obtained by solving problem LBC'($\tilde{z},\pi$) where LBC'($\tilde{z},\pi$) is the modification of LBC($\tilde{z},\pi$) in which the objective function is replaced by

$$\text{Minimize} \sum_{p\varepsilon P} \bar{c}_p \bar{x}_p + \sum_{k\varepsilon DC(z)} \bar{f}_k \bar{z}_k + \min_{k\varepsilon FR\cap RS(j)} f_k + \sum_{k\varepsilon DC(z)\cup[FR\cap RS(j)]} (\underline{\pi}_k \underline{V}_k - \bar{\pi}_k \bar{V}_k)$$

Let $LB_1(\tilde{z}) = v(LBC'(\tilde{z},\pi))$.

The second bound takes cognizance of transportation economies which could be obtained by opening DC's which are not in RS(j). Such economies could only be realized by opening a minimum of two DC's. This bound is obtained by solving problem LBC"($\tilde{z},\pi$), where LBC"($\tilde{z},\pi$) is derived from LBC($\tilde{z},\pi$) by replacing $\min_{k\varepsilon FR} f_k$ in (5.18) with

$$\min_{k\varepsilon FR\cap RS(j)} f_k + \min_{k\varepsilon \overline{FR\cap RS(j)}} f_k$$

Let $LB_2(\tilde{z}) = \upsilon(LBC''(\tilde{z},\pi))$

Then $LB(\tilde{z}) = \text{Min } \{LB_1(\tilde{z}), LB_2(\tilde{z})\}$

Case 2: The current configuration satisfies the reverse star demands for all customers.

In this case,

Solve $LBC(\tilde{z},\Pi)$ to obtain the lower bound.

$LB_1(\tilde{z})$ is a valid lower bound on the completions of z because $LBC'(\tilde{z},\pi)$ is a Lagrangean relaxation of $MCTP(z,\pi)$ with respect to the most optimistic completion (that is, minimum transportation costs and minimum fixed charges), if the set of free DC's is limited to those which are in RS(j).

$LB_2(\tilde{z})$ and $\upsilon(LBC(\tilde{z},\pi)$ are also valid lower bounds. They each represent the Lagrangean relaxation of $MCTP(z,\pi)$ with respect to the most optimistic completion if any free DC can be selected. These two formulations differ only in the constant fixed charge term.

It is important to note that in a sequence of forward branching steps, the solutions to $LBC(\tilde{z},\pi)$ change only by the constant term $\text{Min}_{k\epsilon FR} f_k$. This is because branching steps always result in DC's being opened and can never result in closures. This observation enables an efficient update of $LB(\tilde{z})$ wherever a sequence of branching steps is encountered. Specifically, when moving from node $\tilde{z}$ to node $\tilde{z}'$, a completion of $\tilde{z}$,

$$LB(\tilde{z}') = LB(\tilde{z}) + \underset{k\epsilon FR'}{\text{Min}} f_k$$

where FR' is the set of free DC's with respect to node z'.

5.2.3 Branching Rules

Step 2 of the algorithm requires the selection of a free DC as the next facility to open. A number of branching rules have been tested and incorporated within the software package described in Chapter VI.

The branching rules tested make use of cumulative information which is collected and updated as the algorithm progresses. This information provides some insight into the cost relationships which exist between each of the candidate DC's.

Eight different branching rule options are embedded in the MEDOS code. Instead of treating these as different selection rules to be independently exercised, the algorithm utilizes combinations of rules in a prioritized hierarchy. The rules are:

1. Reverse Star Supply (RSS)
2. Maximum Upper Bound Violation (Max UB)
3. Maximum Throughput (Max TP)
4. Regional Capacity Augmentation (Reg Cap)
5. Minimum Cost Attractiveness Index (Min CI)
6. Maximum Sigma (Max Sigma)
7. Non-zero Lower Bound (Pos LB)
8. Greatest Marginal Savings (GMS)

5.2.3.1 RSS Rule. The RSS Rule selects a free DC which lies on the reverse star of a customer needing supply.

That is, select $k \varepsilon FR \cap RS(j)$

were j is a customer whose reverse star is capacity-deficient.

5.2.3.2 Max UB Rule. The Max UB Rule investigates the solution to $LBC(\tilde{z},\pi)$ or, if (5.20) holds, $LBC'(\tilde{z},\pi)$. The free DC which represents the greatest upper throughput constraint violation, if any, is selected.

That is, if (5.20) holds, select $k\epsilon FR'$ such that

$$\sum_{p\epsilon P} \overline{x}^{*}_{kp}(\tilde{z},\pi) - \overline{V}_k = \max_{i\epsilon FR'} \left\{ 0, \sum_{p\epsilon P} \overline{x}^{*}_{ip} - \overline{V}_i \right\}$$

Otherwise. select $k\epsilon FR$ such that

$$\sum_{p\epsilon P} \overline{x}^{*}_{kp}(\tilde{z},\pi) - \overline{V}_k = \max_{i\epsilon FR} \left\{ 0, \sum_{p\epsilon P} \overline{x}^{*}_{ip} - \overline{V}_i \right\}$$

5.2.3.3 Max TP Rule. The Max TP Rule investigates the solution to $LBC(\tilde{z},\pi)$ or, if (5.20) holds, $LBC'(\tilde{z},\pi)$ and opens the free DC which had the maximum throughput.

That is, if (5.20) holds, select $k\epsilon FR'$ such that

$$\sum_{p\epsilon P} \overline{x}^{*}_{kp}(\tilde{z},\pi) = \max_{i\epsilon FR'} \overline{x}^{*}_{ip}(\tilde{z},\pi)$$

Otherwise, select $k\epsilon FR$ such that

$$\sum_{p\epsilon P} \overline{x}^{*}_{kp}(\tilde{z},\pi) = \max_{i\epsilon FR} \overline{x}^{*}_{ip}(\tilde{z},\pi)$$

5.2.3.4 Reg Cap Rule. The Reg Cap Rule selects a DC, from the set of free DC's, which belongs to a region in which the total aggregate regional capacity is less than the aggregate regional demand.

That is, select $k\epsilon FR\cap R_\ell$, such that

$$\sum_{k\epsilon DC(\tilde{z})\cap R_\ell} \overline{V}_k < \overline{D}_{R_\ell} \qquad R_\ell \epsilon R$$

5.2.3.5 Min CI Rule. The Min CI Rule selects a free DC based on a relative cost attractiveness criterion. On input a cost attractiveness

index (CI) is computed for each DC candidate. It is asserted that problem-specific formulae for the CI indices may produce the most successful results. The only requirement for conformance with the MEDOS software logic is that the CI formula used produce a priority index for each DC on a scale of low to high indices corresponding to high to low importance respectively. Any deviations in the formula presented below will necessitate some internal software alterations, but such repairs will be minor.

The cost attractive index presently used by MEDOS takes the form,

$$CI_k = \frac{\delta(1+(1-\beta)B_k)(f_k+v_k\bar{V}_k)}{(1+\beta B_k)\bar{V}_k} + \frac{(1-\delta)(f_k+v_k\bar{V}_k)}{\bar{D}_{R_\ell}}, \quad k\varepsilon D\cap R \qquad (5.24)$$

where

$\delta\varepsilon[0,1]$ for $\bar{D}_{R_\ell} \neq 0$

$\delta = 1$ otherwise

B_k = number of times algorithm failed to fathom when DC k was added to the configuration (that is, the algorithm branched from DC k).

$\beta\varepsilon\{0,1\}$

This formula is defended on the basis of the following three-part rationale:

1. If DC k is operating close to its nominal capacity, the unit variable cost of throughput is given by the expression $(f_k+v_k\bar{V}_k)/\bar{V}_k$. Branch selection priority (small CI_k) should recognize the relative operating economies of competing candidates.

2. If transportation costs more than offset savings in DC operating costs, the economies of scale criterion of selection is not enough. One possible indicator, which is easily obtained in the course of program execution, is a variable B_k which contains the number of times the algorithm branched when the current configuration (in which DC k was the last DC opened) was a contender. This variable provides some, even though admittedly rough, measurement of the strength of the transportation-operating cost trade-offs which exist relative to DC k. A large B_k (high frequency of branching) suggests that DC k is a high risk choice for branching variable. Once identified, the question is how best to deal with it. Two diametrically different policies emerge. A β parameter setting of zero has an inflationary effect on each CI_k. This implements a "depth-first" policy which has the objective of clustering non-terminal nodes as deeply within the tree as possible. A β equal to one has a deflationary effect on the CI_k's and implements a "breath-first" policy in which non-terminal nodes are clustered as close to the root as possible.
3. Regional assignments, depending on the parameter setting for δ, also can impinge on the value of CI_k. Suppose two DC candidates have identical unit operating costs (based on volume at capacity), but each is assigned to a different region. If the aggregate demands of the two regions differ, intuition might favor the DC assigned to the region with the larger demand in the hope that a newly opened DC is more likely to be

utilized at or near its capacity operating level if the DC comes from a large rather than a small region. Whether or not this expectation is realized depends on the regional capacities which exist in the current configuration. The second term in the CI formula provides for the region-size influence. This influence can be entirely nullified by setting $\delta = 1$. Computational experience has been unable to validate any setting other than $\delta = 1$. In the application discussed in Chapters I and VII, a range of δ's were tried and no δ could be found which improved on this setting, a result which may not hold for other problems.

5.2.3.6 Max Sigma Rule. The Max Sigma Rule is motivated by the Benders Master (BD) (refer to Chapter IV), the solution to which is the optimum configuration for (MDML). The Max Sigma Rule uses the information obtained each time a new feasible solution is found to estimate the value of the optimal scalar variable in the Benders Relaxation (BDR).

(BDR) Minimize $fz + \sigma$

Subject to (4.3) and (4.15R)

If all of the variables were known, each time a new feasible solution was found, it would be possible to construct another constraint for (BDR). Since the configuration is also fixed each time subproblems are solved, it would be possible to reduce the right-hand-side of the new constraint to a scalar quantity. This scalar quantity could be used then to estimate the optimal scalar σ.

From linear programming theory it is known that if (BDR) is bounded then, at optimality, the equality must hold in at least one of the constraints. It could be said, then, that there exists a Sigma

Max which, with the optimal z-vector, will satisfy all constraints at optimality. A possible strategy, then, is to store a variable called SIGMAX equal to the largest right-hand-side of (4.15R) obtained so far.

The SIGMAX branch selection criterion, then, is based on evaluating the $\bar{H}^{th}$ Benders cut with σ = SIGMAX for the current configuration, augmented by each free DC one-at-a-time. Those candidates which, when added to the current configuration, obey the $\bar{H}^{th}$ Benders cut are placed in a queue. From the queue, the DC with the minimum fixed cost is selected as the branching variable.

A difficulty in this estimation approach is that $\underline{\pi}_k$ and $\bar{\pi}_k$ are never explicitly computed. These variables, however, can be estimated from the available learning which takes place during program execution. There are two possibilities--both of which have been incorporated into the MEDOS code and may be activated at the option of the user.

1. One choice is to use the current values of $\bar{\pi}$ and $\underline{\pi}$, the products of the internal pricing mechanism. Stated formally, this method is

· Whenever the $\bar{H}^{th}$ feasible solution is found, compute

$$\sigma = \sum_{p\epsilon P} \sum_{i\epsilon N} \omega_{ip}(\bar{H}) b_{ip} + \sum_{k\epsilon D} (\underline{\pi}_k(\bar{H})\underline{V}_k - \bar{\pi}_k(\bar{H})\underline{V}_k) z_k$$

$$= CON(\bar{H}) + \sum_{k\epsilon D} Q_k(\bar{H}) z_k \qquad (5.25)$$

where

$$CON(\bar{H}) = \sum_{p\epsilon P} \sum_{i\epsilon N} \omega_{ip}(\bar{H}) b_{ip}$$

and

$$Q_k(\bar{H}) = \underline{\pi}_k(\bar{H})\underline{V}_k - \bar{\pi}_k(\bar{H})\underline{V}_k$$

Set SIGMAX ← Max {σ, SIGMAX}

· When branching

Let SFR ⊆ FR = {k' : $CON(\overline{H}) + \sum_{k \in DC(z)} Q_k(\overline{H})$

$+ Q_{k'}(\overline{H}) \leq SIGMAX\}$

Select $\overline{k}$: $f_{\overline{k}} = \min_{k' \in SFR} f_{k'}$

If SFR = ∅, activate the REG CAP and MIN CI rules.

2. The second approach uses $\overline{\rho}_{kp}$ and $\underline{\rho}_{kp}$ the multipliers corresponding to arc upper and lower bounds, to estimate $\overline{\rho}_k$ and $\underline{\rho}_k$. Stated formally this method is

· Whenever the $\overline{H}^{th}$ feasible solution is found

For each k∈DC(z):

$$\text{Set } \overline{\rho}_k = \begin{cases} 0 \text{ if } \sum_{p \in P} x_{kp}(\overline{H}) < \overline{V}_k \\ \max_{p \in P} \overline{\rho}_{kp}(\overline{H}) \quad \text{Otherwise} \end{cases} \tag{5.26}$$

$$\text{Set } \underline{\rho}_k = \begin{cases} 0 \text{ if } \sum_{p \in P} x_{kp}(\overline{H}) > \underline{V}_k \\ \min_{p \in P} \underline{\rho}_{kp}(\overline{H}) \quad \text{Otherwise} \end{cases} \tag{5.27}$$

and

$$\sigma = CON'(\overline{H}) + \sum_{k \in DC(z)} Q_k(\overline{H})$$

where $CON'(\overline{H})$ is derived from the expression for $CON(\overline{H})$ by substituting $\overline{\rho}_k$ for $\overline{\pi}_k(\overline{H})$ and $\underline{\rho}_k$ for $\underline{\pi}_k(\overline{H})$.

Set SIGMAX ← Max {σ, SIGMAX}

· The branching rule is the same as that elaborated under (1).

The second approach is a preferred method because the multipliers used come from a known primal feasible solution. Expressions (5.26) and (5.27) can be shown to be equivalent to the Lagrange multipliers in (LMCTP(z,π)) which yield the same solution as the simple bounded CTP (RMCTP(z,y)) which generated the ρ_{kp}'s used. This relationship is fully developed in Section 5.3.

5.2.3.7 Pos LB Rule. The Pos LB Rule selects a free DC which has non-zero throughput in the solution of LBC($\tilde{z},\pi$). This rule differs from the Max TP Rule in that the purpose is to eliminate from consideration those DC's with zero flow rather than to select that DC with maximum flow. The Pos LB Rule is therefore a useful rule to override higher priority rules.

5.2.3.8 GMS Rule. The GMS Rule was inspired by Akinc and Khumawala's Largest Omega Rule [1] for capacitated location problems. The underlying concept is to open that DC which offers the greatest potential for improving LB(z).

$$\text{Let } S_k = \upsilon(\text{LMCTP}(\hat{z},\pi)) - \text{LB}(z) \qquad k\epsilon FR$$

where

$$\hat{z}_j = \begin{cases} 1 : j\epsilon DC(z)\cup\{k\}, \\ 0 \quad \text{otherwise} \end{cases} \qquad k\epsilon FR$$

The GMS Rule selects k:

$$S_k = \max_{k\epsilon FR\cup RS(j)} S_i$$

where j denotes a customer requiring reverse star supply. If no such customer exists, Max S_i is computed over all $i\epsilon FR$.

The implementation of the GMS Rule bears a high cost. A lower bound must be computed for each configuration generated. Normally, this is not required--that is, if Step 3 of the algorithm eliminates z as a contender, Step 5, which would solve for a configuration lower bound, is bypassed. Additionally, under the GMS Rule, a lower bound must be computed for each candidate completion considered. An exorbitant number of CTP problems must be solved as a result.

5.2.3.9 Priority of Rules. Absolute priority is always given to a free DC which lies on the reverse star of a customer needing supply. Accordingly, the RS Rule is always activated first. If this rule unmasks several DC candidates, the Max UB, Max TP, and Reg Cap Rules are activated in that order to decide which of the candidates to open. If the Reg Cap Rule produces more than one candidate the Min CI rule is used. When the Min CI Rule is in effect the Pos LB Rule disqualifies any DC candidate which had zero throughput in the bounding problem solution. If there does not exist a customer requiring reverse star supply the branching rules are implemented in the following order: Max UB, Max TP, Reg Cap, Min CI, Pos LB.

Max Sigma and GMS are alternatives to the other rules.

5.3 Price and Resource Direction

The MEDOS algorithm employs price and resource direction as a means for simultaneously strengthening both lower and upper bounds. For a fixed configuration z and Lagrange multiplier vector π, the solution to problem LMCTP(z,π) yields a lower bound on MCTP(z). A sequence of such solutions, using a subgradient price adjustment mechanism,

are produced in the price-directive loop (Steps 9-10) of the algorithm. An upper bound on MCTP(z) (and on (MDML) as well) is obtained by solving RMCTP(z,y) with a vector of simple bounds obtained by allocating to each commodity, and for each open DC, a share of the total upper and lower throughput. A sequence of these solutions, using a subgradient resource adjustment scheme, are produced by the resource-directive loop (Steps 7-8) of the algorithm. ε-optimality can be assured if the greatest lower bound from price direction is within ε percentage of the least upper bound from resource direction. Such a test of optimality can be devised if a procedure exists for updating the lower bounds as improved upper bounds are generated. The following theorem provides the basis for using the optimal multipliers from RMCTP(z,y) as input to LMCTP(z,π) in order to update LB(z).

The following definitions and formulations are relevant.

Dual of Multicommodity Flow Problem (MCTP(z))

$$\text{(DMCTP(z))}\quad \text{Maximize} \quad \sum_p \omega_p b_p + \sum_k (\underline{\pi}_k \underline{V}_k - \overline{\pi}_k \overline{V}_k) - \sum_p \mu_p u_p \tag{5.27}$$

Subject to

$$\omega_{ip} - \omega_{jp} - \mu_{ijp} \leq c_{ijp} \qquad (i,j)\in A\text{-}D \tag{5.28}$$

$$\omega_{kp} - \omega_{k+1,p} - \overline{\pi}_k + \underline{\pi}_k \leq c_{kp} \qquad k\in DC(z),\ p\in P \tag{5.29}$$

$$\overline{\pi}_k,\ \underline{\pi}_k \geq 0 \qquad k\in DC(z) \tag{5.30}$$

$$\mu_{ijp} \geq 0 \qquad (i,j)\in A\text{-}D \tag{5.31}$$

Dual of Multicommodity Flow Restriction (RMCTP(z,y))

$$\text{(DRMCTP(z,y))}\quad \text{Maximize} \quad \sum_p \omega_p{}' b_p + \sum_p \sum_k (\underline{\rho}_{kp} \underline{y}_{kp} - \overline{\rho}_{kp} \overline{y}_{kp}) - \sum_p \mu_p{}' u_p \tag{5.32}$$

Subject to

$$\omega'_{ip} - \omega'_{jp} - \mu'_{ijp} \leq c_{ijp} \quad (i,j)\in A\text{-}D \qquad (5.33)$$

$$\omega'_{kp} - \omega'_{k+1,p} - \overline{\rho}_{kp} + \underline{\rho}_{kp} \leq c_{kp} \quad k\in DC(z),\ p\in P \qquad (5.34)$$

$$\overline{\rho}_{kp},\ \underline{\rho}_{kp} \geq 0 \quad k\in DC(z),\ p\in P \qquad (5.35)$$

$$\mu'_{ijp} \geq 0 \qquad (5.36)$$

Optimality Conditions for MCTP(z)

Primal Feasibility:

PF: (5.2), (5.4), (5.7), and (5.8) are satisfied.

Dual Feasibility:

DF1: (5.28) is satisfied.

DF2: (5.29) is satisfied.

DF3: (5.30) is satisfied.

Complementary Slackness:

CS1: $\sum_{p\in P} x_{kp} < \overline{V}_k$, $\overline{\pi}_k = 0$ $\quad k\in DC(z)$

CS2: $\sum_{p\in P} x_{kp} > \underline{V}_k$, $\underline{\pi}_k = 0$ $\quad k\in DC(z)$

CS3: $x_{ijp} > 0,\ \omega_{ip} - \omega_{jp} - c_{ijp} = \mu_{ijp}$ $\quad (i,j)\in A\text{-}D,\ p\in P$

CS4: $x_{kp} > 0,\ \omega_{ip} - \omega_{jp} - \overline{\pi}_k + \underline{\pi}_k = c_{kp}$ $\quad k\in DC(z),\ p\in P$

Let x', ω', μ' be the optimal solution to RMCTP(z,y)

Set $x = x'$, $\omega = \omega'$, $\mu = \mu'$, and

$$\overline{\pi}_k = \left\{ \begin{array}{l} 0 \text{ if } \sum_{p\in P} x_{ip} < \overline{V}_k \qquad (5.37) \\ \max_{p\in P} (\omega_{kp} - \omega_{k+1,p} - c_{kp}) \text{ otherwise} \qquad (5.38) \end{array} \right\} k\in DC(z)$$

$$\underline{\pi}_k = \left\{ \begin{array}{l} 0 \text{ if } \sum_{p\in P} x_{kp} > \underline{V}_k \qquad (5.39) \\ \min_{p\in P} (c_{kp} - \omega_{kp} + \omega_{k+1,p}) \text{ otherwise} \qquad (5.40) \end{array} \right\} k\in DC(z)$$

Theorem: (a) If x is optimal in MCTP(z), ω, μ, and π are optimal multipliers. (b) If x is suboptimal in MCTP(z), all the optimality conditions hold except DF2 and CS4 for some $x_{kp} = \overline{y}_{kp}$.

Proof: Primal feasibility, dual feasibility, and complementary slackness will be tested in order.

Primal feasibility:

1. Clearly, since x is primal feasible in RMCTP(z,y) it must also be primal feasible in MCTP(z), whether optimal or not.

Dual feasibility:

2. DF1 holds whether x is optimal or not. Consider three cases.

   a. $x_{ijp} = 0$ and nonbasic.

   b. $0 \leq x_{ijp} \leq u_{ijp}$ and basic.

   c. $x_{ijp} = u_{ijp}$ and nonbasic.

   If case 2a applies, $\omega'_{ip} - \omega'_{jp} - c_{ijp} \leq 0$. Since $\mu_{ijp} = 0$, DF1 holds.

   If case 2b applies, $\omega'_{ip} - \omega'_{jp} - c_{ijp} = 0$. Since (5.36) must be satisfied and thence (5.31), DF1 holds.

   In case 2c, $\omega'_{ip} - \omega'_{jp} - c_{ijp} \geq 0$. Since (5.28) and (5.31) imply $\mu_{ijp} \geq \omega_{ip} - \omega_{jp} - c_{ijp}$, DF1 follows.

3. DF3 holds whether x is optimal or not. Consider three cases.

   a. $\underline{V}_k < \sum_{p \in P} x_{kp} < \overline{V}_k$

   b. $\sum_{p \in P} x_{kp} = \overline{V}_k$

   c. $\sum_{p \in P} x_{kp} = \underline{V}_k$

If case 3a applies, $\overline{\pi}_k, \underline{\pi}_k = 0$ from (5.37) and (5.39) respectively. Hence, DF3 holds.

If case 3b applies, $x_{kp} = \overline{y}_{kp}$ $p \varepsilon P$ which implies that $\omega'_{kp} - \omega'_{k+1,p} - c_{kp} \geq 0$ $p \varepsilon P$. From (5.28) DF3 follows.

If case 3c applies, $x_{kp} = \underline{y}_{kp}$ $p \varepsilon P$ which implies that $\omega'_{kp} - \omega_{k+1,p} - c_{kp} \leq 0$ $p \varepsilon P$ or $c_{kp} - \omega'_{kp} + \omega'_{k+1,p} \geq 0$ $p \varepsilon P$. From (5.40) DF3 follows.

4. To analyze DF2, consider four cases.

   a. $\sum_{p \varepsilon P} x_{kp} = \overline{V}_k$

   b. $\sum_{p \varepsilon P} x_{kp} = \underline{V}_k$

   c. $\underline{V}_k < \sum_{p \varepsilon P} x_{kp} < \overline{V}_k$ and x is not optimal in MCTP(z)

   d. $\underline{V}_k < \sum_{p \varepsilon P} x_{kp} < \overline{V}_k$ and x is optimal in MCTP(z)

   If case 4a applies, $\underline{\pi}_k = 0$ by (5.39). DF2 is equivalent to $\overline{\pi}_k \geq \omega_{kp} - \omega_{k+1,p} - c_{kp}$ which is guaranteed by (5.38).

   Similarly, if case 4b applies, $\overline{\pi}_k = 0$ by (5.37). DF2 then is equivalent to $\underline{\pi}_k \leq c_{kp} - \omega_{kp} + \omega_{k+1,p}$ which is guaranteed by (5.40).

   If case 4c applies, there must exist an $x_{kp} = \overline{y}_{kp}$ such that $\omega_{kp} - \omega_{k+1,p} - c_{kp} > 0$. Since $\overline{\pi}_k = 0$ by (5.37) and $\underline{\pi}_k = 0$ by (5.39) DF2 does not hold for this kp. However, DF2 holds for all other kp whether x is optimal or not as shown in the discussion of case 4d.

   If case 4d applies, one of the following must be true:

(i) $x_{kp} = \underline{y}_{kp}$, nonbasic and $\omega_{kp} - \omega_{k+1,p} - c_{kp} < 0$ in which case DF2 holds with positive slack.

(ii) $\omega_{kp} - \omega_{k+1,p} - c_{kp} = 0$, in which case DF2 holds with zero slack.

Complementary Slackness:

5. CS1 and CS2 obviously hold whether x is optimal or not.

6. CS3 also holds independent of the optimality of x. If $x'_{ijp} > 0$, $\mu'_{ijp} = \omega'_{ip} - \omega'_{jp} - c_{ijp}$. Hence CS3 holds.

7. In analyzing CS4, three cases are relevant.

   a. $\underline{V}_k < \sum_{p \in P} x_{kp} < \overline{V}_k$

   b. $\sum_{p \in P} x_{kp} = \overline{V}_k$

   c. $\sum_{p \in P} x_{kp} = \underline{V}_k$

   If case 7a applies, one of the following holds:

   (i) $\omega_{kp} - \omega_{k+1,p} - c_{kp} = 0$, in which case CS4 follows immediately from (5.37) and (5.39).

   (ii) $\omega_{kp} - \omega_{k+1,p} - c_{kp} > 0$, in which case CS4 is violated due to (5.37) and (5.39).

   (iii) $\omega_{kp} - \omega_{k+1,p} - c_{kp} < 0$, in which case CS4 is violated due to (5.37) and (5.39).

   Case 7a(ii) can only occur if x is not optimal in MCTP(z), as was shown in 4c and d above. If x is optimal, case 7a(iii) implies $\underline{\pi}_k > 0$ which can only be true if $\sum_{p \in P} x_{kp} = \underline{V}_k$, a contradiction to 7a.

If case 7b applies one of the following holds.

(i) $\omega_{kp} - \omega_{k+1,p} - c_{kp} = \bar{\pi}_k$, in which case CS4 holds.

(ii) $\omega_{kp} - \omega_{k+1,p} - c_{kp} < \bar{\pi}_k$, in which case CS4 is violated due to (5.39).

Case 7b(ii) cannot occur if x is optimal since optimality conditions require $\omega_{kp} - \omega_{k+1,p} - c_{kp} = 0$ for all p.

If case 7c applies, one of the following holds:

(i) $c_{kp} - \omega_{kp} + \omega_{k+1,p} = \underline{\pi}_k$, in which case CS4 holds.

(ii) $c_{kp} - \omega_{kp} + \omega_{k+1,p} > \underline{\pi}_k$, in which case CS4 is violated.

Case 7c(ii) cannot occur if x is optimal for the same reason as stated under 7b(ii).

This concludes the proof.

5.4 Model Enhancements

5.4.1 Regionalization

The regionalization of customers and warehouses is consistent with the view many firms have of their respective distribution and marketing operations. The disection of the distribution network into smaller separate and distinct entities (regions) is a common practice, and this principle is often firmly engrained in the design of accounting systems, marketing strategies, distribution policies, and management accountabilities. It is quite natural, then, that a distribution planning model take recognition of a regional framework and exploit any advantages of such a structure. This has been a design ambition in the present work, which uses a dynamic regional partition to accomplish two goals:

1. To aid in establishing a low cost initial configuration.

2. To provide a criterion for selecting relatively attractive DC's to open when branching.

Section 5.5 describes in detail how the heuristic utilizes the input regional partition to establish an initial configuration. In most cases, the input partition will be described in terms of geographical boundaries; however, other criteria are possible. For example, a "commodity clustering" criterion might regionalize according to commodity groupings, or "bundles," and each customer assigned to the region corresponding to the bundle of maximum demand; and each DC designated to carry a certain range of commodity bundles. Another possibility is a stratification based on customer type (e.g., wholesale or retail), or based on size. The Australian ice cream distributor case provides a good example of a firm which has a diversification of customer types and sizes. Such a heterogeneous customer population requires flexible and adaptive management skills to maintain acceptable service levels overall. Some form of organizational regionalization can, in certain cases, help facilitate the accomplishment of this objective.

After input, the regional partition is subject to revision whenever a new incumbent is generated. At this stage it may cease to even remotely resemble the initial partition. This is no cause for alarm, as the re-partition is internal to the algorithm execution and in no way impacts existing organizational structures. New regionalizations can be made available on computer output but this is not a normal facility provided.

Revised regionalizations are based solely on flow information. If a new incumbent solution reveals that a particular DC is shipping a greater volume of goods into any region other than the region to which that DC has been assigned, the regional assignment will be changed to coincide with the maximum outflow. Similarly, if a customer is found to be receiving more shipments from any region other than the one to which it is assigned, the assignment will be changed to coincide with the maximum inflow. Stated formally,

Re-partition regions, if necessary, so that

$$\sum_{p\epsilon P} \sum_{j\epsilon R_\ell} x_{ijp}(h) \;/\; \sum_{p\epsilon P} \sum_{j\epsilon N} x_{ijp}(h) \qquad (5.41)$$

is maximized over all $i\epsilon R_\ell \cap D$, and

$$\sum_{p\epsilon P} \sum_{j\epsilon I^- \cap R_\ell} x_{jip}(h) \;/\; \sum_{p\epsilon P} D_{ip} \qquad (5.42)$$

is maximized, over all $i\epsilon R_\ell \cap C$

DC's are checked first, and any reassignments are made, before customer assignments are checked.

Regionalization plays an important role in branching decisions through the implementation of the Reg Cap Rule.

5.4.2 Repeat Solutions

If a solution begins repeating (that is, the subgradient adjustment procedure has been incapable of producing a vector which has altered the solution) and continues to repeat for a number of successive iterations, the algorithm will exit the loop. This condition is highly unlikely to occur, but the facility for detecting it is built into the software, and applies to both price-directive and resource-directive

loops. The limit on the number of successive repeats is the input variable MXCNT.

5.4.3 Penalty Arcs

The projection operations in steps 6 and 8 of the algorithm do not guarantee a resource allocation which will be CTP-feasible. Kennington and Shalaby [85] overcome the problem through dual pivots to restore feasibility. The MEDOS algorithm insures all solutions are CTP-feasible by placing artificial penalty-bearing arcs between each plant and each customer. A matrix generator internal to the software appends these arcs to the input data structure. The test for CTP-feasibility, then, is a test of non-zero flow on any of the penalty arcs. In price-direction a detection of this condition leads to a branch or fathom. In resource direction this condition requires an adjustment in the resource vector.

5.5 Description of the Heuristic Algorithm

Additional Problems:

Multicommodity Aggregated Transportation Problem

$$\text{(MATP(z))} \quad \text{Minimize} \sum_{p\epsilon P} \sum_{i\epsilon F} \sum_{R_\ell \epsilon R} \bar{c}_{iR_\ell p}\, \bar{x}_{iR_\ell p} \tag{5.43}$$

Subject to

$$\sum_{i\epsilon F} \bar{x}_{iR_\ell p} = \bar{D}_{R_\ell p} \qquad p\epsilon P,\ R_\ell \epsilon R \tag{5.44}$$

$$\underline{S}_{ip} \le \sum_{R_\ell \epsilon R} \bar{x}_{iR_\ell p} \le \bar{S}_{ip} \qquad p\epsilon P,\ i\epsilon F \tag{5.45}$$

$$\bar{x}_{iR_\ell p} \ge 0 \qquad R_\ell \epsilon R,\ i\epsilon F,\ p\epsilon P \tag{5.46}$$

Commodity-Specific Aggregated Transportation Problem

(MATP(z)) decomposes on commodity to the following commodity-independent formulation:

$$(MATP_p(z)) \quad \text{Minimize} \quad \sum_{i\in F} \sum_{R_\ell \in R} \bar{c}_{iR_\ell p}\, \bar{x}_{iR_\ell p} \tag{5.47}$$

Subject to

$$\sum_{i\in F} \bar{x}_{iR_\ell p} = \bar{D}_{R_\ell p} \qquad R_\ell \in R \tag{5.48}$$

$$\underline{S}_{ip} \leq \sum_{R_\ell \in R} \bar{x}_{iR_\ell p} \leq \bar{S}_{ip} \qquad i\in F \tag{5.49}$$

$$\bar{x}_{iR_\ell p} \geq 0 \qquad R_\ell \in R,\ i\in F \tag{5.50}$$

where

$$\bar{c}_{iR_\ell p} = \sum_{j\in R_\ell \cap C} \hat{\mu}_{ijR_\ell p}\, \hat{c}_{ijp} \qquad i\in F,\ R_\ell \in R,\ p\in P \tag{5.51}$$

and

$$\hat{\mu}_{ijR_\ell p} = D_{jp}/\bar{D}_{R_\ell p} \qquad i\in F,\ j\in C\cap R_\ell,\ p\in P,\ R_\ell \in R \tag{5.52}$$

$\hat{c}_{ijp}$ = length of shortest path from $i\in F$ to $j\in C$.

Additional Notation:

R - set of regions, $R_\ell \in R$ denotes the ℓ^{th} region

S_{stp} - set of arcs or shortest path from source s to sink t for commodity p

$\bar{x}(z)$ - solution to MATP(z)

$\hat{x}(z)$ - disaggregate solution to $MATP_p(z)$, where

$$\hat{x}_{ijp} = \hat{\mu}_{ijR_\ell p}\, \bar{x}_{iR_\ell p} \qquad i\in F,\ j\in R_\ell \cap C \tag{5.53}$$

$\hat{x}^*(z)$ - consolidated solution to MATP(z)

$$\hat{x}^*_{ijp} = \sum_{(s,t)\in S_{stp}} \hat{x}_{stp} \qquad p\in P,\ (i,j)\in A,\ s\in F,\ t\in C \tag{5.54}$$

$$\upsilon^*(\text{MATP}(z)) - \sum_{p \in P} \sum_{(i,j) \in A} c_{ijp}\, x^*_{ijp} \qquad (5.55)$$

5.5.1 Heuristic Algorithm (Refer to Figure 5.2 and Section 5.2.2, Step Details)

Step 1H Initialization

Initialize the regionalization of candidate DC sites and customer zones

Set $DC(z) \leftarrow \emptyset$

$UB \leftarrow \infty$

Initialize π

Step 2H Compute the Aggregate Demand for Each Region

Step 3H Compute the Cost Attractiveness Indices CI_k for Each Candidate DC

Step 4H Establish an Initial Configuration $\hat{z}$ Based on Supply and Demand Considerations

Set $DC(z) \leftarrow \{k : \hat{z}_k = 1\}$

$LB(z) \leftarrow 0$

Step 5H Solve Multicommodity Aggregated Transportation Problem

Solve MATP(z) for $\hat{x}^*(z)$ and $\upsilon^*(\text{MATP}(z))$

If $\hat{x}^*(z)$ is TP-feasible

Set $UB \leftarrow \text{Min}\ \{UB, \upsilon^*(\text{MATP}(z))\}$

Step 6H Compute Lower Bound on Configuration

Solve $\text{LMCTP}(z,\pi)$ for $x^*(z,\pi)$

Set $LB(z) \leftarrow \text{Max}\ \{LB(z), \upsilon^*(\text{LMCTP}(z,\pi))\}$

If $x^*(z,\pi)$ TP-feasible

Set $UB \leftarrow \text{Min}\ \{UB, \upsilon^*(\text{LMCTP})\}$

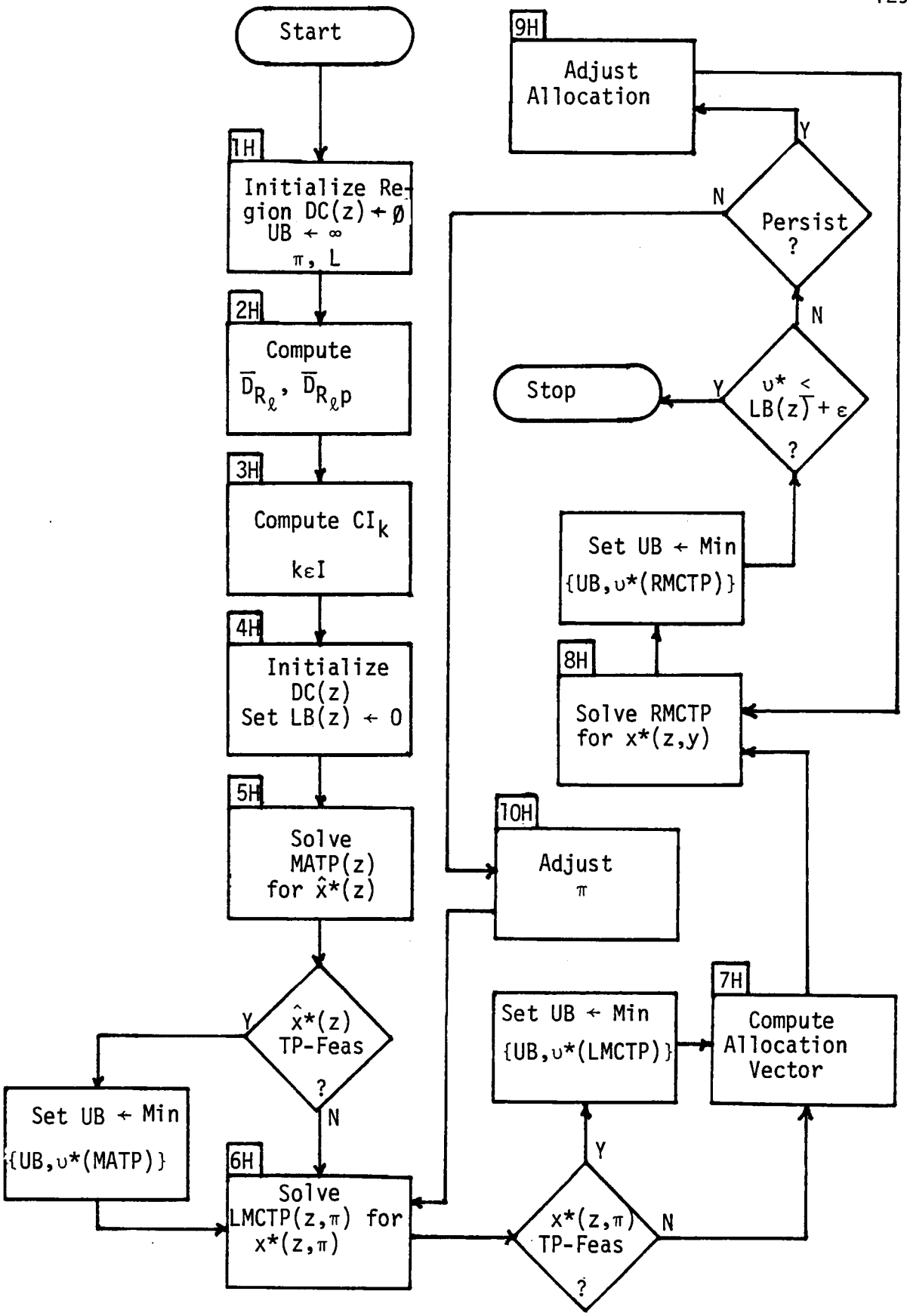

Figure 5.2. Flow Chart for Heuristic

Step 7H Compute an Allocation Vector y by Projecting $\hat{x}^*(z)$ onto y-space

Set $y \leftarrow P[\hat{x}^*(z)]$

$i \leftarrow 0$

Step 8H Solve Multicommodity Flow Problem

Set $i \leftarrow i+1$

Solve RMCTP(z,y) for $x^*(z,y)$

Set UB $\leftarrow$ Min {UB, υ^*(RMCTP(z,y))}

If υ^*(RMCTP(z,y)) $\leq$ LB(z) + ε, EXIT

If i=L, go to Step 10H

Step 9H Adjust Resource Allocation Vector

Go to Step 8H

Step 10H Adjust Price Vector

Go to Step 6H

5.5.2 Step Details

Step 2H Compute $\overline{D}_{R_\ell p} = \sum_{i \in C \cap R_\ell} D_{ip}$ $\quad R_\ell \in R$ (5.56)

$$\overline{D}_{R_\ell} = \sum_{p \in P} \overline{D}_{R_\ell p} \qquad R_\ell \in R \qquad (5.57)$$

Step 3H Use formula (5.24) described in section 5.2.3.5.

Step 4H Open DC's, on a regional basis, so that a convex combination of upper and lower open DC throughput capacities just cover aggregate regional demands. Two criteria are used for selecting which DC's to open. If the aggregate demand not covered is greater than or equal to the upper throughput capacity of the DC candidate

belonging to the region and possessing the lowest CI, that DC is opened, and a convex combination of its upper and lower throughput bounds is deducted from the regional aggregate demand. Otherwise, the DC which can service the demand at the least per unit cost is selected. Stated formally,

Let $K_{R_\ell} = \sum_{k \in DC(z) \cap R_\ell} [\alpha \overline{V}_k + (1 - \alpha) \underline{V}_k]$ $\quad \alpha \in [0,1]$

$B(k) = (f_k + v_k \overline{V}_k) / (\overline{D}_{R_\ell} - K_{R_\ell})$ $\quad k \in D \cap R_\ell$

Select $k' \in FR \cap R_\ell : CI_{k'} = \underset{k \in FR \cap R_\ell}{\text{Min}} CI_k$ and $K_{R_\ell} < \overline{D}_{R_\ell}$ $\quad R_\ell \in R$

If $\overline{D}_{R_\ell} - K_{R_\ell} \geq \overline{V}_{k'}$,

Set $z_{k'} \leftarrow 1$, $DC(z) \leftarrow DC(z) + \{k'\}$

Otherwise,

Select $\tilde{k} \in FR \cap R_\ell : B(\tilde{k}) = \underset{k \in FR \cap R_\ell}{\text{Min}} B(k)$ and $K_{R_\ell} < \overline{D}_{R_\ell}$ $\quad R_\ell \in R$

and set $z_{\tilde{k}} \leftarrow 1$, $DC(z) \leftarrow DC(z) + \{\tilde{k}\}$

Rationale: A feasible solution must contain adequate network capacity to satisfy demand requirements. Demands are more likely to be satisfied at least cost if DC's supply customers within the same region. The tuning parameter alpha provides flexibility to adjust the configuration to a desired average operating level. A high alpha will reduce the number of DC's opened; a low alpha will increase them.

Step 5H a. Compute aggregate weights $\hat{\mu}_{ijR_\ell p}$ $\quad i \in F, j \in C \cap R_\ell$, $p \in P, R_\ell \in R$

b. Compute and store shortest paths S_{ijp} and shortest path lengths $\hat{c}_{ijp}$, $\quad i \in F, j \in C, p \in P$

c. Compute aggregate cost $\overline{c}_{iR_\ell p}$

d. Solve MATP(z) for $\bar{x}(z)$

e. Disaggregate the solution of $MATP_p(z)$ to obtain $\hat{x}_p$ using (5.53)

f. Compute $\hat{x}^*(z)$ from (5.54)

Step 7H The projection operation is described in detail in Section 4.2.1.2.

Step 8H After solving the Multicommodity Flow Problem, the reverse star for each customer is checked. There is no guarantee that Step 4H has produced a z which will be CTP-feasible. CTP-feasibility is forced through the introduction of artificial penalty-burdened arcs connecting each customer directly to each of the plants. If such transfers are allowed (for example, through cross-docking policies) these "Big M" penalty arcs are omitted from such legitimate transportation links. The Step 8H solution must be tested for non-zero flow on penalty arcs. This test checks the reverse star of each customer. When penalty flow is encountered, the DC nearest to the customer receiving the penalty flow is opened, and the algorithm branches back through Steps 6H-8H. Stated formally,

If $x_{ijp} > 0$ for any $i \in F$, $j \in C$: $c_{ij} = PENALTY$

Set $z_{k'} \leftarrow 1$: $c_{k'+1,j} = \underset{k \in FR}{Min}\ c_{k+1,j}$

Set $\hat{x}^*(z) \leftarrow x^*(z,y)$, $DC(z) \leftarrow DC(z) + \{k'\}$

Go to Step 6H.

Step 9H Use subgradient methods as described in Section 4.2.1.2.

Step 10H Use procedure outlined in Step 9 of the MEDOS algorithm.

CHAPTER VI

MEDOS SOFTWARE

6.1 Design Principles

The design procedure for the MEDOS software package followed closely the principles as outlined by Bradley, Brown, and Graves [10]:

1. The code is capable of handling large scale problems. For example, the problem reported in Chapter I and solved using the MEDOS code, consisted of 1270 rows and 7690 columns, not including logical arcs.

2. The code solves general problems. MEDOS has the capability of solving networks which are highly or loosely capacitated. Multiple arcs are accommodated and there is no requirement that the costs be the same on each arc connecting the same pair of nodes. With respect to node numbering, the only requirement is that source nodes be numbered consecutively, then transshipment nodes, and finally sink nodes; and that transshipment (DC) node numbering be all odd or all even numbers to allow for the insertion in numerical sequence of outbound nodes by the program. Arc numbers are entered as permissible transportation links between network entities. Numbering must be consistent with the node-numbering regimen but there is no need to input the actual arc numbers as the program computes these. For example, if a permissible path from plant 2 to customer 79 is via DC 8, the links which are input are 2,8 and 8,79. The software converts these links to the actual chain consisting of arcs 2,8; 8,9; and 9,79. There is no restriction

on the order in which transportation links must be introduced on input. No initial feasible solution is necessary. Modularity is an inherent design feature. The code consists of a main calling program and five support modules embracing a total of 19 subroutines. MEDOS is therefore flexible and can easily be modified to meet any specialized requirements.

3. The code operates on the principle that node information is more economical to store than arc information, since it is assumed that, in general, the number of arcs will be much greater than the number of nodes.

4. MEDOS is machine-independent. The code was written in FORTRAN IV and makes use of no nonstandard language features or machine specific hardware features, with the exception of the algorithm timing mechanism, which of necessity must be machine specific. However, a very minor modification adapts the timing feature to a particular machine configuration, or the timing facility can be easily de-activated.

5. Speed of execution, insofar as possible, has been given preference to economy of data storage.

6. The program is equipped with a number of external tuning parameters which can be utilized to facilitate performance across a variety of network structures.

6.2 Data Structures

Storage requirements fall into five broad categories: node information, arc information, solution information, enumerative data, and heuristic-specific requirements.

6.2.1 Node Information

Network nodes represent plants, DC's, or customers. Plant descriptions are stored in the array PDES and upper and lower plant capacities are stored in the plant-by-product-length arrays SUPHI and SUPLO, respectively. DC descriptions are stored in the array DCDES, upper and lower throughput bounds in TPHI and TPLO respectively, the fixed establishment cost in FC, and the linear unit DC operating cost in VCC. The logical array SUBECH flags those DC's which have no direct link to any plant. Customer descriptions are stored in CDES and demand data in the product-by-customer-length array DEMAND.

The representation of the regionalization of customers and DC's, as described in Chapter V, makes use of linked list structures of the type described by Hughes, Pfleeger, and Rose [77]. DC candidates are first sorted in order of increasing cost attractiveness index. The region-length array COSTHD points to the most attractive DC in each region. The DC-length array DCLINK contains a sequence of pointers which defines the list, in sorted order, of DC's within each region. The region-length array CUSTHD contains pointers to the first customer assigned to each region. The customer-length array CUSLNK contains the sequence of pointers to complete the customer list by region.

6.2.2 Arc Information

The representation of arc information makes use of a data architecture known as the "reverse star" structure. Following input and DC node-splitting, the arcs are sorted by successor, or head, node. All arcs with the same successor are stored in contiguous space. An

arc-length array HEAD is established initially to aid in arc sorting. This array is not saved after input.

The node-length array HD contains pointers to the first arc which has each node as its successor. That is, element HD(•) contains the number of the first arc (i,j) with j = (•). The reverse star structure enables a complete enumeration of all arcs in the network by iterating the node-length array HD. The arc-length array T contains the predecessor, or tail, node for each arc. The reverse star of any node, k, is obtained by examining T for each arc between HD(k) and HD(k+1).

The arc-length arrays CPTEMP and CPERM contain permanent arc capacities and costs respectively. Because costs and capacities may temporarily be changed by the algorithm's pricing and resource mechanisms respectively, there is a need for storing this temporary capacity and cost information in the additional arrays CP and C respectively. Permanent initial lower bounds for all arcs are zero. The resource scheme necessitates non-zero lower bounds be established for the DC arcs and these lower bounds are stored in the array LOWBND.

The pricing mechanism of MEDOS makes use of the enumeration history in establishing initial prices. Each time the algorithm enters the pricing loop initial prices are set equal to the permanent cost vector plus the most recent vector of Lagrange multipliers generated. That is, initial prices for the pricing loop h are the exiting prices for the pricing loop h-1. This price vector is stored in the array CDUAL.

6.2.3 Solution Information

The solution to the current subproblem is contained in the arc-length array XSTAR, the value of the incumbent solution of flow variables in XINCUM, and the variable VINCUM contains the value of the objective function corresponding to the incumbent solution. The incumbent configuration is obtained via a linked list structure. The variable ZFIRST points to the first DC in the configuration and the DC-length array ZLINK contains successive pointers to complete the list. The DC-length array TP contains the throughput for each DC corresponding to any solution. The arrays RHOHI and RHOLO are DC-arc-length arrays containing the upper and lower throughput capacity dual variables (arc numbers) for any commodity-independent capacitated transshipment problem solution.

6.2.4 Enumeration Data

The enumeration tree is stored in the array LINK. The variable NODEND contains a pointer to the node number corresponding to the last live node. The logical array FATH keeps track of fathomed nodes. The array FRELNK contains free variables sorted by cost attractiveness index. The variable HDFREE points to the head of the list. Termination of the algorithm occurs when all variables in LINK have been flagged as having been fathomed.

6.2.5 Heuristic-Specific Requirements

The subroutines AGTRAN and SPATH require five major storage arrays. The shortest paths are stored in array PRED and the shortest path lengths in array CHAT. The array CBAR contains the aggregate costs

between each plant and each region and AGWT is a customer-by-product length array containing the aggregate weights for the aggregated transportation problem (ATP). The solution to the ATP is stored in the array XHAT.

6.3 Subroutine Structure

Figure 6.1 illustrates the modular structure of MEDOS. A main program drives the algorithm, activating five support modules as needed. Each support module consists of a number of subroutine programs to perform specialized and repetitive tasks. A listing and brief description of each subroutine follows.

MAIN PROGRAM:

Function: To provide overall control of the algorithm, including input, output, timing, and termination; the administration of fathoming and branching rules; the execution and termination of price and resource directive loops; and incumbent updates.

SUPPORT MODULE: Data File Creation and Maintenance

Function: To process input data, set up data structures, initialize program variables and constants, set up an initial configuration, and modify data structures as required.

Subroutines: PREPAR, LNKCUS, LINKDC, REGDEM, ARCS, REGION

1. Subroutine PREPAR

   Function: To process input data, set up data structures, initialize program variables and constants, and set up an initial configuration. This

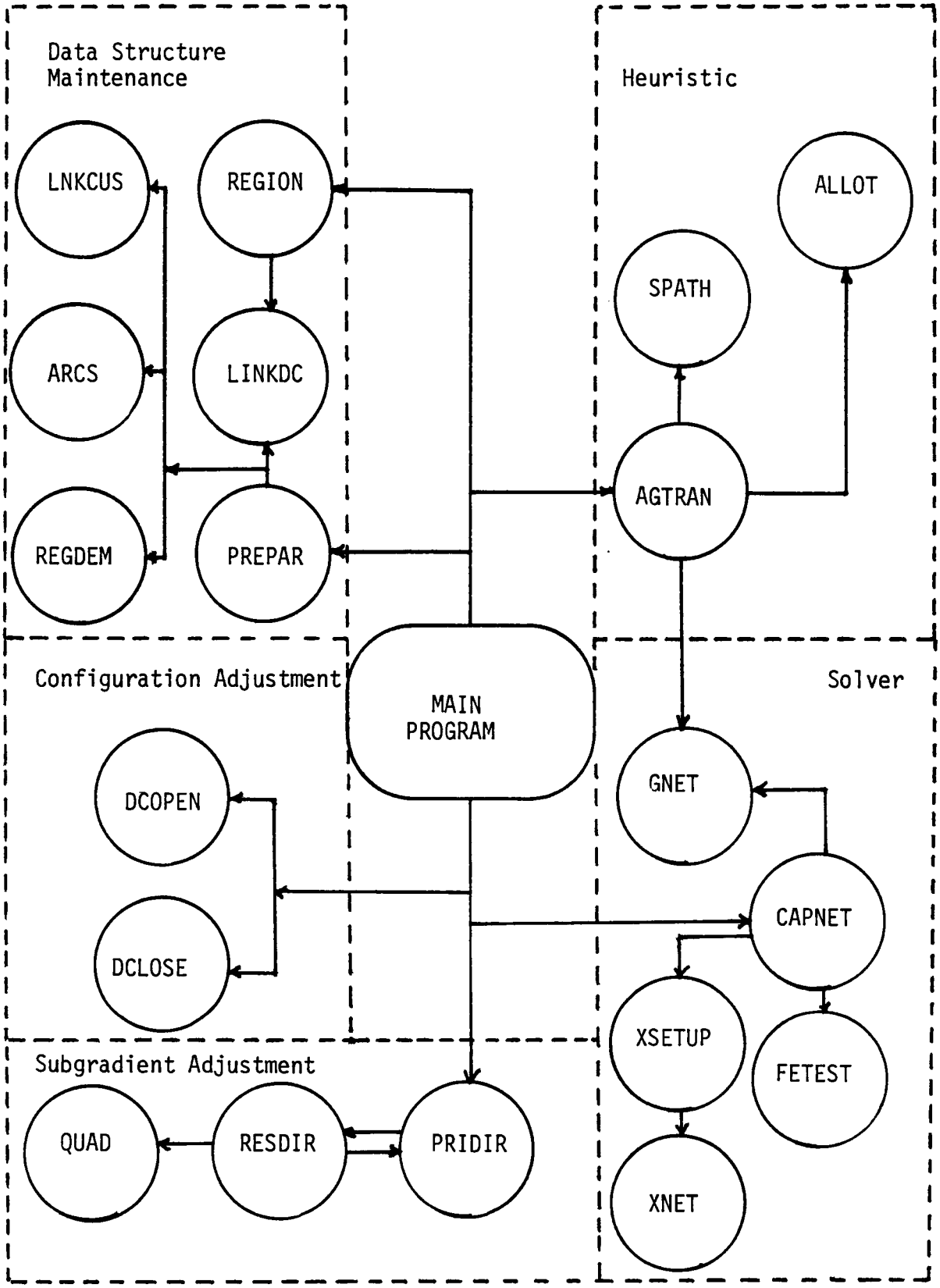

Figure 6.1 MEDOS Program Structure

subroutine is the main program for the support module and drives the subroutines LNKCUS, LINKDC, REGDEM, and ARCS.

2. Subroutine LNKCUS

   Function: To set up a linked list structure of customers by region.

3. Subroutine LINKDC

   Function: To set up a linked list structure of DC candidates by region and in ascending order of cost attractiveness index.

4. Subroutine REGDEM

   Function: To compute the total aggregate regional demand and the aggregate regional demand by product.

5. Subroutine ARCS

   Function: To establish the network arc structure from input data, to set up a reverse star data structure with the arcs sorted by successor, and to write this structure to a disk file if required.

   Remark: MEDOS includes an option to bypass the arc input data if a sorted file of the network arcs exists on disk. When this option is exercised the subroutine ARCS is not called.

6. Subroutine REGION

   Function: To check the regional partition whenever a new incumbent solution is found and adjust the partition if necessary.

SUPPORT MODULE: Heuristic

Function: To adjust the initial configuration set up by subroutine PREPAR if necessary and to compute an initial feasible solution.

Subroutines: AGTRAN, SPATH, ALLOT

7. Subroutine AGTRAN

   Function: To adjust the initial configuration if necessary and to compute an initial feasible solution. This is accomplished by setting up and solving an aggregated transportation problem on the initial configuration, projecting the resulting flows onto the solution space, and opening additional DC's to cover all customer reverse stars. This program controls the support module and drives the subroutines SPATH and ALLOT.

8. Subroutine SPATH

   Function: To compute the shortest paths between sources and sinks required for the aggregate transportation problem.

   Remark: The subroutine SPATH is a variation of a shortest path code used by the Computer Concepts Corporation [117] and is based on the label-correcting scheme proposed by Pape [18,116].

9. Subroutine ALLOT

   Function: To allocate capacity to arcs on the basis of a feasible solution to the aggregated transportation

problem. This allocation is then used in a capacitated transshipment formulation in an attempt to improve the solution.

SUPPORT MODULE: Optimizer

Function: To set up and solve commodity-independent subproblems, and to merge the subproblem solutions into a multicommodity solution.

Subroutines: CAPNET, FETEST, GNET, XSETUP, XNET

Remark: The Optimizer support module utilizes variants of Bradley, Brown, and Graves' GNET and XNET [10]. The choice between the two subroutines is at the option of the user and is specified on input. The relative merits guiding this choice are discussed in Section 6.5.

10. Subroutine CAPNET

    Function: To break the multicommodity network down into single commodity networks, to set up the input for the solver routines (either GNET or XNET), and if the GNET advanced start option is activated, to initially load the candidate queue.

11. Subroutine FETEST

    Function: To check any solution for throughput feasibility.

12. Subroutine GNET

    Function: To obtain an optimal solution to the commodity-independent capacitated transshipment problem or to determine no feasible solution

exists. If the XNET option is activated this subroutine is only called on to solve the aggregated transportation problem.

13. Subroutine XSETUP

    Function: To call subroutine XNET and, if the advance start option has been activated, to initially load the candidate queue. If the GNET option is activated this subroutine is not called.

14. Subroutine XNET

    Function: To obtain an optimal solution to the commodity-independent capacitated transshipment problem or to determine no feasible solution exists. If the GNET option is activated this subroutine is not called.

SUPPORT MODULE: Subgradient Adjustment

Function: To adjust capacities or costs as required for resource direction and price direction.

Subroutines: RESDIR, QUAD, PRIDIR

15. Subroutine RESDIR

    Function: To compute an improving allocation of resources to all the arcs sharing a common pair of nodes.

16. Subroutine QUAD

    Function: To solve the quadratic programming problem required for the subgradiant projection operation.

17. Subroutine PRIDIR

Function: To compute improving Lagrangean multipliers to achieve tighter lower bounds.

SUPPORT MODULE: Configuration Adjustment

Function: To open or close DC's as required when the configuration is changed.

Subroutines: DCOPEN, DCLOSE

18. Subroutine DCOPEN

Function: To open DC's as required and to recompute the total fixed charges.

19. Subroutine DCLOSE

Function: To close DC's as required and to recompute the total fixed charges.

6.4 Input/Output Specifications, File Structures, and Core Requirements

A significant amount of input time can be consumed by the subroutine ARCS depending on the number of arcs in the network; for example, over 2 CPU minutes on a DEC-10 computer was required to set up and sort the 7260 arcs for the Australian case study network described in Chapter I. If it is envisaged that multiple runs of the model will be made, for example for sensitivity analysis, parametric analysis, trade-off analysis, etc., it would seem prudent to perform the costly arc sort only the first time. MEDOS provides this facility. Raw transportation link data is only processed the first time the model is run, unless links are added or deleted. Since this data will not be part of input on subsequent runs, separate file structures must be maintained for

transportation links, and other program inputs. A third file structure is maintained for the sorted arc structure which is a component of input after the first computer run. If desired, storage economies can be achieved by replacing the original transportation link data file with the arc structure file. This action is not recommended, however, because additions and deletions of permissible transportation links are more difficult without the original data file. A possible compromise is to transfer the transportation link data to a permanent storage medium, such as tape. A fourth file structure contains all of the necessary information to start-up the algorithm at an advanced point in the enumeration tree. Specifically, if the algorithm must be terminated prematurely, all of the information necessary for the algorithm to restart precisely where the last run terminated is stored on a file structure. Such a facility makes a lengthy tree search, if necessary, more plausible.

6.4.1 Input Specifications

Transportation Link File. A header card, or record, contains the "from" and "to" numbers uniquely identifying the link. This card or record is then followed by a card or record, for each commodity containing the appropriate cost and capacity information. The format is

```
CARD COLUMNS
1234567890123456 78
BBBBBBFFFFFFTTTTTT
 BLANK  FROM    TO    HEADER CARD
BBBBBBCCCCCCUUUUUU
 BLANK  COST    CAP   COMMODITY CARDS
```

where FROM, TO, COST, and CAPACITY are all right justified integers. For uncapacitated links, a capacity of "-1" should be entered in card columns 17-18.

An end-of-file marker consists of placing zeros in card columns 12 and 18 on a card or record following the last commodity data card or record.

Main Program Data File

- Header Card--this card may contain a title, or descriptive material, or be blank. The program skips over the first record.
- The next four cards initialize the following constants used by the program:

RGLMT the number of regions

PLTLMT the number of plants

PRDLMT the number of commodities

DCLMT the number of DC candidates

CUSLMT the number of customer zones

PENLTY the direct plant-to-customer penalty (may be 0)

MBIG limit on iterations before subgradient stepsize constant is adjusted

HEUR a suitable constant to signify whether the heuristic is to be activated--a value of "0" signifies yes, a "-1," no

MXCNT a number which represents one less than the maximum number of solution repeats permitted in resource or price direction.

HBIG limit on iterations before subgradient stepsize constant is adjusted when GMS branching rule is activated. If GMS branching rule is not used, HBIG can be left blank or set to "0."

JINCUM a switch constant to signify whether the algorithm is cold-starting or whether an existing incumbent is available in the proper format on the advanced start file structure. A value of "0" signifies cold start, a "-1," advanced start.

IARCS a switch constant which signifies the existence of an arc-sorted data structure in proper format on the Arc File Structure. A value of "0" signifies no, a value of "-1" yes.

IGNET a switch constant which signifies the solver option to be used in the Optimizer Support Module. A value of "0" signifies GNET is to be used, a "-1" signifies XNET.

IASTRT a suitable constant which signifies whether the solver will cold-start each time if it is called, or if the candidate queues will contain initial structural arcs. A value of "0" signifies cold start, a "-1" advanced start.

BTIME a time limit constant expressed in milliseconds which nominates a time in the running of the program when the GMS branching rule will be activated. If GMS is not used the constant BTIME will be ignored.

IPRT a print detail constant applicable to the subroutines GNET and XNET. This will normally be set to "0" which suppresses all output. If a detailed solution to each commodity-independent capacitated transshipment problem is desired, set IPRT to "1".

DEBUG a print detail constant for the MEDOS program. A value of 25 provides detail, including optimal flow information and enumeration data, a value of 26 suppresses the flows, and a value of 27 suppresses the enumeration data.

OUT a constant which signifies the desired machine-specific output device. For the DEC-10 computer a value of 5 signifies TTY, 3 the line printer, and 1 or 20-24 disk.

MAXTIM a time limit in milliseconds on running the algorithm. If this time limit is exceeded, the existing solution will be written to the output device specified by OUTDSK.

OUTDSK a constant which specifies the output device to which solutions are written if the algorithm exceeds the time limit set by MAXTIM.

Format for Cards 2-5

	Columns					
Card	1-20	21-30	31-40	41-50	51-60	61-70
2	BLANK	RGLMT	PLTLMT	PRDLMT	DCLMT	CUSLMT
3	BLANK	PENLTY	MBIG	HEUR	MXCNT	HBIG
4	BLANK	JINCUM	IARCS	IGNET	IASTRT	BTIME
5	BLANK	IPRT	DEBUG	OUT	MAXTIM	OUTDSK

NOTE: All variables in cards 2-5 are integers, right justified in the space provided.

Card 6 initializes the following variables:

ARCDSK an integer constant which specifies the output device for the sorted arc structure file.

INDSK an integer constant which specifies the output device for an advanced solution when the algorithm starts from an advanced point and halts prematurely because the time exceeds the limit set by MAXTIM. This feature enables the last two solutions to be retained for comparative purposes. If this is not desired, set INDSK = OUTDSK.

Format for Card 6

```
CARD COLUMNS
12345678901234567890123456789012345678901234567890
BBBBBBBBBBBBBBBBBBBBAAAAAAAAAAIIIIIIIIII
                    BLANK    ARCDSK     INDSK
```

Cards 7 and 8 contain the following floating point variables:

EPSLON the solution tolerance--card 7.

IGAMMA the initial subgradient stepsize for resource direction--card 8.

JGAMMA the initial subgradient stepsize for price direction--card 8.

Format for Cards 7 and 8

```
CARD COLUMNS
12345678901234567890123456789012345678901
BBBBBBBBBBBBBBBBBBBEEE.EEEEEE              CARD 7
                BLANK    EPSLON
BBBBBBBBBBBBBBBBBBBIII.IIIIIIJJJ.JJJJJJ   CARD 8
                BLANK    IGAMMA    JGAMMA
```

Cards 9 and 10 set values for the following constants:

BIGNUM a large number used for invalidating arcs (by placing a BIGNUM cost on throughput) pertaining to closed DC's. Too large a value will cause integer overflow conditions in GNET or XNET. BIGNUM need only exceed the maximum source-to-destination chain length in the network. In the present investigation a BIGNUM of 2^{22} was used successfully.

BIGGER a larger number used to represent infinity for incapacitated arcs. In some applications a BIGGER = BIGNUM may be sufficient. BIGGER needs to at least exceed the maximum total aggregate demands for all commodities. In the present work BIGGER was set to a value equal to 2^{27}.

MAXNUM the maximum number of iterations permitted in the resource-directive loop before the Lower Bound is updated.

PDBIG the maximum number of iterations permitted in the price directive loop.

Format for Cards 9 and 10

```
CARD COLUMNS
12345678901234567890123456789012345678901234567890
BBBBBBBBBBBBBBBBBBBNNNNNNNNNNRRRRRRRRRR   CARD 9
                BLANK    BIGNUM    BIGGER
BBBBBBBBBBBBBBBBBBBMMMMMMMMMMPPPPPPPPPP   CARD 10
                BLANK    MAXNUM     PDBIG
```

Card 11 establishes values for the following floating point constants.

ALPHA the convex combination weight utilized in setting up an initial configuration.

BETA the switching constant used in CI formula for branching history (normally set at 0 or 1, but can attain other values).

DELTA the convex combination weight utilized in the CI formula.

Format for Card 11

```
CARD COLUMNS
12345678901234567890123456789012345678901234567890
BBBBBBBBBBBBBBBBBBBAAAAA.AAAABBBBB.BBBBDDDDD.DDDD
                BLANK     ALPHA      BETA     DELTA
```

Plant Records. After card 11, the next group of cards provides information concerning the plants. For each plant, a header card provides the plant description followed by data cards (one for each commodity) which define upper and lower limits on supply.

Format for Plant Cards

```
CARD COLUMNS
12345678901234567890123456789012345678901
PPPPPPPPPPPPPPPPPPPP                          HEADER CARD
   PLANT DESCRIPTION
####################LLLLLLLLLLHHHHHHHHHH    DATA CARDS
    COMMODITY NUMBER     SUPLO     SUPHI
```

Remarks: All entries on data cards are integers with the exception of "commodity number", and must be right justified in the space provided. No end-of-file marker is necessary.

DC Records. Following the Plant Records the next entries in the file pertain to candidate DC information. DC records consist of an initial header card followed by a card for each DC candidate. The header card may be blank or for convenience describe columnar headings as illustrated below.

Format for DC Cards

			COLUMNS			
CARD	1-20	21-30	31-40	41-50	51-60	61-70
HEADER	"DC"	"FC"	"VCC"	"TPLO"	"TPHI"	"REG"
DATA	NAME	FC	VCC	TPLO	TPHI	REG

Remarks: All entries in data cards with the exception of "NAME" are integers and must be right justified in the space provided. No end-of-file marker is necessary.

Customer Records. The last set of records in the file concerns customer data. Customer records consist of a header card containing the customer description and region assignment, followed by data cards (one for each commodity) containing demand data.

Format for Customer Cards

```
CARD COLUMNS
12345678901234567890123456789012345678901
NNNNNNNNNNNNNNNNNNNNBBBBBBBBBBRRRRRRRRRR    HEADER
                    NAME      BLANK    REGION
####################BBBBBBBBBBDDDDDDDDDD    DATA CARDS
    COMMODITY NUMBER      BLANK    DEMAND
```

Remark: All entries in the data cards, with the exception of the "commodity number" must be integers right justified in the space provided. No end-of-file marker is necessary.

6.4.2 Output

Output detail is controlled by the variables DEBUG and IPRT. The variable IPRT will normally be fixed at zero. Setting IPRT equal to one will provide detailed flow information for each commodity-independent subproblem. This information will normally be of no value. With IPRT fixed at one, three levels of output are of normal interest (other levels are possible, but would generally be required only for the purposes of software or algorithmic diagnostics).

Detailed Output: DEBUG = 25. This level provides summary information of each subproblem solution, the corresponding subproblem configurations, the optimal configuration and costs, optimal flows, and the following enumeration and fathoming statistics:

Enumeration Data

Number of Potential Subproblems (with complete enumeration)

Number of Subproblems Enumerated

Number of Multicommodity Transshipment Problems Solved

Number of Commodity-Independent Capacitated Transshipment Problems Solved

Average CPU Time for Solving Commodity-Independent Capacitated Transshipment Problems

Fathoming History

Number of Reverse Star Configuration Failures

Number of Reverse Star Completion Failures

Number of Aggregate Configuration Capacity Failures

Number of Aggregate Completion Capacity Failures

Lower Bound on Current Configuration Successes

Lower Bound on Completions Resulting in a Fathom

Lower Bound on Completions Resulting in a Branch

Number of Lagrangean Fathoms

In addition the following performance measurements are printed out:

CPU Time for Input

CPU Time for Heuristic

CPU Time for Algorithm Excluding I/O

Total CPU Time

For Price Direction:

Maximum Iterations/Loop

Average Iterations/Loop

Number of Loops

For Resource Direction:

Maximum Iterations/Loop

Average Iterations/Loop

Number of Loops

Average CPU Time for Solving Multicommodity Capacitated Transshipment Problems

This print-out option also provides complete plant, customer, and DC details by node number and description. The node numbers facilitate the interpretation of flow data which is output by node number. Under the DC descriptions the computed cost attractiveness indices are printed along with the cost and capacity information.

Input parameter settings are also printed as well as program options.

Summary Depth Output: DEBUG = 26. This level differs from Detailed Output in the following ways: supply, demand and cost data for plants, DC's, and customers are suppressed, and the subproblem solutions, configurations, and optimal flows are not printed out.

Summary Output: DEBUG = 27. This level is the same as Summary Depth except enumeration statistics, fathoming history, and algorithmic performance are suppressed.

6.4.3 Core Requirements

Let n = number of nodes

m = number of arcs

n_p = number of commodities

n_r = number of regions

n_c = number of customers

n_d = number of DC's

n_f = number of plants

CO = core requirements in words of magnetic memory

Assuming the software is dimensioned to accommodate a problem-specific application, minimum core requirements can be approximated using the following formula:

$$CO = \{4m(2n_r + 1) + n(n_r(n_p + 17) + n_c + 1)\}/n_r + n_p(n_c + n_r + 2n_f + 8) + n_c(28n_d + 6) + 4(n_r + n_f)$$

Applying this formula to the Australian case study problem, where $n = 168$, $m = 7260$, $n_p = 5$, $n_r = 15$, $n_c = 74$, $n_d = 43$, and $n_f = 7$,

$CO \simeq 162K$

CO can be thought of as peak core utilization. In order to reduce the magnitude of CO, the use of overlay structures was investigated, but ruled out since the support modules of MEDOS are not independent executable phases, but require continuous memory throughout the execution of the program.

6.5 Program Options

MEDOS provides the following menu of options.

Options specified on input:

Heuristic. The algorithm will bypass the heuristic and start by opening that DC with the lowest cost attractiveness index if the variable HEUR is given a value of "-1" on input.

Solver. The software is equipped with two solver routines to recover solutions to the capacitated transshipment subproblems, GNET or XNET. The subroutine XNET is a refinement of GNET in which the node numbers (multipliers) are only stored for those nodes which have successors in the predecessor array. Nodes which have no successors are called "danglers". The only purpose of saving the multipliers is for

possible use in pricing out non-basic variables in preparation for selecting an arc to enter the basis. During a typical iteration only a relative few of the multipliers are ever needed. In XNET only multipliers pertaining to those nodes which have successors are updated and stored. It is then possible to update the multipliers for dangler nodes as needed. Economies in execution time can be gained if the basis arborescence contains a number of danglers, since a significant amount of time during each pivot is devoted to updating the multipliers. Such is the case in a network which has many more sinks than sources. In this case, because the successor to a pure sink (a node which only receives) must be a node which is not a pure sink (one which also dispatches), there must be many nodes with no successors. Reference [10] provides a detailed discussion of the benefits which can accrue from selecting the XNET option. In MEDOS this option is specified on input by means of the switch variable IGNET. When this option is activated, the dual multipliers corresponding to the optimal basis, which are required for resource direction, are computed for all nodes prior to exiting the XNET subroutine.

Re-optimization. An option is provided for either GNET or XNET in which the candidate queue (the subset of non-basic variables to be priced out) is initially loaded with structural variables which were basic at the last exit and which remain valid transportation links. Without this mechanism, the candidate queue initially contains all artificial arcs. This advanced start option is nominated through the use of the variable IASTRT.

Run Limit. Due to problem size and/or limited computer resources, it may not be possible to dedicate computing facilities to a complete and continuous run, terminating at optimality, at one time. It may also be desirable to set up short runs to test the efficacy of various tuning parameter settings. And, finally, the user may welcome the ability to halt the algorithm prematurely when it appears that the marginal benefits of continuing would not offset the marginal costs. Through the use of the MAXTIM and JINCUM variables, these ambitions can be accommodated.

Options available by variant:

Branch Selection. Branch selection criteria cannot be altered by input specification. Variants of MEDOS are available which provide for the use of Max Sigma or GMS branching rules. Further work is needed in the area of branching criteria. As improvements are found, these will be incorporated in future versions of the code.

6.6 External Tuning Parameters

The following external tuning parameters exert influence to varying degrees on CPU times.

ALPHA may be set to any number between 0 and 1 inclusive.

BETA normally set at 0 or 1, but may take on any value between 0 and 1.

DELTA may be set to any number between 0 and 1 inclusive.

MBIG may be any positive integer. In the present work a value of 4 was used.

MXCNT may be any positive integer. In the present work a value of 2 was used.

HBIG may be any positive integer. In the present work a value of 32 was used.

BTIME may be any positive integer, and it depends on the structure of the problem and experimentation with runs examining the point at which a switch to the GMS rule might be helpful.

EPSLON may be any number between 0 and 1. In the present work a value of .0025 was used.

IGAMMA may be any number greater than 0 and less than or equal to 2. In the present work a value of 2.0 was used.

JGAMMA may be any number greater than 0 and less than or equal to 2. In the present work a value of 2.0 was used.

MAXNUM may be any positive integer. In the present work a value of 8 was used.

PDBIG may be any positive integer. In the present work a value of 8 was used.

CHAPTER VII

COMPUTATIONAL RESULTS

The MEDOS software was tested using a small test problem, TPRB, and the large-scale problem, AUSPRB, which was described in Chapter I. All experimentation was carried out on the DECSystem 10 computer at The University of Tennessee, Knoxville.

TPRB is a two-commodity problem with three plants, six candidate DC's, three customer zones, and eighty-six arcs. In comparison, the respective dimensions for AUSPRB, as indicated previously in this thesis, are five commodities, seven plants, forty-three DC's, seventy-four customers, and 7260 arcs.

During the period of software development and testing (January-August 1982), numerous computer runs of TPRB were conducted and the results recorded. It was observed that computational performance was influenced by the time of day, day of week, day of month, day of academic quarter, relative demands of other computer users, and similar factors. Consequently, the times reported here are to be interpreted as "typical" results obtained during the course of the study.

Results for TPRB are enumerated as follows. Times are in CPU seconds.

Parameter Settings:

EPSLON	.001
ALPHA	.3
BETA	0
DELTA	.55
MBIG	4
IGAMMA	2.0
JGAMMA	2.0
MAXNUM	20
PDBIG	20

CTP Solver:

GNET with no re-optimization facility.

Heuristic:

Time to Achieve Initial Solution	.174
Solution as a Percent of the Optimum	110.3

Total Time Required:

Excluding I/O and Heuristic	1.111
Total	1.418

Enumeration History:

Number of Potential Subproblems, M	64
Number of Subproblems Enumerated (% of M)	25 (39%)
Number of MCTP's Solved (% of M)	4 (6.25%)

Fathoming History:

Reverse Star Configuration Capacity Test	11
Reverse Star Completion Capacity Test, Fathoms	3
Configuration Aggregate Capacity Test	13
Completion Aggregate Capacity Test, Fathoms	1
Lower Bound Test and Configuration	0
Lower Bound Test on Completions, Branches	10
Lower Bound Test on Completions, Fathoms	8

A number of consecutive runs were made, with increments of .05 to determine the effects of varying the parameter DELTA. No great variation was observed for a DELTA greater than or equal to .55. The best performance was achieved with DELTA = .55. For DELTA less than .55, the performance diminished. At DELTA = .5 the CPU time was 90% higher than for DELTA = .55, 364% greater for DELTA = .45, and then dropped to 153% for DELTA less than or equal to .4. Due to the size of AUSPRB, and the associated cost of computer runs, it was not possible to test a wide range of DELTA settings. However, it was ascertained that, for this particular problem, DELTA = .55 was inferior to DELTA = 1.

The only influence of the parameter ALPHA is in the number of DC's opened in the initial configuration. A large ALPHA will result in more DC's in the initial configuration, a small ALPHA less. In TPRB, an ALPHA = .3 worked best; in AUSPRB an ALPHA = 0 was used.

Four factors control CPU time:

1. the efficiency with which the CTP's are solved.
2. the efficiency with which the MCTP's are solved.
3. the external tuning parameter settings.

4. branch selection criteria.

Problem AUSPRB was used to test all of these factors.

7.1 CTP Efficiency

The original version of MEDOS used a variant of GNET to solve the CTP's. Prior to calling GNET, the software breaks the multicommodity network down into smaller commodity-specific networks. This procedure reduces the number of arcs in each (CTP_p) network by 80% for AUSPRB. The solution time for each such GNET solution averaged .525 CPU seconds. CTP solution time is a significant factor in the overall computational efficiency of the algorithm (tens of thousands of CTP's are solved in AUSPRB). Considerable attention focussed on reducing this time.

The first refinement placed a switch option in the code which permitted, on input, the user to specify XNET, rather than GNET, as the CTP optimizer. XNET is tailored to problems in which the basis arborescence, corresponding to any iteration, contains many dangler nodes. A dangler node is one which has a predecessor but no successor nodes. XNET aggregates the dangler nodes into a more compact network representation and the dual variables corresponding to these nodes are only updated as needed [10]. Since most of the CPU time consumed by each CTP pivot is devoted to pricing, XNET is more efficient than GNET for such networks. The use of XNET in AUSPRB reduced the average solution time, computed over 34,000 solutions, to .484 CPU seconds.

A second refinement incorporated a re-optimization facility into the code. Each time GNET, or XNET, is called, if the re-optimization mechanism is activated, the nonbasic arcs which are initially priced

for basis candidacy are those arcs which were in the optimal basis, and are still valid, the last time a CTP was solved. In running AUSPRB, the re-optimization facility has not appeared to offer any improvement in CPU times over cold-starting. This may be indicative of the efficiency with which GNET and XNET can quickly find "good nonbasics," and also the small number of pivots necessary to converge to optimal CTP solutions. The average number of pivots for the advanced start feature was 260 compared to 274 for cold-starting.

7.2 Efficiency of MCTP's

The tightness of the price directive/resource directive bounds was verified by the small number of iterations per loop required to achieve MCTP optimality. The following relevant statistics reflect the average performance over 29 MCTP solutions.

Price Direction	8.04 iterations/loop
Resource Direction	3.55 iterations/loop
CPU Time to Optimality	31.942 seconds

More experimentation across a wider range of problems is necessary, but the results for AUSPRB appear to be superior to those reported by Kennington and Shalaby [85] for problems of similar dimensions.

7.3 External Tuning Parameter Settings

Due to the cost of running AUSPRB, only limited experimentation with parameter-setting variations was possible. Of those tried, the following settings seemed to work best:

EPSLON	.0025
ALPHA	.100
BETA	0
DELTA	1.00
MBIG	4
MXCNT	2
HBIG	32
BTIME	2000
IGAMMA	2.0
JGAMMA	2.0
MAXNUM	8
PDBIG	8

7.4 Branching Rules

The anticipated benefits from the GMS Rule did not materialize. The extra time required to solve for the greatest marginal savings appeared to exceed the benefit derived from it, even though the number of extra CTP's which had to be solved was generally a small subset of FR (usually no more than 6 to 8).

When the Max Sigma Rule was activated, no noticeable reduction was activated in the number of wasted branching operations (WBO's), when compared to the use of other rules, could be detected.

Accordingly, the branching criteria used in solving AUSPRB were the RS, Max UB, Max TP, Reg Cap, Min CI, and Pos LB prioritized in accordance with Section 5.2.3.9.

The MEDOS algorithm produced quality solutions within reasonable and affordable CPU time frames. The results were enlightening. The heuristic produced an initial solution, in 35 CPU seconds, of $27,930,467. The configuration for this solution consisted of twenty-seven DC's, six more than the firm's existing configuration. The initial configuration differed from the existing configuration in the following ways:

Closures:

Major:	Toowoomba	Minor:	Darwin
	Bundaberg		Grafton
	East Sydney		
	Canberra		

Openings:

Major:	Nambour	Minor:	Lismore
	Wollongong		Tamworth
	North Sydney		Bathurst
	Albury		Whyalla
	East Melbourne		Elizabeth
			Kalgoorlie
			Bunbury

It is interesting to note that the reconfiguration produced by the heuristic nets approximately $2 million savings over the firm's estimated $30 million annual distribution budget.

After only 33 CPU minutes a solution, $S^{\#}$, within 16.9% of the optimum was achieved. The cost of $S^{\#}$ was $19,056,397 and the impact of the corresponding configuration on the existing configuration was:

Closures:

Major:	Toowoomba	Minor:	Darwin
	Bundaberg		Grafton
	Newcastle		Launceston
	East Sydney		
	Canberra		
	Ballarat		
	Perth		
	Hobart		

Openings:

Major:	Nambour	Minor:	Tamworth
	Wollongong		Bathurst
	North Sydney		
	Albury		

After an additional 63 CPU minutes a solution, $S^{\dagger}$, which was within 1.7% of the optimum was found. The cost of this solution was $16,579,249 and the association configuration differed from the existing configuration as follows:

Closures:

Major:	Toowoomba	Minor:	Darwin
	Newcastle		Launceston
	East Sydney		
	Canberra		
	Geelong		
	North Melbourne		
	Ballarat		
	Adelaide		
	Perth		
	Hobart		

Openings:

Major:	Nambour	Minor:	Tamworth
	North Sydney		Bathurst
	Woolongong		Elizabeth
	Albury		

Termination of the algorithm required 809 CPU minutes with only marginal improvement in $S^{\dagger}$. The optimum solution S* had a cost of $16,308,677 which dictated the following changes to the existing configuration:

Closures:

Major:	Toowoomba	Minor:	Cairns
	Townsville		Darwin
	East Sydney		Grafton
	Canberra		Broken Hill
	Geelong		Launceston
	North Melbourne		
	Ballarat		
	Adelaide		

Openings:

Major:	Nambour	Minor:	Lismore
	East Melbourne		
	Albury		

What are the implications of these results? Significant savings can be realized at a nominal computer cost. For a cost of about $200 (DEC-10), a potential savings of $11 million (or 35%) is indicated. An additional $400 of computing increases the savings to $13.5 million (or 45%).

Detailed statistics comparing solutions S^*, $S^\dagger$, and $S^\#$ follow. These solutions were obtained using XNET with re-optimization.

Enumeration History:

	S^*	$S^\dagger$	$S^\#$
Potential Subproblems, M	8.8×10^{12}		
Subproblems Enumerated, M_e	13108	2501	552
% of M	1.49×10^{-7}	2.84×10^{-8}	6.28×10^{-9}
MCTP's Solved	31	30	29
% of M	3.52×10^{-10}	3.41×10^{-10}	3.30×10^{-10}
% of M_e	2.36×10^{-1}	1.20	5.25
CTP's Solved	83850	26990	6095
CTP Average Time (secs.)	.509	.490	.504
Average CTP Pivots	265	261	258

Fathoming History:

	S^*	$S^\dagger$	$S^\#$
Reverse Star Configuration Capacity Test	11535	2457	514
Reverse Star Completion, Fathoms	85	38	23
Aggregate Configuration Capacity Test	198	0	0
Aggregate Completion Capacity, Fathoms	9	0	0
Lower Bound on Configuration Test	1344	14	9
Lower Bound on Completions, Fathoms	6459	1205	246
Lower Bound on Completions, Branches	6524	1228	254

CHAPTER VIII

NONLINEAR COST STRUCTURES

8.1 Cost Elements

Costs which accrue from the operation and maintenance of a physical distribution system have many origins, and in general can be classified as either fixed (volume independent) or variable (some function of volume).

Fixed costs include all of those annual costs associated with the establishment and possession of a DC. These include, but are not limited to, depreciation or leasing of building, plant, and equipment; interest on capital; salaried labor; insurance; and portions of utilities and taxes.

Variable costs generally include all those costs incurred in the operation of the system. Labor for in and out handling, inventory maintenance and control, and order picking and packing are normally the major sources of variable cost. Examples of other costs which are variable are computing, spoilage and pilferage, packing and shipping materials, and pallets.

Hard-to-classify costs are sometimes subjectively labeled "pseudo-fixed," or "pseudo-variable"; otherwise some formula must be devised to apportion the total to each of the two categories. An example of a cost element which is not easily classified is electricity for the operation of a high rise cold store warehouse. How does the cost compare for chilling a half empty warehouse to a warehouse filled to

capacity? It is important that such dilemmas be resolved and that the respective fixed and variable cost components be as accurate a reflection of reality as possible. Having properly isolated the variable cost elements, the analysis of how those costs respond to perturbations in volume is of interest in formulating planning models and preparing input for those models.

The accuracy of distribution planning models depends on the precision with which the behavior of all relevant cost functions can be approximated. Research is deficient in the area of distribution cost modeling, a point reiterated in Chapter IX of this thesis. Consequently distribution modelers have had to cope without the luxury of robust cost generators and rely on either available cost information, which may be grossly inadequate, or invest a great deal of effort in compiling cost data tailored to the specific requirements of the model used. Even in the latter case, the modeler often finds he is at the mercy of a firm's management information system which was not designed with the needs of strategic distribution planning in mind. Two difficulties commonly arise:

1. Most cost information collected is accounting-oriented for use in the financial management of the enterprise. As a result, costs are normally aggregated by responsibility area, rather than by functional area. This procedure makes it extremely difficult, if not impossible, to isolate those cost elements which pertain to the distribution function. Even the aggregated costs which are available paint a somewhat

distorted picture. Traditional accounting practice recoups administrative and overhead expenses by levying on each responsibility operating area an aggregate per unit burden rate applied to some meaningful unit of production (e.g., man hours, machine hours, direct labor dollars). Such rates are useful for the purpose of recovering indirect costs, but of little use for cost analysis. Additionally, there is an underlying assumption of linearity which may be invalid; and the rates are set with respect to either a single volume reference point for a past period or some projected reference point in the future.

2. The level of activity in any given existing DC does not typically fluctuate enough to use as a benchmark for broad cost-volume relationships. That is, it would be unusual to find a DC for which cost information was available across the entire range of permissible volumes, lower to upper throughput bounds. Since the majority of variable costs are non-discretionary (that is, labor), it is usually not possible to sustain an operation with significant fluctuations in volume. DC's will normally operate within a restricted range and the operations will be on a commensurate scale.

It is the purpose of this chapter to theorize on possible cost structures and suggest some strategies which may be helpful in introducing some more generalized cost functions into the multi-echelon planning model.

8.2 Theoretical Cost Structures

In practice variable cost functions may be convex (economies of scale), concave (diseconomies of scale), linear (directly proportional), or may be some compound cost function which is neither strictly convex nor strictly concave over the entire throughput spectrum. Additionally, functions may be continuously differentiable or have points of discontinuity and/or non-differentiability, and be well defined or ill defined.

A range of possibilities is illustrated in Figures 8.1-8.8.

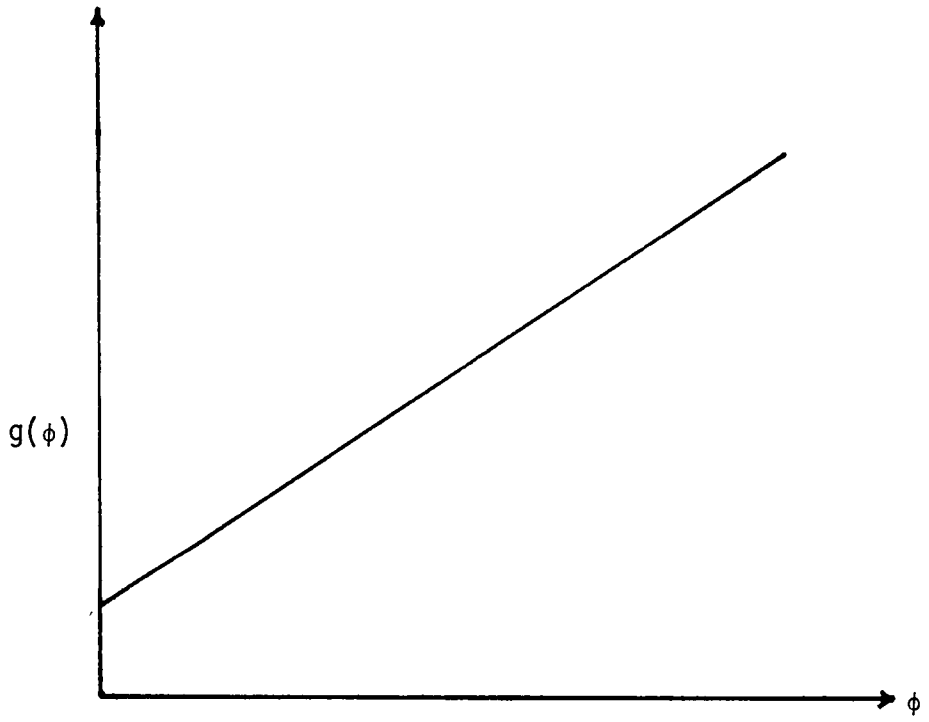

Figure 8.1. Linear Cost Function

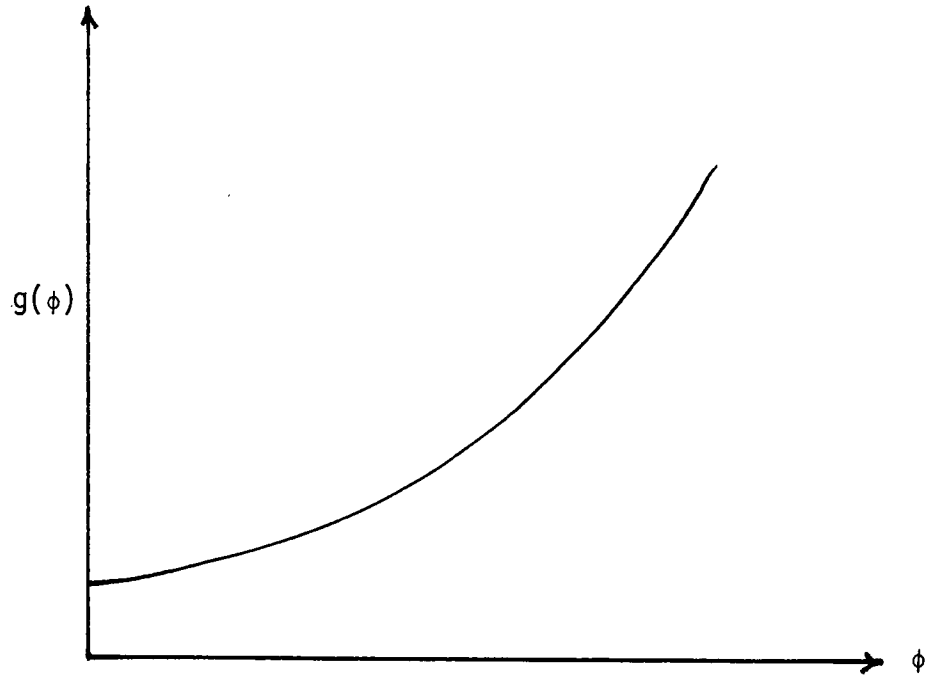

Figure 8.2. Convex Continuous Cost Function

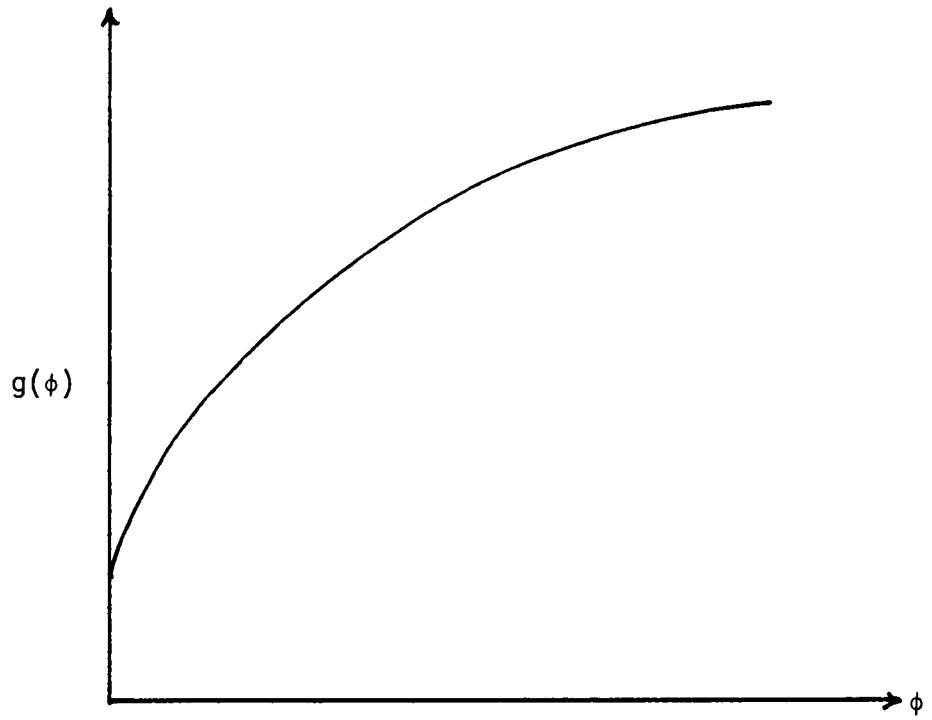

Figure 8.3. Concave Continuous Cost Function

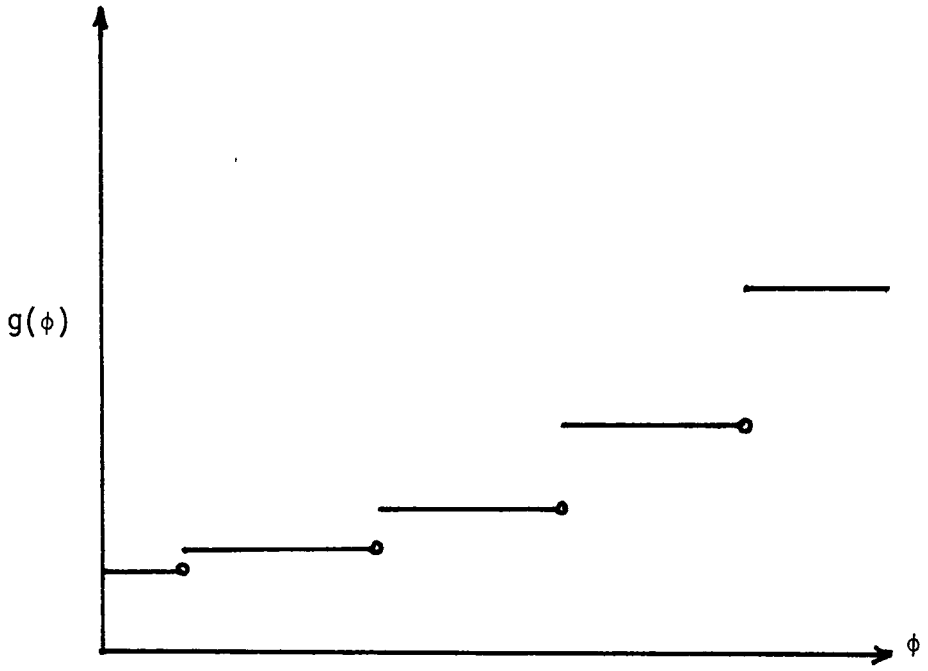

Figure 8.4. Convex Discontinuous Cost Function

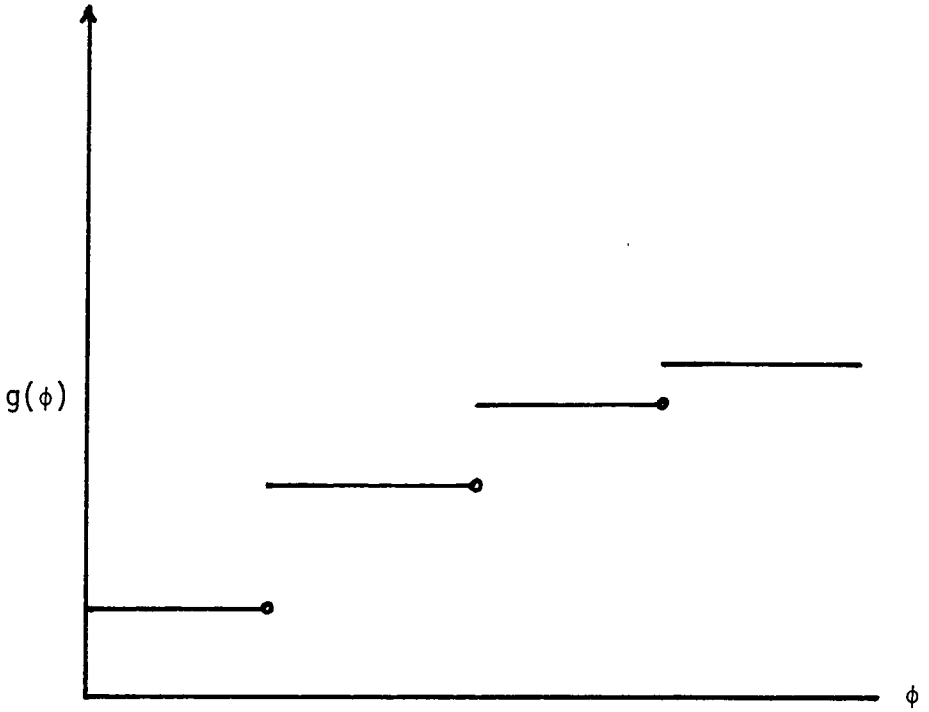

Figure 8.5. Concave Discontinuous Cost Function

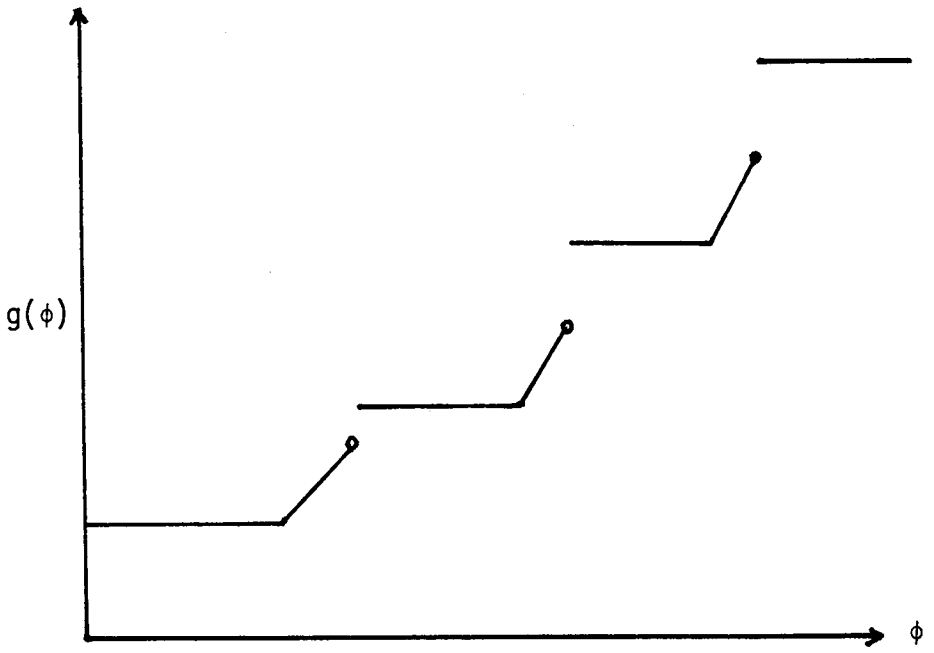

Figure 8.6. Convex Discontinuous Cost Function

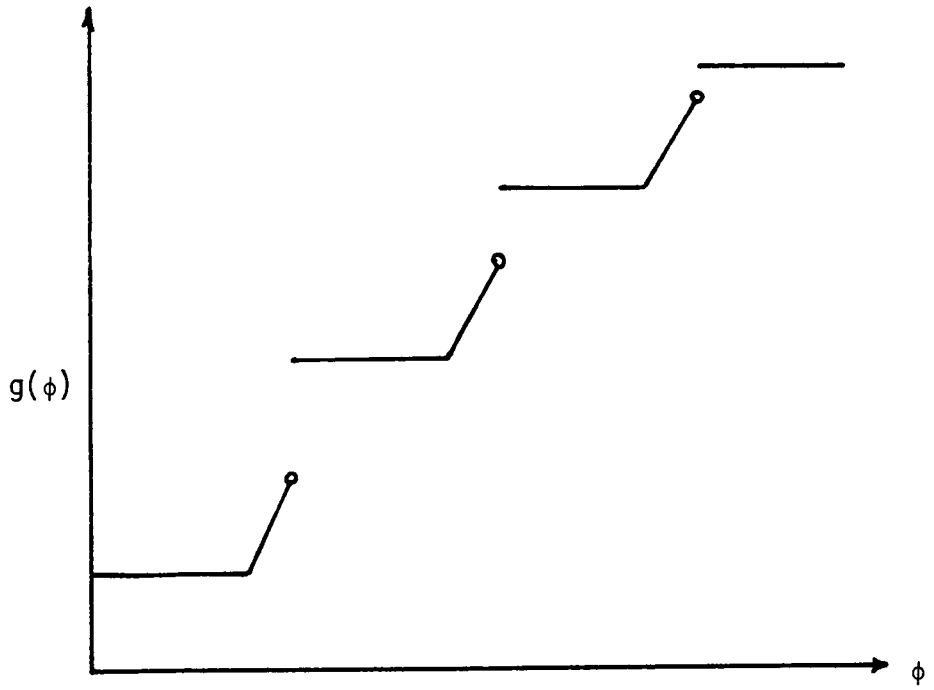

Figure 8.7. Concave Discontinuous Cost Function

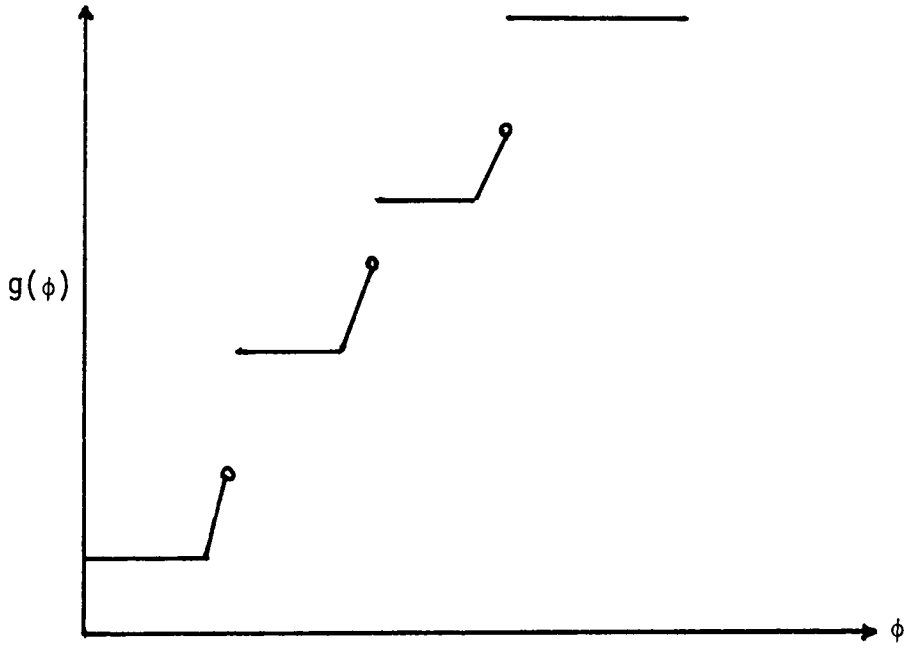

Figure 8.8. Concave-Convex Discontinuous Cost Function

The cost structures illustrated in Figures 8.4-8.8 stem from the idea that marginal costs do not change incrementally with each unit change in volume. Conversely, a given level of costs will apply across a certain volume range, at which point the costs will jump to a higher level (e.g., when another worker must be hired, another forklift truck purchased, etc.). Figures 8.6-8.8 illustrate the management practice of "stretching" the practical capacity through the use of overtime, short term leasing, and the like.

Figure 8.8 is the likely shape encountered in many systems. If a third order polynomial were "fit" to the data points, this curve would have a characteristic reflected "S" shape as shown in Figure 8.9.

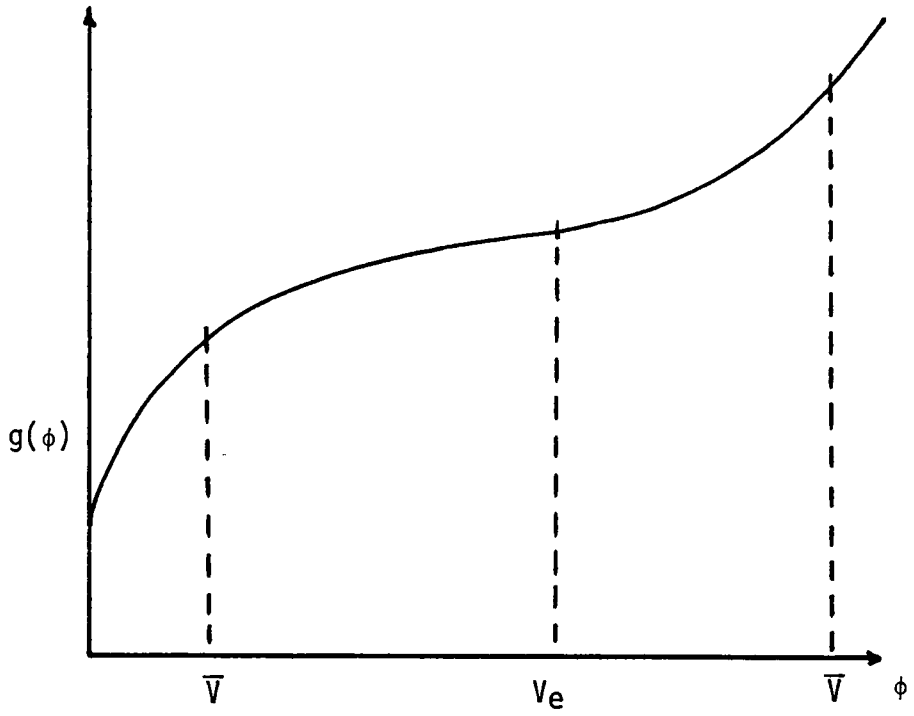

Figure 8.9. Concave-Convex Continuous Cost Function

The cost structure across the domain of interest depends on how the term "upper throughput capacity" is defined. If the term "capacity" is taken to mean "maximum" capacity, this upper bound is not likely to reflect an economical operating level (perhaps far from it). In this case the level of greatest operating efficiency, symbolized by V_e in Figure 8.9, will be a point less than $\overline{V}$. If the operating cost curve is concave over most of the throughput range, one would expect a point of inflection to be reached at which point the marginal costs are stationary with respect to volume changes. Beyond this point (as the warehouse becomes over-filled, access is restricted, aisles are blocked, damage and spoilage rates increase, and personnel are overworked) the marginal costs increase at an increasing rate.

8.3 Technical Modeling Difficulties

No difficulties are experienced when the cost structure is assumed to be linear or a piece-wise linearization of a convex function. In the linear case it has been shown in the present work that the operating costs can be merged with the transportation costs for a compact model representation. Similarly, convex functions pose no problems. Assume DC X has a uniformly increasing convex cost function which has been linearized as shown in Figure 8.10.

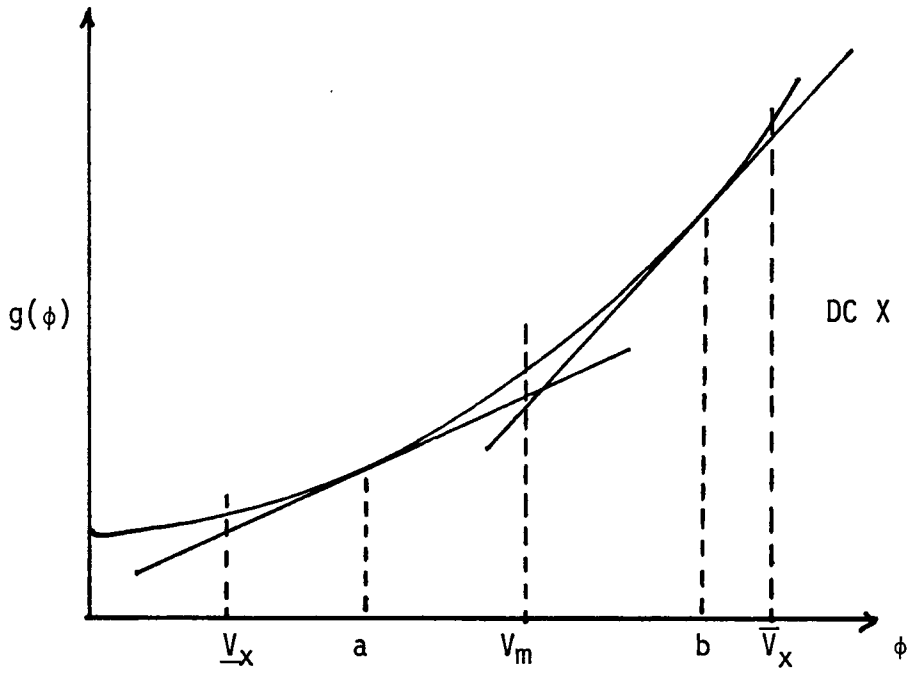

Figure 8.10. Linearization of Convex Cost Function

A simple trick can be employed to accommodate this structure within the framework of a linear model. This device temporarily subdivides the single candidate DC into a number of bogus DC's, one corresponding to each linear piece. For example, referring to Figure 8.10, DC X

becomes DC X_a and DC X_b. DC X_a has fixed cost $g(\underline{V}_x)$, variable cost $\nabla g(a)$, and upper and lower throughput bounds of V_m and $\underline{V}_x$, respectively. DC X_b has fixed costs $g(V_m)$, variable costs $\nabla g(b)$, and upper and lower throughput bounds of $\overline{V}_x$ and V_m respectively. Due to the convexity of g and the assumption that g increases uniformly in the domain of interest, $\nabla g(a) \leq \nabla g(b)$, and $g(\underline{V}_x) \leq g(V_m)$. A cost minimization algorithm will always select DC X_a in preference to DC X_b for the first V_m units. This can be shown as follows:

Let C_h = {r} = the set of feasible chains in the network,

and $\psi_x \supseteq C_h$ = {r: x lies on chain r}.

Let c_r = the length of chain r with respect to transportation costs.

The objective function of the distribution/location model (MDM, Chapter III) can be expressed in the following arc-chain formulation.

$$\text{Minimize} \sum_r c_r x_r \sum_{k \in D-\{x\}} [f_k z_k + g_k(\phi_k)]$$

$$+ [g(\underline{V}_x) z_{x_a} + \nabla g_x(a) \sum_{r \in \psi_x} x_r] + [g(V_m) z_{x_b} + \nabla g_x(b) \sum_{r \in \psi_x} x_r]$$

Clearly, for $r \in \psi_x$, c_r is not affected by the values of z_{x_a} and z_{x_b}. Also, $x_r \geq 0$ for $r \in \psi_x$ only if $z_{x_a} = 1$, $z_{x_b} = 1$, or both. Otherwise, $x_r = 0$.

Since $g(\underline{V}_x) \leq g(V_m)$ and $\nabla g_x(a) \leq \nabla g_x(b)$, $z_{x_a} = 1$ will always yield a lower objective function value than $z_{x_b} = 1$. Even if both DC X_a and DC X_b are locked open, DC X_b could not be an attractive variable to enter the basis until DC X_a were out of the basis at the upper bound.

If g is concave, the arguments above do not hold; $g(\underline{V}_x) \leq g(V_m)$, but if g is concave, $\nabla g_x(a) \geq \nabla g_x(b)$. Hence, a trade-off exists

between fixed and variable costs, and there is no assurance that a cost minimizer will open the DC's in the correct order (X_a before X_b) or that, once open, DC X_a arcs will be fully saturated before flow is permitted on the corresponding arcs of DC X_b.

What is the alternative? The literature bears silent testimony to the need for research in nonlinear modeling as applied specifically to distribution planning. There is an emerging body of literature which has revitalized some concepts which have been successfully used in the oil industry for over two decades. These techniques extract solutions to nonlinear minimization problems in the context of a linear modeling framework [107,108,115]. The major drawback with these methods is the requirement that the cost functions be well-defined and twice-continuously differentiable. Even nonlinear programming methods rely on this underlying assumption. A second major technical difficulty, then, lies in the fact that most distribution cost functions encountered in practice are not well defined mathematically and are step-type functions which have points which are discontinuous or which represent abrupt changes in direction at which points derivatives are undefined. This difficulty can sometimes be successfully overcome by approximating the function with a regression model or some curve-fitting technique applied to empirically collected data.

8.4 Possible Solution Strategies

It is the purpose of this section to suggest some nonlinear strategies which can fully utilize the linear machinery developed by the

research documented in this thesis. Two cases will be considered. In Case I the precise mathematical functions are unknown, but it is assumed that the functional values and gradients can be estimated at key reference points. In Case II, it is assumed that a well-defined twice-continuously differentiable cost function is known for each candidate DC (obtained perhaps by applying some polynomial matching technique to scatter diagrams of empirical data).

8.4.1 Case I--Functions Unknown but Point Estimates Available

Refer to Figure 8.11. The concave function $g(\phi)$ is the actual, but known cost function for some hypothetical DC with upper and lower throughputs $\overline{V}$ and $\underline{V}$ respectively, and fixed costs f'. The linearization g' represents an accounting-oriented approximation with respect to the volume reference point V_e. If V_e is the projected annual level of activity, g' suits the purposes of fiscal management--that is, if distribution services are costed at the rate g', under absorption of costs will occur if actual volume is less than V_e, over absorption if greater than V_e, with breakeven occurring at V_e. The reference point V_e is manipulated year by year to coincide with forecasted throughput levels. For the purposes of distribution modelling, one can do much better than g'.

The most obvious refinement stems from the observation that the domain of interest is $\underline{V}$, $\overline{V}$ only. Since a DC will not be permitted to operate at a level below $\underline{V}$, $g(\underline{V})$ can be thought of as a component of fixed cost. If $g(\underline{V})$ is known, or can be estimated, the improved linearization, g", illustrated in Figure 8.12, can be constructed,

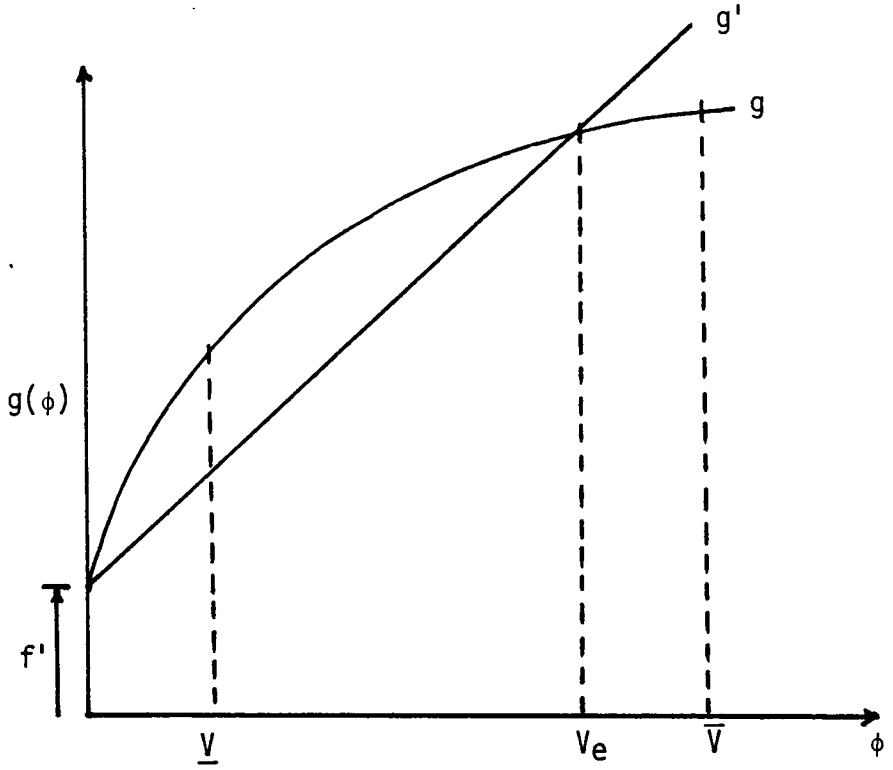

Figure 8.11. Linearization of Concave Cost Function

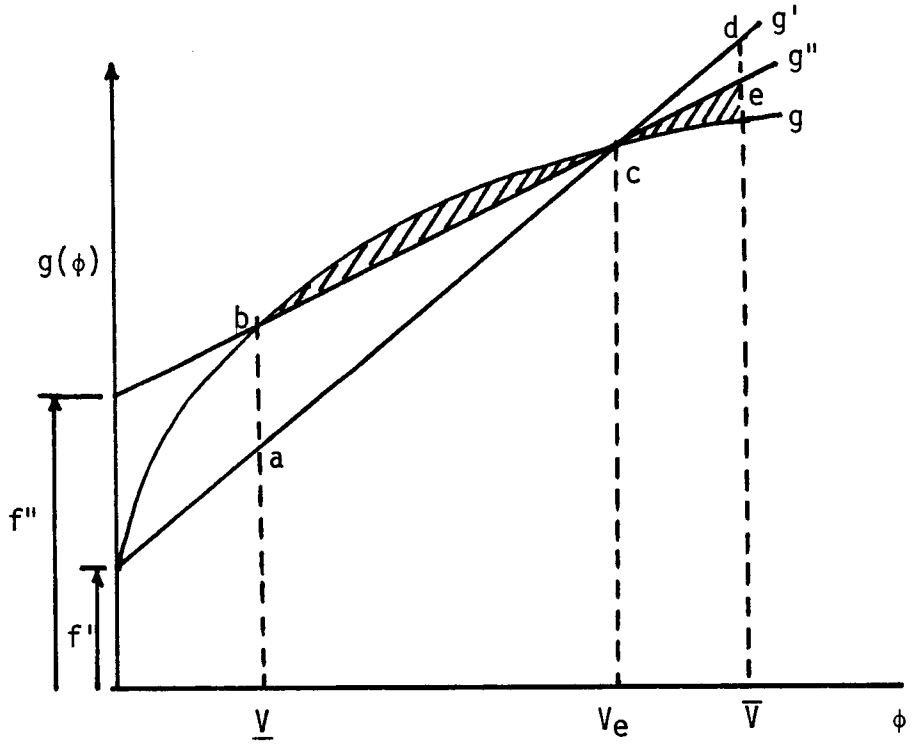

Figure 8.12. Improved Linearization of Concave Cost Function

with $\nabla g'' = (g(V_e) - g(\underline{V})) / (V_e - \underline{V})$ and $f'' = g(\underline{V}) - \nabla g\underline{V}$. Function g" is clearly superior to g', since the error of the estimate (represented by the cross-hatched area), induced by g" is less than the error induced by g'. The precise measure of this difference is the sum of the areas of triangles abc and cde.

Alternative approximations can be used to establish bounds on the error of estimate. Refer to Figure 8.13. If it can be assumed that $g(\overline{V})$ is known, function h provides a lower bound on g since it underestimates g everywhere due to the concavity of g.

Obtaining an upper bound on g is more difficult. An obvious choice is h' (again, refer to Figure 8.13) where $\nabla h' = \nabla h$. The difficulty may lie in locating point (V', g(V')). An alternative which may prove to work well in practice is h", which can be constructed if $\nabla g(\overline{V})$ can be estimated (or it may be preferable to construct h" with $\nabla h'' = \nabla g(V_e)$.

For notational reference, the use of h' will be called interval slope upper bounding (ISUB), and the use of h" least slope upper bounding (LSUB).

Define the following pair of problems.

$(MDML_L)$ Minimize $\sum_{p\epsilon P} \tilde{c}_p x_p + \sum_{k\epsilon D} f_k^L z_k$

Subject to (3.7) and (3.10)-(3.12)

where

$\tilde{c}_{ijp} = c_{ijp}$ $\quad (i,j)\epsilon A\text{-}D$

$\tilde{c}_{kp} = c_{kp} + \nabla h$ $\quad k\epsilon D$

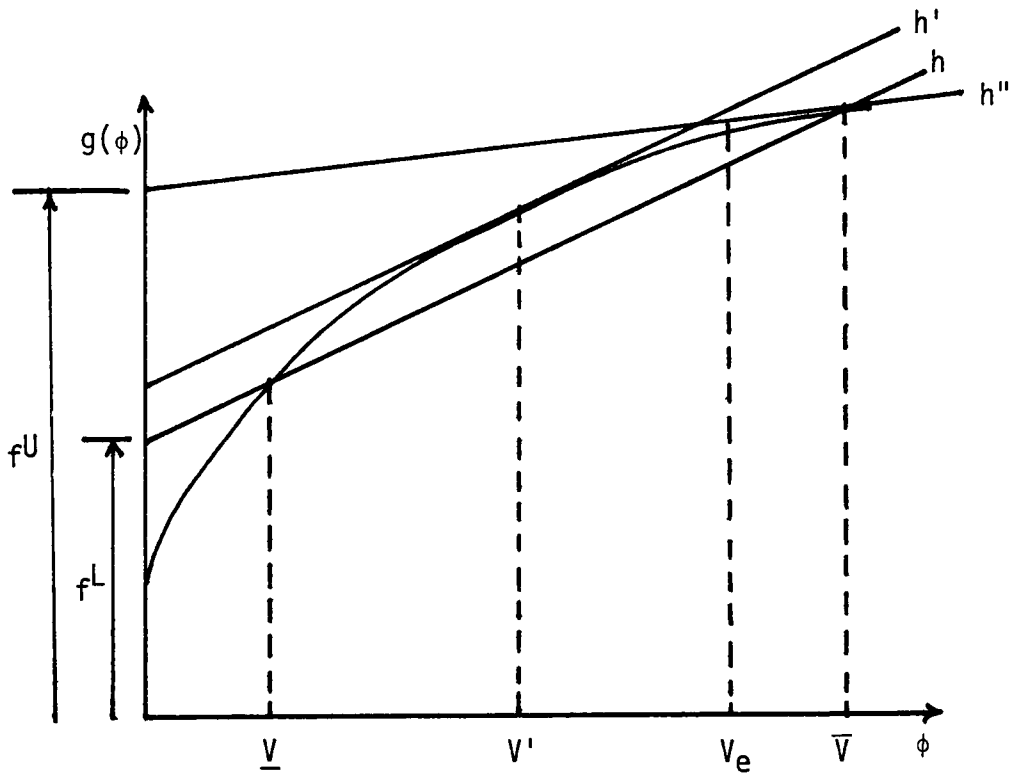

Figure 8.13. Upper and Lower Bounding Linearizations

$$(\text{MDML}_U) \quad \text{Minimize} \sum_{p\epsilon P} \tilde{c}_p x_p + \sum_{k\epsilon D} f_k^U z_k$$

Subject to (3.7) and (3.10)-(3.12)

where

$$\tilde{c}_{ijp} = c_{ijp} \qquad (i,j)\epsilon A\text{-}D$$

$$\tilde{c}_{kp} = \begin{cases} c_{kp} + \nabla h' & k\epsilon D \text{ and ISUB Method} \\ c_{kp} + \nabla h'' & k\epsilon D \text{ and LSUB Method} \end{cases}$$

A possible solution approach is to solve MDML_L for $\upsilon^*(\text{MDML}_L)$ and MDML_U for $\upsilon^*(\text{MDML}_U)$ using MEDOS or some other cost minimizer. A nonlinearity gap can be computed as $\varepsilon_n = \upsilon^*(\text{MDML}_U) - \upsilon^*(\text{MDML}_L) \; / \; \upsilon^*(\text{MDML}_L)$.

8.4.2 Case II--Functions Well Defined and Twice-Continuously Differentiable

The Method of Approximate Programming (MAP)[107,108], or Successive Linear Programming (SLP)[115], as it is referred to by some practitioners, is a technique for solving nonlinear programs via a sequence of linear programs.

Assume $g_k(\phi_k)$ defines the cost function for DC k, and that g_k is twice-continuously differentiable in the domain $\underline{V}_k$ to $\overline{V}_k$.

Define the Jacobian matrix for each DC k by

$$J_k(\phi) = (\delta g_k(\phi)/\delta\phi) \ .$$

Starting with a base point $\overline{\phi}_k$ (for instance $\underline{V}_k$) each cost function g_k is approximated by a first order Taylor series about point $\overline{\phi}_k$.

$$g_k(\phi_k) \tilde{=} g_k(\overline{\phi}_k) + J_k(\overline{\phi}_k)\Delta\phi_k \qquad (8.1)$$

where

$$\Delta\phi_k = \phi_k - \overline{\phi}_k \ .$$

Substituting (8.1) into (MDM) yields

(MDML') Minimize $\sum_{p\epsilon P} c_p x_p + \sum_{k\epsilon D} [f_k z_k + \nabla g_k \Delta\phi_k] + g_k(\overline{\phi}_k)$

Subject to (3.7), (3.10)-(3.11) and

$$\underline{V}_k z_k - \overline{\phi}_k \leq \Delta\phi_k \leq \overline{V}_k z_k - \overline{\phi}_k \qquad k\epsilon D \qquad (8.2)$$

$$\ell_{kp} \leq \Delta x_{kp} \leq u_{kp} \qquad k\epsilon D,\ p\epsilon P \qquad (8.3)$$

which reduces to

Minimize $\sum_{p\epsilon P} \tilde{c}_p \tilde{x}_p + \sum_{k\epsilon D} f_k z_k$

Subject to (3.7), (3.10)-(3.11), (8.2), and (8.3)

where

$$\tilde{c}_{ijp} = c_{ijp} \qquad (i,j)\in A\text{-}D,\ p\in P$$

$$\tilde{c}_{kp} = c_{kp} + \nabla g_k \qquad k\in D,\ p\in P$$

$$\tilde{x}_{ijp} = x_{ijp} \qquad (i,j)\in A\text{-}D,\ p\in P$$

$$\tilde{x}_{kp} = \Delta x_{kp} \qquad k\in D,\ p\in P$$

$$\Delta\phi_k = \sum_{p\in P} \Delta x_{kp} \qquad k\in D$$

Using the MEDOS algorithm, solve MDML' for an optimal $\tilde{x}$, $\upsilon^*(MDML')$.

Compute

$$\Delta\phi_k = \sum_{p\in P} \tilde{x}_{kp} \qquad p\in P$$

Set

$$\bar{\phi}_k \leftarrow \bar{\phi}_k + \Delta\phi_k \qquad k\in D$$

and repeat the process. The stopping criterion is when the improvement in $\upsilon^*(MDML')$ is negligible from one iteration to the next.

8.5 Future Directions

A possible shortcoming in the use of the methods proposed in the previous section is the requirement for multiple computer runs of the model. This can represent a significant cost for large-scale problems, and may be hard to justify on cost-benefit grounds when the cost functions used do not have a solid scientific base.

Cost modeling remains a lucrative area for further investigation. Linear optimizers and accompanying software packages will likely continue to dominate physical distribution modeling for some time. Where cost functions are well defined some efficient machinery presently exists for solving nonlinear network problems. See for example

Rosenthal [121]. A portable and robust distribution cost generator model would have immense value not only in the application of nonlinear methodologies but to complement general purpose linear algorithms such as the one described in this thesis. It is hoped that this work and the compelling requisite in the marketplace will provide the necessary catalysts to incite the research energies necessary to fulfill this need.

CHAPTER IX

DIRECTIONS FOR FURTHER RESEARCH

During the course of the research reported in this dissertation, two major issues arose which, due to practical time limitations, could not be resolved within the scope of the present work: (1) nonlinear cost generation and (2) DC configuration pegging. Additionally, it became apparent that certain further extensions to the modeling capability would be of immense practical benefit, in that such extensions would expand the range of model applicability.

1. Nonlinear DC Operating Costs Modeling. As indicated in Chapter VIII, a portable and robust cost generator model, which can provide consistent and accurate operating costs as a function of throughput volume, is vitally needed to complement the existing modeling capability.

2. Nonlinear Transportation Costs. An enrichment would be to incorporate concave transportation costs, as functions of volume shipped, into the model. In one respect the incorporation of concave transportation costs is easier than the inclusion of concave DC operating costs; in another respect, harder. Transportation costs as a function of volume are normally well defined for a given mode and route. Consequently, the generation of cost data with respect to transportation does not pose any great problems. On the other hand, the sheer magnitude of large-scale problems could render pricewise linearization approaches computationally impractical. There is a vast difference in computing and updating the gradients for 43 DC's and in doing so for more than 7000 arcs.

3. Generalized Costs. The procedures outlined in Chapter VIII were based on assumptions of concavity, continuity, and differentiability. A model which could totally relax all of these assumptions and deal with a generalized cost structure would be of great value in practice. Complete generality would be very difficult to achieve. Steepest descent methods rely on the properties of convexity, continuity, and differentiability. Relaxing the latter property paves the way for utilizing subgradients in the place of gradients, but subdifferentials do not exist at points of discontinuity.

4. DC Pegging Rules. Branch and bound convergence could be accelerated if pegging rules were available, which enabled certain DC's to be fixed open or closed in any completion of a given configuration. There are no obvious criteria for a multi-echelon system in which DC's are permitted to inter-transact, short of solving a number of bounding CTP problems.

5. Single-sourcing. An enhancement of the model would enforce single-sourcing of customers by DC's, at least for a "bundle" of commodities. This extension would require a new set of binary variables to represent the customer-DC assignments. Computationally, for large problems, branch and bound as a solution technique may be impractical since the number of potential subproblems is greatly magnified.

6. Simultaneous Plant and DC Location. It may be desirable in some cases to incorporate the location of the sources as well as the transshipment ports into the model. The effect of this extension would be to add the fixed cost term $\sum_{i \in F} g_i y_i$ to the objective function where g_i is the establishment cost of plant i and y_i is the 0-1 variable

associated with the location decision. Constraints (3.2) become

$$\underline{S}_{ip} y_i \leq \sum_{j \in N} x_{ijp} \leq \overline{S}_{ip} y_i \qquad i \in F,\ p \in P$$

For a fixed configuration of plants, the model reduces to the Fixed Change Multicommodity Flow Problem.

7. Stochastic Demand. It would be a valuable extension to relax the assumption of known demand and extend the modeling capability to deal with demand patterns which are stochastic. This is not an easy extension to achieve. It would appear that deterministic mathematical programming techniques which depend on fixed feasible solution spaces would be ruled out as possible solution methodologies.

8. Dynamic Multi-period Planning. Replacing the single-period fixed horizon model with a multi-period variable horizon model would be a realistic extension and would likely lead to improved decisions. The present model bases location/distribution decisions on annual data (for a particular reference year which may or may not typify all years in the planning horizon). This is inconsistent with the strategic nature of the decision. A multi-period model would take into account the cash out-flows, the supplies, and the demands, across the entire planning horizon (which could be ten to twenty years or more), and base decisions on the long-term average minimum system costs. A dynamic extension would have the effect of adding an additional subscript on all coefficients and on the x-variables. For example, x_{ijp} would become x_{ijps} where s indexes the period, or stage, and similarly, D_{jp} would become D_{jps}. There are no apparent technical difficulties with this extended problem; however, computational impracticalities may

arise since the constraint set is enlarged by a multiplicative factor equal to the number of periods.

LIST OF REFERENCES

LIST OF REFERENCES

1. Akinc, U. and B. M. Khumawala, "An Efficient Branch and Bound Algorithm for the Capacitated Warehouse Location Problem," Management Science, Vol. 23, 585-594 (1977).

2. Assad, A. A., "Multicommodity Network Flows--A Survey," Networks, Vol. 8, 37-91 (1978).

3. Balachandran, V. and S. Jain, "Optimal Facility Location Under Random Demand with General Cost Structure," Naval Research Logistics Quarterly, Vol. 23, 421-436 (1976).

4. Balinski, M. L., "On Finding Integer Solutions to Linear Programs," Mathematica, Princeton, NJ (1964).

5. Balinski, M. L. and P. Wolfe, "Nondifferentiable Optimization," Mathematical Programming Study, Vol. 3, North Holland Publishing Company (1975).

6. Bazaraa, M. S. and J. J. Jarvis, Linear Programming and Network Flows, John Wiley and Sons (1977).

7. Benders, J. F., "Partitioning Procedures for Solving Mixed-Variables Programming Problems," Numerische Mathematik, Vol. 4, 238-252 (1962).

8. Bilde, O. and J. Krarup, "Sharp Lower Bounds for the Simple Location Problem," Annals of Discrete Mathematics, Vol. 1, 79-97 (1977).

9. Bowersox, Donald J., O. K. Helferich, E. J. Marien, P. Gilmore, M. L. Lawrence, F. W. Morgan, and R. T. Rogers, "Dynamic Simulation of Physical Distribution Systems," East Lansing, Michigan: Michigan State University Business Studies, Division of Research (1972).

10. Bradley, G. H., G. G. Brown, and G. W. Graves, "Design and Implementation of Large Scale Primal Transshipment Algorithms," Management Science, Vol. 24, 1-33 (1977).

11. Chen, S. and C. G. DeWald, "A Generalized Chain Labeling Algorithm for Solving Multicommodity Flow Problems," Computers and Operations Research, Vol. 1, 437-465 (1974).

12. Chen, S. and R. Saigal, "A Primal Algorithm for Solving a Capacitated Network Flow Problem with Additional Linear Constraints," Networks, Vol. 7, 59-79 (1977).

13. Cooper, M. W., "A Survey of Methods for Pure Nonlinear Integer Programming," Management Science, Vol. 27, 353-361 (1981).

14. Cremeans, J. E., R. A. Smith, and G. R. Tyndall, "Optimal Multicommodity Network Flows with Resource Allocation," Naval Research Logistics Quarterly, Vol. 17, 269-280 (1970).

15. Cunningham, W. H., "A Network Simplex Method," Mathematical Programming, Vol. 11, 105-116 (1976).

16. Dantzig, G. B., and P. Wolfe, "Decomposition Principle for Linear Programs," Operations Research, Vol. 8, 101-111 (1960).

17. Dearing, P. M., and F. C. Newruck, "A Capacitated Bottleneck Facility Location Problem," Management Science, Vol. 11, 1093-1104 (1979).

18. Dial, R., F. Glover, D. Karney, and D. Klingman, "A Computational Analysis of Alternative Algorithms and Labeling Techniques for Finding Shortest Path Trees," Networks, Vol. 9, 215-248 (1979).

19. Efroymson, M. A., and T. L. Ray, "A Branch and Bound Algorithm for Plant Location," Operations Research, Vol. 14, 361-368 (1966).

20. Eilon, S., "Management Perspectives in Physical Distribution," OMEGA--International Journal of Management Science, Vol. 5, 437-462 (1977).

21. Erlenkotter, D., "A Dual-Based Procedure for Uncapacitated Facility Location," Operations Research, Vol. 26, 992-1009 (1978).

22. Escudero, L. F., "A Comparative Analysis of Linear Fitting for Nonlinear Functions of Optimization," European Journal of Operational Research, Vol. 2, 398-408 (1978).

23. Evans, J. R., "A Combinatorial Equivalence Between a Class of Multicommodity Flow Problems and the Capacitated Transportation Problem," Mathematical Programming, Vol. 10, 401-404 (1976).

24. Evans, J. R., "A Single-Commodity Transformation for Certain Multicommodity Networks," Operations Research, Vol. 26, 673-681 (1978).

25. Evans, J. R., "The Simplex Method for Integral Multicommodity Networks," Naval Research Logistics Quarterly, Vol. 25, 31-38 (1978).

26. Evans, J. R., and J. J. Jarvis, "Network Topology and Integral Multicommodity Flow Problems," Networks, Vol. 8, 107-120 (1978).

27. Falk, P. G., "Experiments in Mixed Integer Linear Programming in a Manufacturing System," OMEGA--International Journal of Management Science, Vol. 8, 473-484 (1980).

28. Ferland, J. A., "Minimum Cost Multicommodity Circulation Problem with Convex Arc Costs," Transportation Science, Vol. 8, 355-360 (1974).

29. Firth, D., F. R. Denham, K. R. Griffin, J. Heffernan, S. H. Press, N. C. Robson, and A. I. Saipe, Distribution Management Handbook, McGraw-Hill Ryerson, Toronto (1980).

30. Fisher, M. L., "The Lagrangean Relaxation Method for Solving Integer Programming Problems," Management Science, Vol. 27, 1-18 (1981).

31. Fisher, M. L., "A Large-Scale Mixed Integer Progamming System for Scheduling Industrial Gas Deliveries." Seminar delivered at The University of Tennessee, Knoxville, Management Science Seminar Series (October 1981).

32. Fisher, M. L., R. Jaikumar, and J. T. Lester, "A Computerized Vehicle Routing Application." Decision Sciences Working Paper 81-10-03, University of Pennsylvania (August 1981).

33. Fisher, M. L., W. D. Northup, and J. F. Shapiro, "Using Duality to Solve Discrete Optimization Problems: Theory and Computational Experience," Mathematical Programming Study, Vol. 3, 56-94, North Holland Publishing Company (1975).

34. Ford, L. R., and D. R. Fulkerson, "A Suggested Computation for Maximal Multicommodity Network Flow," Management Science, Vol. 5, 97-101 (1958).

35. Ford, L. R., and D. R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, New Jersey (1962).

36. Frank, M., and P. Wolfe, "An Algorithm for Quadratic Programming," Naval Research Logistics Quarterly, Vol. 3, 95-110 (1956).

37. Garfinkel, R. S., "Branch and Bound Methods for Integer Programming," in Combinatorial Optimization, N. Christofides, ed., 1-20, John Wiley and Sons, New York (1979).

38. Garfinkel, R. S., and G. L. Nemhauser, Integer Programming, John Wiley and Sons, New York (1972).

39. Garfinkel, R. S., and M. R. Rao, "The Bottleneck Transportation Problem," Naval Research Logistics Quarterly, Vol. 18, 465-472 (1971).

40. Geoffrion, A. M., "Integer Programming by Implicit Enumeration and Balas' Method," SIAM Review, Vol. 9, 178-190 (1967).

41. Geoffrion, A. M., "Primal Resource-Directive Approaches for Optimizing Nonlinear Decomposable Systems," Operations Research, Vol. 18, 375-403 (1970).

42. Geoffrion, A. M., "Generalized Benders Decomposition," Journal of Optimization Theory and Applications, Vol. 10, 237-260 (1972).

43. Geoffrion, A. M., "Lagrangean Relaxation for Integer Programming," Mathematical Programming Study, Vol. 2, 82-114, North Holland Publishing Company (1974).

44. Geoffrion, A. M., "A Guide to Computer-Assisted Methods for Distribution Systems Planning," Sloan Management Review, Vol. 16, 17-41 (1975).

45. Geoffrion, A. M., "The Purpose of Mathematical Programming is Insight, Not Numbers," Interfaces, Vol. 7, 81-92 (1976).

46. Geoffrion, A. M., "Guided Tour of Recent Practical Advances in Integer Linear Programming," OMEGA--International Journal of Management Science, Vol. 4, 49-57 (1976).

47. Geoffrion, A. M., "Better Distribution Planning with Computer Models," Harvard Business Review (July-August 1976).

48. Geoffrion, A. M., "Customer Aggregation in Distribution Modeling," Western Management Science Institute Working Paper No. 259, UCLA (October 1976).

49. Geoffrion, A. M., "Using an Auxiliary Model to Guide the Design of a Very Large Logistic Planning Model," Western Management Science Institute Working Paper No. 262, UCLA (January 1977).

50. Geoffrion, A. M., "Aggregation Theory and Its Application to Modeling in Mathematical Programming," Western Management Science Institute Working Paper No. 278, UCLA (December 1977).

51. Geoffrion, A. M., "A Priori Error Bounds for Procurement Commodity Aggregation in Logistics Planning Models," Naval Research Logistics Quarterly, Vol. 24, 201-212 (1977).

52. Geoffrion, A. M., "Objective-Function Approximations in Mathematical Programming," Mathematical Programming, Vol. 13, 23-37 (1977).

53. Geoffrion, A. M., "Making Better Use of Optimization Capability in Distribution System Planning," AIIE Transactions, Vol. 11, 96-108 (1979).

54. Geoffrion, A. M., private correspondence (July 18, 1979).

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

56. Geoffrion, A. M., G. W. Graves, and S. Lee, "Strategic Distribution System Planning: A Status Report," Chapter 7 in A. Hax (ed.), Studies in Operations Management, North Holland/American Elsavier (1978).

57. Geoffrion, A. M., G. W. Graves, and S. Lee, "A Portable Management Support System for Distribution Planning," unpublished paper, Graduate School of Management, UCLA (1979).

58. Geoffrion, A. M., and R. E. Marsten, "Integer Programming Algorithms: A Framework and State-of-the-Art Survey," Management Science, Vol. 18, 465-491 (1972).

59. Geoffrion, A. M., and R. McBride, "Lagrangean Relaxation Applied to the Capacitated Facility Location Problem," AIIE Transactions, Vol. 10, 40-47 (1978).

60. Geoffrion, A. M., and R. F. Powers, "Facility Location Analysis is Just the Beginning (If You Do It Right)," Interfaces, Vol. 10, 22-30 (1980).

61. Geoffrion, A. M., and T. J. Van Roy, "Caution: Common Sense Planning Can be Hazardous to Your Corporate Health," Sloan Management Review, Vol. 20, 31-42 (1979).

62. Gilmore, P. (ed.), Physical Distribution Management in Australia, 2nd ed., Longman-Cheshire (1979).

63. Gizelis, A. G., and Jesus-Emmanuel Samoulidis, "A Simplified Algorithm for the Depot Location Problem," OMEGA--International Journal of Management Science, Vol. 8, 465-472 (1980).

64. Glover, F., D. Karney, D. Klingman, and R. Russell, "Solving Singly Constrained Transshipment Problems," Transportation Science, Vol. 12, 277-297 (1978).

65. Glover, F., and D. Klingman, "Some Recent Practical Misconceptions about the State-of-the-Art of Network Algorithms," Operations Research, Vol. 2, 370-379 (1978).

66. Golden, B. L., "A Minimum-Cost Multicommodity Network Flow Problem Concerning Imports and Exports," Networks, Vol. 5, 331-356 (1975).

67. Golden, B. L., and T. L. Magnanti, "Deterministic Network Optimization: A Bibliography," Networks, Vol. 7, 149-183 (1977).

68. Graves, G. W., and T. J. Van Roy, "Decomposition for Large Scale Linear and Mixed Integer Linear Programming," accepted for publication by Mathematical Programming.

69. Grigoriadis, M. D., and W. W. White, "A Partitioning Algorithm for the Multicommodity Network Flow Problem," Mathematical Programming, Vol. 3, 157-177 (1972).

70. Guignard, M., "Fractional Vertices, Cuts, and Facets of the Simple Plant Location Problem," Mathematical Programming Study, Vol. 12, 150-162, North Holland Publishing Company (1980).

71. Hartman, J. K., and L. S. Lasdon, "A Generalized Upper Bounding Algorithm for Multicommodity Network Flow Problems," Networks, Vol. 1, 333-354 (1972).

72. Held, M., P. Wolfe, and H. D. Crowder, "Validation of Subgradient Optimization," Mathematical Programming, Vol. 6, 62-68 (1974).

73. Helgason, R. V., and J. L. Kennington, "An Efficient Procedure for Implementing a Dual-Simplex Network Flow Algorithm," AIIE Transactions, Vol. 9, 63-68 (1977).

74. Holloway, C. A., "An Extension of the Frank and Wolfe Method of Feasible Directions," Mathematical Programming, Vol. 6, 14-27 (1974).

75. Hu, T. C., "Multicommodity Network Flows," Operations Research, Vol. 11, 344-360 (1963).

76. Hu, T. C., "Minimum-Cost Flows in Convex-Cost Networks," Naval Research Logistics Quarterly, Vol. 13, 1-9 (1966).

77. Hughes, C. E., C. P. Pfleeger, and L. L. Rose, Advanced Programming Techniques, John Wiley and Sons, New York (1978).

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

79. Jarvis, J. J., "On the Equivalence Between the Node-Arc and Arc-Chain Formulations for the Multicommodity Maximal Flow Problem," Naval Research Logistics Quarterly, Vol. 15, 525-529 (1969).

80. Karkazis, J., and T. B. Boffey, "The Multi-commodity Facilities Location Problem," Journal of Operational Research Society, Vol. 32, 803-814 (1981).

81. Kaufman, L., M. V. Eede, and P. Hansen, "A Plant and Warehouse Location Problem," Operational Research Quarterly, Vol. 28, 547-554 (1977).

82. Kennington, J. L., "A Survey of Linear Cost Multicommodity Network Flows," Operations Research, Vol. 26, 209-236 (1978).

83. Kennington, J. L., "Solving Multicommodity Transportation Problems Using a Primal Partitioning Simplex Technique," Naval Research Logistics Quarterly, Vol. 24, 309-325 (1977).

84. Kennington, J. L., and R. V. Helgason, Algorithms for Network Programming, John Wiley and Sons, New York (1980).

85. Kennington, J. L., and M. Shalaby, "An Effective Subgradient Procedure for Minimal Cost Multicommodity Flow Problem," Management Science, Vol. 23, 994-1004 (1977).

86. Kennington, J. L., and E. Unger, "A New Branch and Bound Algorithm for the Fixed-Charge Transportation Problem," Management Science, Vol. 22, 1116-1126 (1976).

87. Khumawala, B. M., "An Efficient Branch and Bound Algorithm for the Warehouse Location Problem," Management Science, Vol. 18, B718-B731 (1972).

88. Khumawala, B. M., "An Efficient Heuristic Procedure for the Uncapacitated Warehouse Location Problem," Naval Research Logistics Quarterly, Vol. 20, 109-121 (1973).

89. Khumawala, B. M., and A. W. Neebe, "A Note on Warszawski's Multicommodity Location Problem," Journal of Operational Research Society, Vol. 29, 171-172 (1978).

90. Khumawala, B. M., and D. C. Whybark, "Solving the Dynamic Warehouse Location Problem," International Journal of Physical Distribution, Vol. 6, 238-251 (1976).

91. Klee, V., "Combinatorial Optimization: What is the State of the Art," Mathematics of Operations Research, Vol. 5, 1-26 (1980).

92. Kochman, G. A., and C. J. McCallum, Jr., "Facility Location Models for Planning a Trans Atlantic Communications Network," European Journal of Operations Research, Vol. 6, 205-211 (1981).

93. Koontz, H., C. O'Donnell, and H. Weihrich, Management, Seventh ed., McGraw-Hill Kogakusha, Tokyo (1980).

94. Kuehn, A. A., and M. J. Hamburger, "A Heuristic Program for Locating Warehouses," Management Science, Vol. 9, 643-666 (1963).

95. Land, A. H., and Doig, A. G., "An Automatic Method of Solving Discrete Programming Problems," Econometrica, Vol. 28, 497-520 (1960).

96. Laporte, G., and Y. Nobert, "An Exact Algorithm for Minimizing Routing and Operating Costs in Depot Location," European Journal of Operations Research, Vol. 6, 224-226 (1981).

97. Lasdon, L. S., Optimization Theory for Large Systems, Macmillan, New York (1970).

98. Lasdon, L. S., and A. D. Waren, "Survey of Nonlinear Programming Applications," Operations Research, Vol. 28, 1029-1073 (1980).

99. Le Blanc, L. J., "A Heuristic Approach for Large Scale Discrete Stochastic Transportation-Location Problems," Computers and Mathematics with Applications, Vol. 3, 87-94 (1977).

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

101. Markland, R. E., and R. J. Newett, "Production Distribution Planning in a Large Scale Commodity Processing Network," Decision Sciences, Vol. 7, 579-594 (1976).

102. Marsten, R. E., W. W. Hogan, and J. W. Blankenship, "The Boxstep Method for Large-Scale Optimization," Operations Research, Vol. 23, 389-405 (1975).

103. McDaniel, D., and M. Devine, "Modified Benders Partitioning Algorithm for Mixed Integer Programming," Management Science, Vol. 24, 312-319 (1977).

104. Meidan, A., "The Use of Quantitative Techniques in Warehouse Location," International Journal of Physical Distribution and Materials Management, Vol. 8, 347-358 (1978).

105. Morris, J. G., "Extent to Which Certain Fixed-Charge Depot Location Problems Can Be Solved by L. P.," Journal of the Operational Research Society, Vol. 29, 71-76 (1978).

106. Murphy, R. A., and R. E. Allen, "A Problem of Assigning Suppliers to Plants," AIIE Transactions, Vol. 11, 303-307 (1979).

107. Murtagh, B. A., "Large-Scale Linearly Constrained Optimization," Mathematical Programming, Vol. 14, 41-72 (1978).

108. Murtagh, B. A., Advanced Linear Programming: Computation and Practice, McGraw-Hill, New York (1981).

109. Nambiar, J. M., L. F. Gelders, and L. N. Van Wassenhove, "A Large Scale Location-Allocation Problem in the Natural Rubber Industry," European Journal of Operations Research, Vol. 6, 183-189 (1981).

110. National Council of Distribution Management, "Measuring Productivity in Physical Distribution--A $40 Billion Dollar Goldmine," a report prepared under contract by A. T. Kearney and Co., Vol. 1 (1978).

111. Nauss, R. M., "An Improved Algorithm for the Capacitated Facility Location Problem," Journal of the Operational Research Society, Vol. 29, 1195-1201 (1978).

112. Neebe, A. W., and B. M. Khumawala, "An Improved Algorithm for the Multi-commodity Location Problem," Journal of the Operational Research Society, Vol. 32, 143-169 (1981).

113. Negelhout, R. V., and G. L. Thompson, "A Cost Operator Approach to Multistage Location-Allocation," European Journal of Operational Research, Vol. 6, 149-161 (1981).

114. Orden, A., "The Transshipment Problem," Management Science, Vol. 2, 276-285 (1956).

115. Palacios-Gomez, F., L. Lasdon, and M. Engquist, "Nonlinear Optimization by Successive Linear Programming," Working paper 81/82-3-1, Department of General Business, The University of Texas at Austin.

116. Pape, U., "Implementation and Efficiency of Moore-Algorithms for the Shortest Route Problem," Mathematical Programming, Vol. 7, 212-222 (1974).

117. "PAPE Routine Listing," correspondence from D. Holmes and S. Cliff, Computer Concepts Corporation, Knoxville, TN (February 24, 1982).

118. Productivity Promotion Council of Australia, "Physical Distribution Management," a management assistance cost reduction pamphlet prepared in association with the Australian Department of Productivity.

119. Rardin, R. L., and R. E. Rosenthal, "Graph Structures Underlying Submatrices of Network Flow Tableaux," Transportation Science, Vol. 13, 98-112 (1979).

120. Rosenbaum, B. A., "Service Level Relationships in a Multi-echelon Inventory System," Management Science, Vol. 27, 926-945 (1981).

121. Rosenthal, R. E., "A Nonlinear Network Flow Algorithm for Maximization of Benefits in a Hydroelectric Power System," Operations Research, Vol. 29, 763-786 (1981).

122. Rothschield, B., and A. Whinston, "Feasibility of Two Commodity Network Flows," Operations Research, Vol. 14, 1121-1129 (1966).

123. Rothschield, B., and A. Whinston, "On Two Commodity Network Flows," Operations Research, Vol. 14, 377-387 (1966).

124. Sakarovitch, M., "Two Commodity Network Flows and Linear Programming," Mathematical Programming, Vol. 4, 1-20 (1973).

125. Sawdy, L. C. W., The Economics of Distribution, Halsted Press, John Wiley and Sons, New York (1972).

126. Senju, S., and Y. Toyoda, "An Approach to Linear Programming with 0-1 Variables," Management Science, Vol. 15, B196-B207 (1968).

127. Spielberg, K., "Algorithms for the Simple Plant-Location Problem with Some Side Constraints," Operations Research, Vol. 17, 85-111 (1969).

128. Stoker, R. B., "Determining the Optimal Number of Depots in a Physical Distribution System According to Market Characteristics," European Journal of Operational Research, Vol. 4, 107-117 (1980).

129. Sweeney, D. J., and R. A. Murphy, "A Method of Decomposition for Integer Programs," Operations Research, Vol. 27, 1128-1141 (1979).

130. Swoveland, C., "A Two-Stage Decomposition Algorithm for a Generalized Multicommodity Flow Problem," INFOR, Vol. 11, 232-244 (1973).

131. Tomlin, J. A., "An Improved Branch and Bound Method for Integer Programming," Operations Research, Vol. 19, 1070-1075 (1971).

132. Warszawski, A., "Multi-dimensional Location Problems," Operational Research Quarterly, Vol. 24, 165-179 (1973).

133. Wollmer, R. D., "Multicommodity Networks with Resource Constraints: The Generalized Multicommodity Flow Problem," Networks, Vol. 1, 245-263 (1972).

134. Van Roy, T. J., and L. F. Gelders, "Solving a Distribution Problem with Side Constraints," European Journal of Operational Research, Vol. 6, 61-66 (1981).

135. Zipkin, P. H., "Bounds for Aggregating Nodes in Network Problems," Mathematical Programming, Vol. 19, 155-177 (1980).

APPENDIX

AUSTRALIAN CASE STUDY DATA

NODE 1	PLANT DESCRIPTION	BRISBANE					
PRODUCT GROUP	1	LOWER BOUND	70	UPPER BOUND	170		
PRODUCT GROUP	2	LOWER BOUND	500	UPPER BOUND	1280		
PRODUCT GROUP	3	LOWER BOUND	20	UPPER BOUND	170		
PRODUCT GROUP	4	LOWER BOUND	100	UPPER BOUND	300		
PRODUCT GROUP	5	LOWER BOUND	10	UPPER BOUND	80		
NODE 2	PLANT DESCRIPTION	TOWNSVILLE					
PRODUCT GROUP	1	LOWER BOUND	10	UPPER BOUND	60		
PRODUCT GROUP	2	LOWER BOUND	100	UPPER BOUND	450		
PRODUCT GROUP	3	LOWER BOUND	10	UPPER BOUND	60		
PRODUCT GROUP	4	LOWER BOUND	40	UPPER BOUND	110		
PRODUCT GROUP	5	LOWER BOUND	0	UPPER BOUND	30		
NODE 3	PLANT DESCRIPTION	GRAFTON					
PRODUCT GROUP	1	LOWER BOUND	0	UPPER BOUND	40		
PRODUCT GROUP	2	LOWER BOUND	100	UPPER BOUND	320		
PRODUCT GROUP	3	LOWER BOUND	0	UPPER BOUND	40		
PRODUCT GROUP	4	LOWER BOUND	30	UPPER BOUND	80		
PRODUCT GROUP	5	LOWER BOUND	0	UPPER BOUND	20		
NODE 4	PLANT DESCRIPTION	PRESTON					
PRODUCT GROUP	1	LOWER BOUND	70	UPPER BOUND	350		
PRODUCT GROUP	2	LOWER BOUND	700	UPPER BOUND	3000		
PRODUCT GROUP	3	LOWER BOUND	80	UPPER BOUND	360		
PRODUCT GROUP	4	LOWER BOUND	200	UPPER BOUND	660		
PRODUCT GROUP	5	LOWER BOUND	30	UPPER BOUND	150		
NODE 5	PLANT DESCRIPTION	MELBOURNE					
PRODUCT GROUP	1	LOWER BOUND	70	UPPER BOUND	350		
PRODUCT GROUP	2	LOWER BOUND	700	UPPER BOUND	3000		
PRODUCT GROUP	3	LOWER BOUND	80	UPPER BOUND	360		
PRODUCT GROUP	4	LOWER BOUND	200	UPPER BOUND	660		
PRODUCT GROUP	5	LOWER BOUND	30	UPPER BOUND	150		
NODE 6	PLANT DESCRIPTION	TASMANIA					
PRODUCT GROUP	1	LOWER BOUND	0	UPPER BOUND	40		
PRODUCT GROUP	2	LOWER BOUND	100	UPPER BOUND	320		
PRODUCT GROUP	3	LOWER BOUND	0	UPPER BOUND	60		
PRODUCT GROUP	4	LOWER BOUND	30	UPPER BOUND	80		
PRODUCT GROUP	5	LOWER BOUND	0	UPPER BOUND	30		
NODE 7	PLANT DESCRIPTION	PERTH					
PRODUCT GROUP	1	LOWER BOUND	0	UPPER BOUND	70		
PRODUCT GROUP	2	LOWER BOUND	100	UPPER BOUND	350		
PRODUCT GROUP	3	LOWER BOUND	0	UPPER BOUND	60		
PRODUCT GROUP	4	LOWER BOUND	30	UPPER BOUND	90		
PRODUCT GROUP	5	LOWER BOUND	0	UPPER BOUND	20		
NODE 94	CUSTOMER ZONE	NORTH BRISBANE		REGION	1		
PRODUCT GROUP	1	DEMAND	14				
PRODUCT GROUP	2	DEMAND	115				
PRODUCT GROUP	3	DEMAND	26				
PRODUCT GROUP	4	DEMAND	32				
PRODUCT GROUP	5	DEMAND	7				
NODE 95	CUSTOMER ZONE	EAST BRISBANE		REGION	1		
PRODUCT GROUP	1	DEMAND	14				
PRODUCT GROUP	2	DEMAND	115				
PRODUCT GROUP	3	DEMAND	26				
PRODUCT GROUP	4	DEMAND	32				
PRODUCT GROUP	5	DEMAND	7				
NODE 96	CUSTOMER ZONE	SOUTH BRISBANE		REGION	1		
PRODUCT GROUP	1	DEMAND	14				
PRODUCT GROUP	2	DEMAND	115				
PRODUCT GROUP	3	DEMAND	26				
PRODUCT GROUP	4	DEMAND	32				

```
   PRODUCT GROUP      5   DEMAND            79
NODE     97   CUSTOMER ZONE      WEST BRISBANE          REGION      1
   PRODUCT GROUP      1   DEMAND            14
   PRODUCT GROUP      2   DEMAND           115
   PRODUCT GROUP      3   DEMAND            26
   PRODUCT GROUP      4   DEMAND            32
   PRODUCT GROUP      5   DEMAND             7
NODE     98   CUSTOMER ZONE      TOOWCOMBA              REGION      1
   PRODUCT GROUP      1   DEMAND             8
   PRODUCT GROUP      2   DEMAND            73
   PRODUCT GROUP      3   DEMAND            19
   PRODUCT GROUP      4   DEMAND            20
   PRODUCT GROUP      5   DEMAND            37
NODE     99   CUSTOMER ZONE      TWEED HEADS            REGION      1
   PRODUCT GROUP      1   DEMAND            24
   PRODUCT GROUP      2   DEMAND            25
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND            23
   PRODUCT GROUP      5   DEMAND             1
NODE    100   CUSTOMER ZONE      NAMBOUR                REGION      1
   PRODUCT GROUP      1   DEMAND            21
   PRODUCT GROUP      2   DEMAND            65
   PRODUCT GROUP      3   DEMAND            10
   PRODUCT GROUP      4   DEMAND            17
   PRODUCT GROUP      5   DEMAND             2
NODE    101   CUSTOMER ZONE      BUNDABERG              REGION      1
   PRODUCT GROUP      1   DEMAND             3
   PRODUCT GROUP      2   DEMAND            62
   PRODUCT GROUP      3   DEMAND            11
   PRODUCT GROUP      4   DEMAND            11
   PRODUCT GROUP      5   DEMAND             2
NODE    102   CUSTOMER ZONE      DALBY                  REGION      1
   PRODUCT GROUP      1   DEMAND             1
   PRODUCT GROUP      2   DEMAND            27
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             5
   PRODUCT GROUP      5   DEMAND             0
NODE    103   CUSTOMER ZONE      MARYBOROUGH            REGION      1
   PRODUCT GROUP      1   DEMAND             4
   PRODUCT GROUP      2   DEMAND            31
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             6
   PRODUCT GROUP      5   DEMAND             1
NODE    104   CUSTOMER ZONE      ST GEORGE              REGION      1
   PRODUCT GROUP      1   DEMAND             0
   PRODUCT GROUP      2   DEMAND             3
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             1
   PRODUCT GROUP      5   DEMAND             0
NODE    105   CUSTOMER ZONE      GOONDIWINDI            REGION      1
   PRODUCT GROUP      1   DEMAND             1
   PRODUCT GROUP      2   DEMAND            20
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             5
   PRODUCT GROUP      5   DEMAND             0
NODE    106   CUSTOMER ZONE      TOWNSVILLE             REGION      2
   PRODUCT GROUP      1   DEMAND             8
   PRODUCT GROUP      2   DEMAND            61
   PRODUCT GROUP      3   DEMAND            11
   PRODUCT GROUP      4   DEMAND            17
```

```
   PRODUCT GROUP      5   DEMAND            19
NODE   107   CUSTOMER ZONE      ROCKHAMPTON          REGION      2
   PRODUCT GROUP      1   DEMAND             9
   PRODUCT GROUP      2   DEMAND           176
   PRODUCT GROUP      3   DEMAND             6
   PRODUCT GROUP      4   DEMAND            23
   PRODUCT GROUP      5   DEMAND             5
NODE   108   CUSTOMER ZONE      MACKAY               REGION      2
   PRODUCT GROUP      1   DEMAND            12
   PRODUCT GROUP      2   DEMAND           101
   PRODUCT GROUP      3   DEMAND             9
   PRODUCT GROUP      4   DEMAND            12
   PRODUCT GROUP      5   DEMAND             2
NODE   109   CUSTOMER ZONE      MT ISA               REGION      3
   PRODUCT GROUP      1   DEMAND             3
   PRODUCT GROUP      2   DEMAND            21
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             4
   PRODUCT GROUP      5   DEMAND             0
NODE   110   CUSTOMER ZONE      CHARLEVILLE          REGION      3
   PRODUCT GROUP      1   DEMAND             0
   PRODUCT GROUP      2   DEMAND            14
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             2
   PRODUCT GROUP      5   DEMAND             0
NODE   111   CUSTOMER ZONE      CUNNAMULLA           REGION      3
   PRODUCT GROUP      1   DEMAND             0
   PRODUCT GROUP      2   DEMAND             7
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             1
   PRODUCT GROUP      5   DEMAND             0
NODE   112   CUSTOMER ZONE      LONGREACH            REGION      3
   PRODUCT GROUP      1   DEMAND             1
   PRODUCT GROUP      2   DEMAND            26
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             3
   PRODUCT GROUP      5   DEMAND             0
NODE   113   CUSTOMER ZONE      EMERALD              REGION      3
   PRODUCT GROUP      1   DEMAND             2
   PRODUCT GROUP      2   DEMAND            26
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             4
   PRODUCT GROUP      5   DEMAND             0
NODE   114   CUSTOMER ZONE      ROMA                 REGION      3
   PRODUCT GROUP      1   DEMAND             0
   PRODUCT GROUP      2   DEMAND            17
   PRODUCT GROUP      3   DEMAND             0
   PRODUCT GROUP      4   DEMAND             3
   PRODUCT GROUP      5   DEMAND             0
NODE   115   CUSTOMER ZONE      CAIRNS               REGION      4
   PRODUCT GROUP      1   DEMAND             5
   PRODUCT GROUP      2   DEMAND            93
   PRODUCT GROUP      3   DEMAND             2
   PRODUCT GROUP      4   DEMAND            12
   PRODUCT GROUP      5   DEMAND             1
NODE   116   CUSTOMER ZONE      DARWIN               REGION      4
   PRODUCT GROUP      1   DEMAND             5
   PRODUCT GROUP      2   DEMAND            84
   PRODUCT GROUP      3   DEMAND             2
   PRODUCT GROUP      4   DEMAND            11
```

```
   PRODUCT GROUP      5   DEMAND              1
NODE   117   CUSTOMER ZONE      LISMORE                    REGION      5
   PRODUCT GROUP      1   DEMAND              1
   PRODUCT GROUP      2   DEMAND             19
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              4
   PRODUCT GROUP      5   DEMAND              0
NODE   118   CUSTOMER ZONE      GRAFTON                    REGION      5
   PRODUCT GROUP      1   DEMAND              3
   PRODUCT GROUP      2   DEMAND             20
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              4
   PRODUCT GROUP      5   DEMAND              0
NODE   119   CUSTOMER ZONE      ARMIDALE                   REGION      5
   PRODUCT GROUP      1   DEMAND              3
   PRODUCT GROUP      2   DEMAND             19
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              3
   PRODUCT GROUP      5   DEMAND              0
NODE   120   CUSTOMER ZONE      TAMWORTH                   REGION      5
   PRODUCT GROUP      1   DEMAND              4
   PRODUCT GROUP      2   DEMAND             28
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              5
   PRODUCT GROUP      5   DEMAND              0
NODE   121   CUSTOMER ZONE      MURWILLUMBAH               REGION      5
   PRODUCT GROUP      1   DEMAND              1
   PRODUCT GROUP      2   DEMAND             19
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              3
   PRODUCT GROUP      5   DEMAND              0
NODE   122   CUSTOMER ZONE      PORT MACQUARIE             REGION      5
   PRODUCT GROUP      1   DEMAND              1
   PRODUCT GROUP      2   DEMAND             19
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              3
   PRODUCT GROUP      5   DEMAND              0
NODE   123   CUSTOMER ZONE      MOREE                      REGION      5
   PRODUCT GROUP      1   DEMAND              1
   PRODUCT GROUP      2   DEMAND             21
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              3
   PRODUCT GROUP      5   DEMAND              0
NODE   124   CUSTOMER ZONE      NEWCASTLE                  REGION      6
   PRODUCT GROUP      1   DEMAND             10
   PRODUCT GROUP      2   DEMAND             74
   PRODUCT GROUP      3   DEMAND             11
   PRODUCT GROUP      4   DEMAND             21
   PRODUCT GROUP      5   DEMAND              3
NODE   125   CUSTOMER ZONE      BATHURST                   REGION      6
   PRODUCT GROUP      1   DEMAND              3
   PRODUCT GROUP      2   DEMAND             23
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              4
   PRODUCT GROUP      5   DEMAND              0
NODE   126   CUSTOMER ZONE      WOOLONGONG                 REGION      6
   PRODUCT GROUP      1   DEMAND             11
   PRODUCT GROUP      2   DEMAND             83
   PRODUCT GROUP      3   DEMAND             12
   PRODUCT GROUP      4   DEMAND             23
```

```
   PRODUCT GROUP     5   DEMAND          3
NODE   127   CUSTOMER ZONE     PARAMATTA          REGION     6
   PRODUCT GROUP     1   DEMAND          6
   PRODUCT GROUP     2   DEMAND         43
   PRODUCT GROUP     3   DEMAND          6
   PRODUCT GROUP     4   DEMAND         12
   PRODUCT GROUP     5   DEMAND          1
NODE   128   CUSTOMER ZONE     NORTH SYDNEY       REGION     6
   PRODUCT GROUP     1   DEMAND         49
   PRODUCT GROUP     2   DEMAND        332
   PRODUCT GROUP     3   DEMAND         51
   PRODUCT GROUP     4   DEMAND         94
   PRODUCT GROUP     5   DEMAND         14
NODE   129   CUSTOMER ZONE     EAST SYDNEY        REGION     6
   PRODUCT GROUP     1   DEMAND         49
   PRODUCT GROUP     2   DEMAND        332
   PRODUCT GROUP     3   DEMAND         51
   PRODUCT GROUP     4   DEMAND         94
   PRODUCT GROUP     5   DEMAND         14
NODE   130   CUSTOMER ZONE     WEST SYDNEY        REGION     6
   PRODUCT GROUP     1   DEMAND         49
   PRODUCT GROUP     2   DEMAND        332
   PRODUCT GROUP     3   DEMAND         51
   PRODUCT GROUP     4   DEMAND         94
   PRODUCT GROUP     5   DEMAND         14
NODE   131   CUSTOMER ZONE     SOUTH SYDNEY       REGION     6
   PRODUCT GROUP     1   DEMAND         49
   PRODUCT GROUP     2   DEMAND        332
   PRODUCT GROUP     3   DEMAND         51
   PRODUCT GROUP     4   DEMAND         94
   PRODUCT GROUP     5   DEMAND         14
NODE   132   CUSTOMER ZONE     LIVERPOOL          REGION     6
   PRODUCT GROUP     1   DEMAND          6
   PRODUCT GROUP     2   DEMAND        102
   PRODUCT GROUP     3   DEMAND          2
   PRODUCT GROUP     4   DEMAND         13
   PRODUCT GROUP     5   DEMAND          1
NODE   133   CUSTOMER ZONE     PENRITH            REGION     6
   PRODUCT GROUP     1   DEMAND          5
   PRODUCT GROUP     2   DEMAND         87
   PRODUCT GROUP     3   DEMAND          2
   PRODUCT GROUP     4   DEMAND         11
   PRODUCT GROUP     5   DEMAND          1
NODE   134   CUSTOMER ZONE     GOULBURN           REGION     6
   PRODUCT GROUP     1   DEMAND          4
   PRODUCT GROUP     2   DEMAND         27
   PRODUCT GROUP     3   DEMAND          0
   PRODUCT GROUP     4   DEMAND          5
   PRODUCT GROUP     5   DEMAND          0
NODE   135   CUSTOMER ZONE     ORANGE             REGION     6
   PRODUCT GROUP     1   DEMAND          4
   PRODUCT GROUP     2   DEMAND         29
   PRODUCT GROUP     3   DEMAND          0
   PRODUCT GROUP     4   DEMAND          5
   PRODUCT GROUP     5   DEMAND          0
NODE   136   CUSTOMER ZONE     BROKEN HILL        REGION     7
   PRODUCT GROUP     1   DEMAND          5
   PRODUCT GROUP     2   DEMAND         95
   PRODUCT GROUP     3   DEMAND          2
   PRODUCT GROUP     4   DEMAND         12
```

```
   PRODUCT GROUP      5   DEMAND              1
NODE   137   CUSTOMER ZONE      DUBBO                    REGION      7
   PRODUCT GROUP      1   DEMAND              3
   PRODUCT GROUP      2   DEMAND             20
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              3
   PRODUCT GROUP      5   DEMAND              0
NODE   138   CUSTOMER ZONE      ALBURY                   REGION      8
   PRODUCT GROUP      1   DEMAND              4
   PRODUCT GROUP      2   DEMAND             76
   PRODUCT GROUP      3   DEMAND              2
   PRODUCT GROUP      4   DEMAND             10
   PRODUCT GROUP      5   DEMAND              1
NODE   139   CUSTOMER ZONE      CANBERRA                 REGION      8
   PRODUCT GROUP      1   DEMAND             10
   PRODUCT GROUP      2   DEMAND             74
   PRODUCT GROUP      3   DEMAND             11
   PRODUCT GROUP      4   DEMAND             21
   PRODUCT GROUP      5   DEMAND              3
NODE   140   CUSTOMER ZONE      GEELONG                  REGION      9
   PRODUCT GROUP      1   DEMAND              7
   PRODUCT GROUP      2   DEMAND             56
   PRODUCT GROUP      3   DEMAND              8
   PRODUCT GROUP      4   DEMAND             15
   PRODUCT GROUP      5   DEMAND              2
NODE   141   CUSTOMER ZONE      NORTH MELBOURNE          REGION      9
   PRODUCT GROUP      1   DEMAND             37
   PRODUCT GROUP      2   DEMAND            270
   PRODUCT GROUP      3   DEMAND             41
   PRODUCT GROUP      4   DEMAND             76
   PRODUCT GROUP      5   DEMAND             11
NODE   142   CUSTOMER ZONE      EAST MELBOURNE           REGION      9
   PRODUCT GROUP      1   DEMAND             37
   PRODUCT GROUP      2   DEMAND            270
   PRODUCT GROUP      3   DEMAND             41
   PRODUCT GROUP      4   DEMAND             76
   PRODUCT GROUP      5   DEMAND             11
NODE   143   CUSTOMER ZONE      WEST MELBOURNE           REGION      9
   PRODUCT GROUP      1   DEMAND             37
   PRODUCT GROUP      2   DEMAND            270
   PRODUCT GROUP      3   DEMAND             41
   PRODUCT GROUP      4   DEMAND             76
   PRODUCT GROUP      5   DEMAND             11
NODE   144   CUSTOMER ZONE      SOUTH MELBOURNE          REGION      9
   PRODUCT GROUP      1   DEMAND             37
   PRODUCT GROUP      2   DEMAND            270
   PRODUCT GROUP      3   DEMAND             41
   PRODUCT GROUP      4   DEMAND             76
   PRODUCT GROUP      5   DEMAND             11
NODE   145   CUSTOMER ZONE      LILYDALE                 REGION      9
   PRODUCT GROUP      1   DEMAND              2
   PRODUCT GROUP      2   DEMAND             16
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              3
   PRODUCT GROUP      5   DEMAND              0
NODE   146   CUSTOMER ZONE      WARRNAMBOOL              REGION      9
   PRODUCT GROUP      1   DEMAND              3
   PRODUCT GROUP      2   DEMAND             22
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              4
```

```
   PRODUCT GROUP      5   DEMAND            0
NODE   147   CUSTOMER ZONE      MORWELL              REGION      9
   PRODUCT GROUP      1   DEMAND            3
   PRODUCT GROUP      2   DEMAND           20
   PRODUCT GROUP      3   DEMAND            0
   PRODUCT GROUP      4   DEMAND            4
   PRODUCT GROUP      5   DEMAND            0
NODE   148   CUSTOMER ZONE      MOE                  REGION      9
   PRODUCT GROUP      1   DEMAND            3
   PRODUCT GROUP      2   DEMAND           21
   PRODUCT GROUP      3   DEMAND            0
   PRODUCT GROUP      4   DEMAND            4
   PRODUCT GROUP      5   DEMAND            0
NODE   149   CUSTOMER ZONE      BALLARAT             REGION     10
   PRODUCT GROUP      1   DEMAND            8
   PRODUCT GROUP      2   DEMAND           61
   PRODUCT GROUP      3   DEMAND           11
   PRODUCT GROUP      4   DEMAND           17
   PRODUCT GROUP      5   DEMAND           19
NODE   150   CUSTOMER ZONE      BENDIGO              REGION     10
   PRODUCT GROUP      1   DEMAND            5
   PRODUCT GROUP      2   DEMAND           98
   PRODUCT GROUP      3   DEMAND            2
   PRODUCT GROUP      4   DEMAND           13
   PRODUCT GROUP      5   DEMAND            1
NODE   151   CUSTOMER ZONE      SHEPPARTON           REGION     10
   PRODUCT GROUP      1   DEMAND            3
   PRODUCT GROUP      2   DEMAND           22
   PRODUCT GROUP      3   DEMAND            0
   PRODUCT GROUP      4   DEMAND            4
   PRODUCT GROUP      5   DEMAND            0
NODE   152   CUSTOMER ZONE      ADELAIDE             REGION     11
   PRODUCT GROUP      1   DEMAND           55
   PRODUCT GROUP      2   DEMAND          399
   PRODUCT GROUP      3   DEMAND           61
   PRODUCT GROUP      4   DEMAND          112
   PRODUCT GROUP      5   DEMAND           17
NODE   153   CUSTOMER ZONE      WHYALLA              REGION     11
   PRODUCT GROUP      1   DEMAND            4
   PRODUCT GROUP      2   DEMAND           28
   PRODUCT GROUP      3   DEMAND            0
   PRODUCT GROUP      4   DEMAND            5
   PRODUCT GROUP      5   DEMAND            0
NODE   154   CUSTOMER ZONE      MT GAMBIER           REGION     11
   PRODUCT GROUP      1   DEMAND            3
   PRODUCT GROUP      2   DEMAND           22
   PRODUCT GROUP      3   DEMAND            0
   PRODUCT GROUP      4   DEMAND            4
   PRODUCT GROUP      5   DEMAND            0
NODE   155   CUSTOMER ZONE      ELIZABETH            REGION     11
   PRODUCT GROUP      1   DEMAND            6
   PRODUCT GROUP      2   DEMAND          105
   PRODUCT GROUP      3   DEMAND            2
   PRODUCT GROUP      4   DEMAND           14
   PRODUCT GROUP      5   DEMAND            1
NODE   156   CUSTOMER ZONE      PORT PIRIE           REGION     11
   PRODUCT GROUP      1   DEMAND            3
   PRODUCT GROUP      2   DEMAND           20
   PRODUCT GROUP      3   DEMAND            0
   PRODUCT GROUP      4   DEMAND            3
```

```
   PRODUCT GROUP      5   DEMAND              0
NODE   157   CUSTOMER ZONE      PORT AUGUSTA          REGION     11
   PRODUCT GROUP      1   DEMAND              2
   PRODUCT GROUP      2   DEMAND             13
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              2
   PRODUCT GROUP      5   DEMAND              0
NODE   158   CUSTOMER ZONE      PORT LINCOLN          REGION     11
   PRODUCT GROUP      1   DEMAND              1
   PRODUCT GROUP      2   DEMAND             24
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              4
   PRODUCT GROUP      5   DEMAND              0
NODE   159   CUSTOMER ZONE      PERTH                 REGION     12
   PRODUCT GROUP      1   DEMAND             35
   PRODUCT GROUP      2   DEMAND            256
   PRODUCT GROUP      3   DEMAND             39
   PRODUCT GROUP      4   DEMAND             72
   PRODUCT GROUP      5   DEMAND             11
NODE   160   CUSTOMER ZONE      KALGOORLIE            REGION     12
   PRODUCT GROUP      1   DEMAND             17
   PRODUCT GROUP      2   DEMAND             28
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              5
   PRODUCT GROUP      5   DEMAND              0
NODE   161   CUSTOMER ZONE      BUNBURY               REGION     12
   PRODUCT GROUP      1   DEMAND              3
   PRODUCT GROUP      2   DEMAND             20
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              3
   PRODUCT GROUP      5   DEMAND              0
NODE   162   CUSTOMER ZONE      ALBANY                REGION     12
   PRODUCT GROUP      1   DEMAND              2
   PRODUCT GROUP      2   DEMAND             14
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              2
   PRODUCT GROUP      5   DEMAND              0
NODE   163   CUSTOMER ZONE      HOBART                REGION     13
   PRODUCT GROUP      1   DEMAND              8
   PRODUCT GROUP      2   DEMAND             63
   PRODUCT GROUP      3   DEMAND             39
   PRODUCT GROUP      4   DEMAND             17
   PRODUCT GROUP      5   DEMAND              2
NODE   164   CUSTOMER ZONE      NEW NORFOLK           REGION     13
   PRODUCT GROUP      1   DEMAND              0
   PRODUCT GROUP      2   DEMAND             15
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              2
   PRODUCT GROUP      5   DEMAND              0
NODE   165   CUSTOMER ZONE      LAUNCESTON            REGION     14
   PRODUCT GROUP      1   DEMAND              8
   PRODUCT GROUP      2   DEMAND             64
   PRODUCT GROUP      3   DEMAND             11
   PRODUCT GROUP      4   DEMAND             17
   PRODUCT GROUP      5   DEMAND             20
NODE   166   CUSTOMER ZONE      BURNIE                REGION     14
   PRODUCT GROUP      1   DEMAND              3
   PRODUCT GROUP      2   DEMAND             19
   PRODUCT GROUP      3   DEMAND              0
   PRODUCT GROUP      4   DEMAND              3
```

```
   PRODUCT GROUP      5   DEMAND            0
NODE   167   CUSTOMER ZONE      DEVONPORT              REGION     14
   PRODUCT GROUP      1   DEMAND            2
   PRODUCT GROUP      2   DEMAND           18
   PRODUCT GROUP      3   DEMAND            0
   PRODUCT GROUP      4   DEMAND            3
   PRODUCT GROUP      5   DEMAND            0
NODE   168   CUSTOMER ZONE      DUMMY                  REGION     15
   PRODUCT GROUP      1   DEMAND          302
   PRODUCT GROUP      2   DEMAND         2618
   PRODUCT GROUP      3   DEMAND          347
   PRODUCT GROUP      4   DEMAND          495
   PRODUCT GROUP      5   DEMAND          122
NODE     8   CANDIDATE DC      BRISBANE
   LOWER BOUND THROUGHPUT          350      UPPER BOUND THROUGHPUT        1200
   FIXED COST      1064000       VARIABLE CAPACITY COST          430
NODE    10   CANDIDATE DC      TOOWOOMBA
   LOWER BOUND THROUGHPUT          150      UPPER BOUND THROUGHPUT         600
   FIXED COST       517000       VARIABLE CAPACITY COST          590
NODE    12   CANDIDATE DC      TWEED HEADS
   LOWER BOUND THROUGHPUT          150      UPPER BOUND THROUGHPUT         750
   FIXED COST       680000       VARIABLE CAPACITY COST          590
NODE    14   CANDIDATE DC      NAMBOUR
   LOWER BOUND THROUGHPUT          150      UPPER BOUND THROUGHPUT         600
   FIXED COST       490000       VARIABLE CAPACITY COST          590
NODE    16   CANDIDATE DC      BUNDABERG
   LOWER BOUND THROUGHPUT          150      UPPER BOUND THROUGHPUT         600
   FIXED COST       545000       VARIABLE CAPACITY COST          590
NODE    18   CANDIDATE DC      TOWNSVILLE
   LOWER BOUND THROUGHPUT          120      UPPER BOUND THROUGHPUT         600
   FIXED COST       517000       VARIABLE CAPACITY COST          590
NODE    20   CANDIDATE DC      ROCKHAMPTON
   LOWER BOUND THROUGHPUT           50      UPPER BOUND THROUGHPUT         350
   FIXED COST       300000       VARIABLE CAPACITY COST          630
NODE    22   CANDIDATE DC      MACKAY
   LOWER BOUND THROUGHPUT           50      UPPER BOUND THROUGHPUT         350
   FIXED COST       300000       VARIABLE CAPACITY COST          630
NODE    24   CANDIDATE DC      MT ISA
   LOWER BOUND THROUGHPUT           35      UPPER BOUND THROUGHPUT         350
   FIXED COST       315000       VARIABLE CAPACITY COST          650
NODE    26   CANDIDATE DC      CAIRNS
   LOWER BOUND THROUGHPUT           50      UPPER BOUND THROUGHPUT         350
   FIXED COST       315000       VARIABLE CAPACITY COST          650
NODE    28   CANDIDATE DC      DARWIN
   LOWER BOUND THROUGHPUT           50      UPPER BOUND THROUGHPUT         350
   FIXED COST       320000       VARIABLE CAPACITY COST          670
NODE    30   CANDIDATE DC      LISMORE
   LOWER BOUND THROUGHPUT           25      UPPER BOUND THROUGHPUT         350
   FIXED COST       315000       VARIABLE CAPACITY COST          650
NODE    32   CANDIDATE DC      GRAFTON
   LOWER BOUND THROUGHPUT           25      UPPER BOUND THROUGHPUT         350
   FIXED COST       315000       VARIABLE CAPACITY COST          650
NODE    34   CANDIDATE DC      ARMIDALE
   LOWER BOUND THROUGHPUT           25      UPPER BOUND THROUGHPUT         350
   FIXED COST       315000       VARIABLE CAPACITY COST          650
NODE    36   CANDIDATE DC      TAMWORTH
   LOWER BOUND THROUGHPUT           25      UPPER BOUND THROUGHPUT         350
   FIXED COST       320000       VARIABLE CAPACITY COST          600
NODE    38   CANDIDATE DC      NEWCASTLE
   LOWER BOUND THROUGHPUT          250      UPPER BOUND THROUGHPUT         750
```

```
   FIXED COST          680000
NODE    40   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          315000
NODE    42   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          880000
NODE    44   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          880000
NODE    46   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST         1675000
NODE    48   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST         1675000
NODE    50   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST         1675000
NODE    52   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST         1675000
NODE    54   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          315000
NODE    56   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          315000
NODE    58   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          800000
NODE    60   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          800000
NODE    62   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          800000
NODE    64   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST         1675000
NODE    66   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST         1675000
NODE    68   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST         1675000
NODE    70   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST         1675000
NODE    72   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          800000
NODE    74   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST          800000
NODE    76   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST         1064000
NODE    78   CANDIDATE DC
   LOWER BOUND THROUGHPUT
 VARIABLE CAPACITY COST          590
BATHURST
     50      UPPER BOUND THROUGHPUT          350
 VARIABLE CAPACITY COST          630
WOLLONGONG
    250      UPPER BOUND THROUGHPUT          800
 VARIABLE CAPACITY COST          590
PARRAMATTA
    250      UPPER BOUND THROUGHPUT          800
 VARIABLE CAPACITY COST          590
NORTH SYDNEY
    700      UPPER BOUND THROUGHPUT         1800
 VARIABLE CAPACITY COST          400
EAST SYDNEY
    700      UPPER BOUND THROUGHPUT         1800
 VARIABLE CAPACITY COST          400
WEST SYDNEY
    700      UPPER BOUND THROUGHPUT         1800
 VARIABLE CAPACITY COST          400
SOUTH SYDNEY
    700      UPPER BOUND THROUGHPUT         1800
 VARIABLE CAPACITY COST          400
BROKEN HILL
     35      UPPER BOUND THROUGHPUT          350
 VARIABLE CAPACITY COST          630
DUBBO
     35      UPPER BOUND THROUGHPUT          350
 VARIABLE CAPACITY COST          630
ALBURY
     60      UPPER BOUND THROUGHPUT          800
 VARIABLE CAPACITY COST          590
CANBERRA
     60      UPPER BOUND THROUGHPUT          800
 VARIABLE CAPACITY COST          590
GEELONG
    250      UPPER BOUND THROUGHPUT          800
 VARIABLE CAPACITY COST          590
NORTH MELBOURNE
    500      UPPER BOUND THROUGHPUT         1800
 VARIABLE CAPACITY COST          400
EAST MELBOURNE
    500      UPPER BOUND THROUGHPUT         1800
 VARIABLE CAPACITY COST          400
WEST MELBOURNE
    500      UPPER BOUND THROUGHPUT         1800
 VARIABLE CAPACITY COST          400
SOUTH MELBOURNE
    500      UPPER BOUND THROUGHPUT         1800
 VARIABLE CAPACITY COST          400
BALLARAT
     70      UPPER BOUND THROUGHPUT          800
 VARIABLE CAPACITY COST          590
BENDIGO
     70      UPPER BOUND THROUGHPUT          800
 VARIABLE CAPACITY COST          590
ADELAIDE
    200      UPPER BOUND THROUGHPUT         1200
 VARIABLE CAPACITY COST          430
WHYALLA
    100      UPPER BOUND THROUGHPUT          600
```

```
   FIXED COST       590000
NODE    80   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST       590000
NODE    82   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST       315000
NODE    84   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST      1064000
NODE    86   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST       415000
NODE    88   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST       415000
NODE    90   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST      1120000
NODE    92   CANDIDATE DC
   LOWER BOUND THROUGHPUT
   FIXED COST       675000
```

```
   VARIABLE CAPACITY COST          590
MT GAMBIER
      100      UPPER BOUND THROUGHPUT          600
   VARIABLE CAPACITY COST          590
ELIZABETH
       50      UPPER BOUND THROUGHPUT          350
   VARIABLE CAPACITY COST          630
PERTH
      100      UPPER BOUND THROUGHPUT         1200
   VARIABLE CAPACITY COST          430
KALGOORLIE
       50      UPPER BOUND THROUGHPUT          350
   VARIABLE CAPACITY COST          630
BUNBURY
       50      UPPER BOUND THROUGHPUT          350
   VARIABLE CAPACITY COST          630
HOBART
       40      UPPER BOUND THROUGHPUT         1200
   VARIABLE CAPACITY COST          450
LAUNCESTON
       40      UPPER BOUND THROUGHPUT          750
   VARIABLE CAPACITY COST          600
```

VITA

Charles Harold Aikens, III was born in Atlanta, Georgia on March 19, 1943. He attended elementary schools in Decatur and DeKalb County and was graduated from Avondale High School in June 1961. The following September he entered the Georgia Institute of Technology, and in June 1966 he received a Bachelor of Industrial Engineering degree. Upon graduation he was commissioned Ensign in the United States Naval Supply Corps.

After five years in the Navy, which included two tours of duty in Viet-Nam, and a progression to the rank of Lieutenant, he was released from active duty and joined the Industrial Engineering Department at Kingsport Press, Inc., Kingsport, Tennessee. Mr. Aikens commenced study toward a Master of Science degree in Industrial Engineering at The University of Tennessee, Knoxville, and this degree was awarded in June 1974.

In 1973 Mr. Aikens was promoted to Manager, Production Control and Cost Estimating, at Kingsport Press, Inc., a position he held until the end of 1974 when he resigned in order to move his family to Australia. In early 1975 he accepted an appointment as Lecturer in Quantitative Methods with the School of Business Studies, Darling Downs Institute of Advanced Education (DDIAE), Toowoomba, Queensland.

In 1979 the Governing Council of DDIAE granted Mr. Aikens a twelve-month sabbatical leave to enroll in a doctoral program at The University of Tennessee, Knoxville. At the conclusion of the sabbatical leave, he returned to DDIAE where he was promoted to Senior Lecturer and made

Head of Programme, Operations Management. In 1981 he was the recipient of the Walter Melville Bonham Fellowship in the Collegeof Business Administration at The University of Tennessee, Knoxville. He was granted an extended leave of absence from DDIAE and returned to The University of Tennessee, Knoxville, with the help of a graduate teaching assistantship. He completed the requirements for the Doctor of Philosophy degree with a major in Management Science in December 1982. Following graduation, Mr. Aikens resumed his academic duties at DDIAE.

He has diverse consulting experience and has been a contributor to several textbooks and a syndicated small business newspaper column. He is a member of Phi Kappa Phi, the Institution of Engineers, Australia; the Institute of Industrial Engineers of Australia; and the Australian Society of Operations Research. He is a senior member of the American Institute of Industrial Engineers.

Mr. Aikens is married to the former Helen Rae Maskell of Brisbane, Queensland, Australia, and they have two children, Christopher Harold and Jocelyn Reneé.