UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

DTIC FILE COPY

(1)

AD-A196 874

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER AFIT/CI/NR 88- 31	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) MULTIATTRIBUTE MULTICOMMODITY FLOWS IN TRANSPORTATION NETWORKS		5. TYPE OF REPORT & PERIOD COVERED PHD THESIS
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) DOUGLAS ALLEN POPKEN		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS AFIT STUDENT AT: UNIVERSITY OF CALIFORNIA - BERKELEY		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE 1988
		13. NUMBER OF PAGES 102
14. MONITORING AGENCY NAME & ADDRESS (If different from Controlling Office) AFIT/NR Wright-Patterson AFB OH 45433-6583		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

DISTRIBUTED UNLIMITED: APPROVED FOR PUBLIC RELEASE

DTIC ELECTE AUG 0 4 1988 S D H

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

SAME AS REPORT

18. SUPPLEMENTARY NOTES

Approved for Public Release: IAW AFR 190-1
LYNN E. WOLAVER *Lynn Wolaver* 18 Jul 88
Dean for Research and Professional Development
Air Force Institute of Technology
Wright-Patterson AFB OH 45433-6583

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

ATTACHED

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Multiattribute Multicommodity Flows in Transportation Networks

Douglas A. Popken

Abstract

Previously developed minimum cost multicommodity network flow models do not simultaneously consider the weights, volumes, and inventory holding costs of the commodities. Ignoring one or more of these attributes may prevent detection of potential savings; however, simultaneously accounting for all three attributes leads to a problem considerably more difficult to solve. This thesis examines a multiattribute multicommodity flow formulation of a transportation network with transshipment terminals, which seeks to minimize total vehicle and inventory related costs. First, a decomposition strategy transforms the model formulation, P, into an equivalent formulation, P'. In P', the vehicle flow variables may be found as a function of the commodity flow variables; furthermore, the vehicle capacity constraints need not be explicitly considered. The decomposition, however, creates a situation whereby a commodity's incremental cost function on a given arc may contain concave and/or convex portions. This feature implies the presence of numerous local optima, over which an exhaustive search for a global optimum is computationally infeasible. To overcome this difficulty, an algorithm is developed to find a local optimum that is "good", in the sense of being superior to nearby local optima. The procedure alternates between a version of the Franke-Wolfe algorithm to find local optima, and a search routine based upon "adjacent concave flows" to provide local improvements. Computational results from a series of test problems measure solution quality and algorithm efficiency.

Randolph Hall

Multiattribute Multicommodity Flows in Transportation Networks

By

Douglas Allen Popken

B.S. (Cornell University) 1980
M.Eng. (Cornell University) 1981

DISSERTATION

Submitted in partial satisfaction of the requirements for the degree of

DOCTOR OF PHILOSOPHY

in

ENGINEERING

INDUSTRIAL ENGINEERING AND OPERATIONS RESEARCH

in the

GRADUATE DIVISION

OF THE

UNIVERSITY OF CALIFORNIA, BERKELEY

Approved: 4/11/88
Chairman Date

....

....

....

Multiattribute Multicommodity Flows in Transportation Networks

Douglas A. Popken

Abstract

Previously developed minimum cost multicommodity network flow models do not simultaneously consider the weights, volumes, and inventory holding costs of the commodities. Ignoring one or more of these attributes may prevent detection of potential savings; however, simultaneously accounting for all three attributes leads to a problem considerably more difficult to solve. This thesis examines a multiattribute multicommodity flow formulation of a transportation network with transshipment terminals, which seeks to minimize total vehicle and inventory related costs. First, a decomposition strategy transforms the model formulation, P, into an equivalent formulation, P'. In P', the vehicle flow variables may be found as a function of the commodity flow variables; furthermore, the vehicle capacity constraints need not be explicitly considered. The decomposition, however, creates a situation whereby a commodity's incremental cost function on a given arc may contain concave and/or convex portions. This feature implies the presence of numerous local optima, over which an exhaustive search for a global optimum is computationally infeasible. To overcome this difficulty, an algorithm is developed to find a local optimum that is "good", in the sense of being superior to nearby local optima. The procedure alternates between a version of the Franke-Wolfe algorithm to find local optima, and a search routine based upon "adjacent concave flows" to provide local improvements. Computational results from a series of test problems measure solution quality and algorithm efficiency.

Randolph Hall

Acknowledgements

I would like to thank Professor Randolph Hall for his numerous suggestions and guidance throughout the preparation of this dissertation. Professors William Jewell and Carlos Daganzo also provided help in defining the direction of my research, and helpful comments towards improving the contents of this dissertation.

I also wish to acknowledge the Air Force Institute of Technology and the US Air Force for supporting my PhD program and providing me with the opportunity to attend UC Berkeley.

In particular, I would like to thank my wife, Susie, for her understanding, patience, and support.

Table of Contents

	pg.
Chapter 1. Introduction	1
1.1 Physical Context	1
1.2 Tradeoffs Involved	4
1.3 Example	6
1.4 Thesis Objectives	15
1.5 Thesis Summary	16
Chapter 2. Literature Review	17
2.1 Linear Network Flow Models	17
2.2 Nonlinear Network Flow Models	21
2.3 Tradeoff Analysis	24
2.4 Summary	25
Chapter 3. Mathematical Analysis	26
3.1 Problem Description	26
3.2 Mathematical Programming Formulation	29
3.3 Analysis of the Arc Cost Function	31
3.4 Bounds on the Degree of Nonlinearity	44
3.5 Summary	51
Chapter 4. Local Optima	53
4.1 Marginal Definition of a Local Optimum	55
4.2 Adjacent Concave Flows	59
4.3 A Stronger Definition of a Local Optimum	63
4.4 Summary	69
Chapter 5. Computational Results	71
5.1 Quality of Solutions	71
5.2 Dependence Between Initial and Final Solutions	82
5.3 Level n Optimality	85
5.4 Summary	90
Chapter 6. Conclusions	92
References	96
Appendix 1A	98
Appendix 1B	100

1. INTRODUCTION

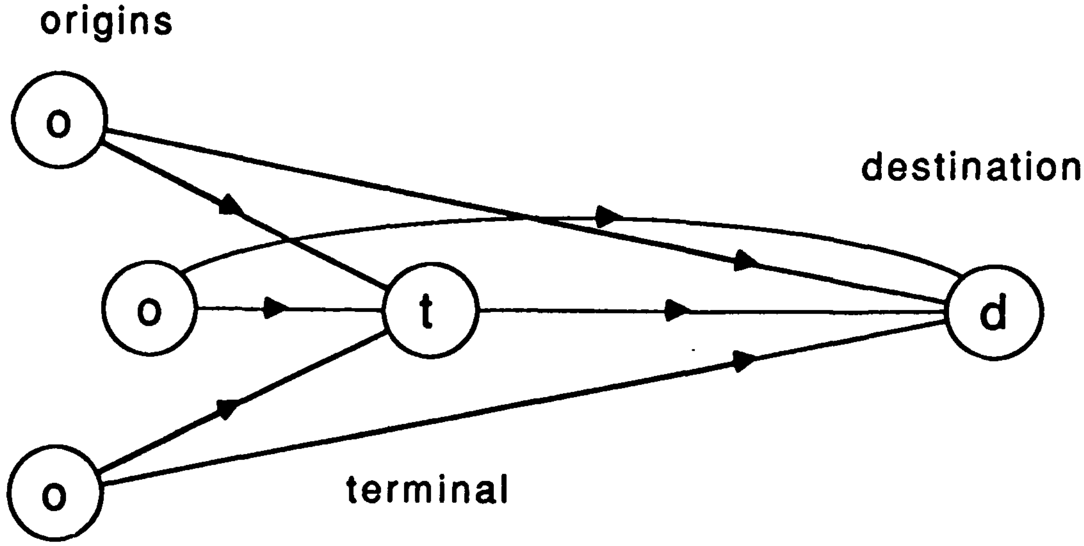
A transportation system is more than just a set of warehouses, vehicles, and routes. It is also a set of policies that direct the timing and the procedures to use in the logistical actions of transporting, handling, and storing goods. These actions all involve costs, with the magnitude of the costs dependent upon the policies in place. Since every item we use requires numerous types of logistical actions to bring it to its final destination, logistics related costs form a significant component of the total cost of a good, and ultimately, the price paid by the consumer. It would obviously be useful to devise policies that make the logistical process more efficient.

1.1 Physical Context

In this thesis we will assume the viewpoint of a single shipping firm which desires to minimize its overall shipping costs. The firm transports loads of goods from multiple origins to multiple destinations via routes that are either direct, or that have intermediate stops at transshipment terminals. At the terminals, loads are reassigned to vehicles that take them to their next stop. We can easily represent this physical system in graph form. Figure 1.1 illustrates such a graph for a simple example with only one terminal and one destination. Though many of the concepts developed in this study are applicable to a more generalized network, the emphasis will be on networks that allow commodities a maximum of one terminal stop on any path between their origin and their destination. Furthermore, shipping strategies where the vehicles make stops at multiple numbers of origins or destinations, known as collecting and peddling [2], will not be considered here.

Figure 1.1

Basic Network Configuration in Single Terminal - Single Destination Case



Each of the commodities will have an associated weight, volume, and dollar value. This is a departure from previous network flow studies that have tended to concentrate on one, or perhaps two, of these attributes. As we will see later, ignoring one or more of these attributes may prevent detection of potential savings; however, simultaneously accounting for all three attributes leads to a problem considerably more difficult to solve.

We will assume that shipping occurs continuously, with commodities being shipped between specific origin-destination pairs at a constant rate per unit time. The firm is responsible for all costs, from the time a unit is available at an origin, to the time it reaches its final destination. These assumptions affect the way that time related costs are determined.

The time related costs, or inventory carrying costs, result from the time that commodities must spend in transit, and the average time that they must spend waiting to be picked up by vehicles. These costs represent the opportunity cost of money being "tied up" in inventory, as well as costs associated with the storage of goods over time. The inventory carrying cost occuring in transit is assumed to be proportional to the distance traveled; however, this cost could also include a fixed component to reflect unit material handling costs at the beginning and/or the end of travel on the arc, if so desired.

Costs are associated with the vehicles themselves, and come about from such things as fuel costs, driver wages, highway use fees, and amortization costs. We will assume that all vehicles are identical; thus, the vehicles incur identical costs for travel on a given arc. The vehicle costs on an arc will consist of a fixed component, common to all arcs in the network, plus a variable component proportional to the distance traveled on that particular arc.

The solution to the problem will consist of two simultaneously solved parts. The first part will be the determination of the optimal vehicle flow rate (shipping frequency), in vehicles per unit time, across each arc. Specific vehicle routes will not be considered. The other part of the solution is the determination of the optimal commodity flow across each arc. The vehicle flow rate will affect the total inventory carrying costs of the commodities; moreover, there is a physical link between vehicle flow and commodity flow as a result of the volume and weight capacities of the vehicies.

1.2 Tradeoffs Involved

Designing a transportation system involves the evaluation of numerous tradeoffs. The most basic of these is the tradeoff between transportation costs and inventory carrying costs, both of which depend upon vehicle flow. These two factors are related to vehicle flow such that increasing vehicle flow increases transportation costs, but decreases inventory carrying costs. To illustrate, let:

$\mathbf{x}_{ij}$ = the multicommodity flow vector on arc (i,j),
$y_{ij}(\mathbf{x}_{ij})$ = vehicle flow on arc (i,j) (an increasing function of $\mathbf{x}_{ij}$),
$\mathbf{h}$ = the vector of multicommodity inventory holding cost parameters,
$\mathbf{hx}_{ij}$ = the total inventory holding cost per unit time on (i,j)[1], and
C_{ij} = the transportation cost per vehicle on arc (i,j).

Then we can say:

$$\text{total arc cost/unit time} = C_{ij}y_{ij}(\mathbf{x}_{ij}) + \mathbf{hx}_{ij}/2y_{ij}(\mathbf{x}_{ij}) \ . \qquad (1.1)$$

Through calculus, one can determine the optimal value of $y_{ij}(\mathbf{x}_{ij})$ for a fixed level of $\mathbf{x}_{ij}$. Complications arise in this study because we must simultaneously determine the optimal values of the commodity flows (which also affect inventory costs), and the vehicle flows.

Another tradeoff involves balancing capacity utilization and the total distance a unit of commodity must travel. This tradeoff arises when we consider both the weight and the volume of the commodities. Vehicles may be filled either to their weight capacity, their volume

---

1. The convention throughout this thesis will be that vectors are denoted by bold-face type. A symbol such as $\mathbf{hx}_{ij}$ will refer to the multiplication of two vectors.

capacity, or possibly both, depending upon the mixture of commodities being carried within the vehicle. It seems intuitive that filling a vehicle simultaneously to both its weight and volume capacities generates a certain amount of efficiency. This is not always possible, as the proper mix of commodities may not be available at a certain location and/or during a certain time interval. One way to accomplish this objective is to consolidate high and low density vehicle loads originating at different locations. This is achieved in this thesis through the use of one or more transshipment terminals. The disadvantages of this approach are the additional number of item-miles traveled, which tends to increase vehicle costs, and the additional time required to ship the commodities, which increases inventory costs.

In the case of the capacity/distance tradeoff, we can achieve savings by consolidating commodities having low densities with commodities having high densities. It may also be possible to achieve savings by consolidating commodities having low inventory costs with commodities having high inventory costs. This is possible due to the effects of two factors. The first of these is capacity utilization. Note that it may be least costly to ship expensive commodities in partially filled vehicles. In some cases we can then use less expensive goods to fill the truck to capacity at the consolidation terminal, thus saving on vehicle costs. The second factor relates to the inventory costs themselves. Average inventory costs on the terminal-to-destination links are reduced as the less expensive commodities make up a greater proportion of the loads leaving the terminal. Net savings will only be realized if the extra shipping time involved for the expensive commodities is small. On the other hand, the presence of high inventory costs might nullify any savings achieved by consolidating high and low density commodities.

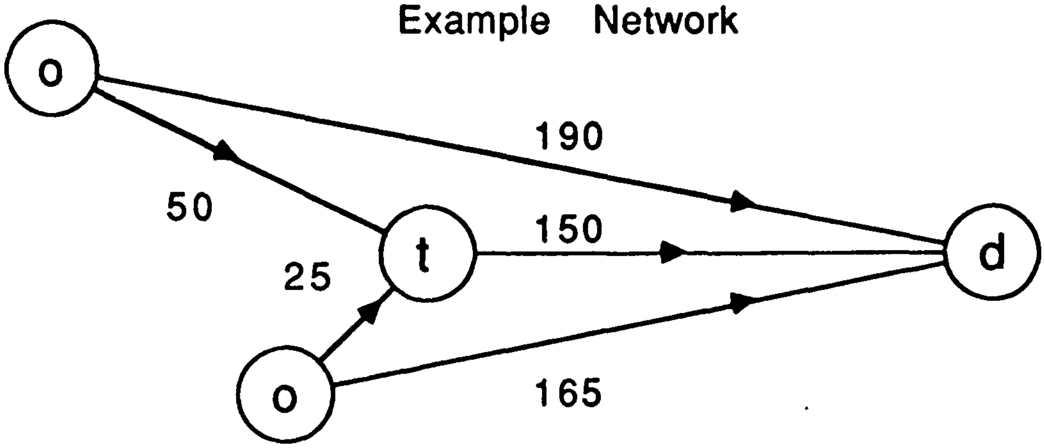
The following example should serve to illustrate these tradeoffs, and give some insight as to when they might occur.

1.3 Example

Consider a transportation network with two origins and one destination, with each origin producing a single unique commodity. Flow can be routed directly from an origin to the destination, or first to a transshipment terminal and then to the destination. Such a network is shown in Figure 1.2.

Figure 1.2

Example Network



In deriving costs, we will initially assume that inventory costs are negligible. Later in this example, we will analyze the effects of adding inventory costs to the formula. We will also assume that the vehicle flow is a continuous variable. This provides no inaccuracy if we assume that vehicles can be dispatched in non-integer time intervals. For example, a value of 1.5 vehicles per week is equivalent to

3.0 vehicles every two weeks. Assume also that commodities fill a vehicle proportionately. This assumption is valid as long as a shipping unit (e.g. box) of a commodity is small in relation to vehicle capacity, thus allowing us to load vehicles with little or no wasted space. We will also assume that all vehicles have the same capacity. This will not have any effect upon the methodology, but it will reduce the number of variables that must be considered.

Let :

s^k = the density of commodity k; k=1,2 ; in lbs./ft^3,

s^0 = the weight capacity of a vehicle/ volume capacity of a vehicle,

d^k = the demand for commodity k in weight loads/unit time, and

C_{ij} = the cost per vehicle on arc (i,j).

Assume that the weight capacity of a vehicle is 80,000 lbs., and that the volume capacity is 4200 ft^3, typical values for large trucks operating on U.S. highways; thus s^0 =19.05 lbs./ft^3. The parameters of the example are summarized in Table 1.1 below.

Table 1.1

commodity k (or node k)	d^k (demand)	C_{kt}	C_{kd}	s^k (density)
0	-	-	-	19.05
1	.0125	50	190	1
2	5.0	25	165	50
t	-	-	150	-

Observe s^0, referred to by Daganzo [3] as the "ideal density". Commodity 1 has a density of 1 lb/ft^3, and is therefore "light" in relation to the ideal density. Commodity 2 has a density of 50 lbs/ft^3, and is therefore "heavy" in relation to the ideal density. Thus, there would appear to be some potential savings in combining the two commodities at the transshipment terminal. In this way we might be able to produce loads with a density closer to that of the ideal density, thus utilizing vehicle capacities more efficiently. However, we need to weigh any efficiency savings against the added cost of the additional routing circuity.

Vehicle Costs Only

We will now derive the cost function and use this to compare the costs of direct shipping versus consolidating the flows. The only costs involved with this initial formulation are vehicle costs.

Let:

x_{ij}^{k} = flow of commodity k on arc (i,j) in weight loads/unit time, and

y_{ij} = vehicle flow on arc (i,j) in vehicles/unit time .

Note that $\Sigma_k s^0 x_{ij}^k / s^k$ is equivalent to the amount of commodity k flow on arc (i,j) measured in *volume loads*/unit time. The minimum number of vehicles required per unit time on arc (i,j) can be given by:

$$y_{ij} = \max\{ \Sigma_k x_{ij}^k, \Sigma_k x_{ij}^k s^0 / s^k \} . \quad (1.2)$$

Note that the first term in equation (1.2) applies when loads are "heavy", and the second term applies when loads are "light". When loads are heavy we will say that costs are weight dominated; when loads are light we will say that costs are volume dominated. Let K be the total shipping costs incurred per unit time. This can be written as:

$$K = \Sigma_{i,j} C_{ij} y_{ij} . \quad (1.3)$$

Note that costs on an arc depend only upon the particular attribute that currently dominates the arc cost function.

Case 1: Direct Shipping Only

Let K_d be the total shipping cost per unit time when all commodities are shipped direct. By using equation (1.2) we obtain:

$$y_{1d} = d^1 s^0/s^1,\ y_{2d} = d^2,\ y_{1t} = y_{2t} = y_{td} = 0\ .$$

By using the above results in equation (1.3) we obtain:

$$K_d = \$870.20\ /\text{unit time}\ .$$

Case 2: Direct and Terminal Shipping

To calculate the costs when we include the possibility of using the transshipment terminal, we first must determine the proper proportion of the flows to be sent there. Let:

f^{k*} = the optimal proportion of commodity k flow to be routed via the transshipment terminal.

Daganzo [3] uses a matching technique to determine the optimal f^k's for networks constructed similarly to that of Figure 1.2, but with any number of origins. Using this technique, one finds that $f^{1*} = 1.0$ and $f^{2*} = .0729$ are the optimal values. It can be shown that for any optimal solution involving consolidation: 1) at least one commodity will always be shipped through the terminal in its entirety, and 2) all loads leaving the transshipment terminal will have the ideal density. The vehicle flows can be found by again using equation (1.2) to obtain:

$$y_{1t} = d^1 s^0/s^1,\ y_{2t} = f^{2*}d^2,\ y_{2d} = (1-f^{2*})d^2,\ y_{1d} = 0, \text{ and } y_{td} = f^{2*}d^2+d^1.$$

By using the above results in equation (1.3) we obtain:

K = $842.43 /unit time .

This is a net reduction of approximately 3.2% over direct shipping.[1] This reduction is possible through more efficient utilization of capacity, which in this case dominates the extra costs incurred by additional item-miles traveled.

Vehicle Costs and Inventory Costs

We will now consider the same network with inventory costs included in the cost function. However, to simplify the analysis, we will assume that the time spent in transit is small compared to the average time that a unit of commodity spends waiting to be loaded. In this way, we can ignore the "in-transit" inventory carrying costs. Let:

h^k = inventory cost/weight load/unit time for commodity k.

The total inventory cost per unit time on each arc is: $\Sigma_k h^k x^k_{ij}/2y_{ij}$, which is merely the summation of the products of the average inventory levels (in weight loads) and the h^k coefficients. The total shipping costs per unit time can now be written as:

$$K = \Sigma_{i,j} [\ C_{ij} y_{ij} + \Sigma_k h^k x^k_{ij}/2y_{ij}\ ] \ . \qquad (1.4)$$

By minimizing this function with respect to y_{ij} we find that, in the absence of capacity constraints, the optimal number of vehicles per

---

1. The example could have illustrated a much higher percentage cost reduction by increasing commodity 1 demand. However, the existing parameters were chosen to illustrate several types of tradeoffs without having to later change previously defined parameters.

unit time on arc (i,j) can be given as

$$y_{ij} = (\Sigma_k h^k x_{ij}^k / 2C_{ij})^{\frac{1}{2}} . \tag{1.5}$$

If we account for the capacity constraints of equation (1.2), the fact that equation (1.4) is convex in the y_{ij}'s implies that the optimal value of y_{ij} will then be given either by equation (1.5), or by a value at the edge of the feasible region. This leads to:

$$y_{ij} = \max\{ \Sigma_k x_{ij}^k, \Sigma_k x_{ij}^k s^0 / s^k, (\Sigma_k h^k x_{ij}^k / 2C_{ij})^{\frac{1}{2}} \} . \tag{1.6}$$

Note that equation (1.4) is a summation of equations of the form (1.1), where the optimal $y_{ij}(x_{ij})$'s are defined by equation (1.6). Earlier, we made the convention that when the first term in equations (1.2) or (1.6) is largest, we say that costs are weight dominated, and when the second term is largest, we say the costs are volume dominated. Similarly, when the third term in equation (1.6) is largest, we say that costs are dollar dominated. This implies that marginal costs depend only upon the inventory holding cost of the commodities. That is, weight and volume are relevant only when vehicles are filled to one of these capacities. Note however, that if costs are weight or volume dominated, there will still be an inventory cost component.

In this example we will assume that h^1 = \$80,000/weight load/unit time and h^2 = \$8.0/weight load/unit time. In relation to each other, commodity 1 is very expensive, and commodity 2 is quite cheap, even on a per unit basis.

Case 1: Direct Shipping Only

In the case of direct shipping we use equation (1.6) to obtain:

$$y_{1d} = (h^1 d^1 / 2C_{1d})^{\frac{1}{2}}, \quad y_{2d} = d^2, \text{ and } y_{1t} = y_{2t} = y_{td} = 0 \; .$$

Using these results in equation (1.4) we obtain:

$$K_d = \$1445.44 \; / \text{ unit time} \; .$$

This is clearly greater than the case where inventory costs are not considered.

Case 2(a): Direct and Terminal Shipping (earlier derived f^{k*}'s)

Observe what happens when we consolidate commodities in the proportions used earlier (f^1=1.0,f^2=.0729). In that case use of equation (1.6) results in:

$$y_{1t} = (h^1 d^1 / 2C_{1t})^{\frac{1}{2}}, \quad y_{1d} = 0, \quad y_{td} = (\Sigma_k h^k x^k_{td} / 2C_{td})^{\frac{1}{2}},$$
$$y_{2t} = f^2 d^2, \text{ and } \quad y_{2d} = (1-f^2) d^2 \; .$$

Using these results in equation (1.4) we obtain:

$$K = \$1646.8 \; / \text{ unit time}.$$

Consolidating flows in the once optimal mix gives us a *higher* cost than direct shipment. This comes about mainly from the additional waiting time incurred at the transshipment terminal. Observe that the loads on the terminal to destination arc still have the ideal density. However, vehicles leaving the terminal are no longer filled to capacity. This is because inventory costs dominate the physical capacity considerations, causing vehicles to leave more frequently than would occur otherwise.

The above analysis does not necessarily imply that we should not consolidate commodities when high inventory costs are present. On the contrary, we might be able to achieve savings by sending *more* "cheap" goods (commodity 2) to the terminal than before to take advantage of the unused space on the outbound vehicles. In addition, average inventory costs per unit time are reduced as the cheaper commodity makes up a larger porportion of the vehicle loads leaving the terminal. We will examine whether or not these effects lead to net savings.

Case 2(b): Direct and Terminal Shipping (optimal proportions)

Suppose now that f^2 is set to 1.0 . At this level, all of the excess capacity generated by the expensive commodity 1 is used up, that is, $\Sigma_k x^k_{td} > (\Sigma_k h^k x^k_{td}/2C_{td})^{\frac{1}{2}}$. However, average inventory costs will continue to decrease as we add more commodity 2 flow through the terminal. Now use of equation (1.6) results in:

$$y_{td} = \Sigma_k x^k_{td},\ y_{1t} = (h^1 d^1/2C_{1t})^{\frac{1}{2}},\ y_{2t} = d^2,\ y_{1d} = y_{2d} = 0 .$$

Using these results in equation (1.4) we obtain:

$$K = \$1300.80 \text{ /unit time}$$

We obtain a reduction over direct routing of approximately 10%, and a reduction over the non-inventory based solution of 27%. Without considering the inventory cost attribute, we would have consolidated commodities in a highly nonoptimal manner. However, note that the savings reduction above does not directly depend upon the relative densities of the commodities either. To see this, suppose that s^2 was decreased from 50 to 12.5, h^2 was decreased from 8.0 to 2.0, and d^2 was increased from 5 to 20. The same solution, that is, $f^1 = f^2 = 1.0$, produces identical costs and savings. However now both commodities are light: $s^1 = 1.0$, $s^2 = 12.5 < s^0 = 19.05$.

Example Summary

We have illustrated several tradeoffs in comparing alternative shipping strategies. The net effects of a change in commodity flow routing has been shown to depend upon the relative values of weight, volume, and inventory cost for each commodity. We should not only consider consolidating commodities with high and low densities, but also commodities with high and low inventory costs.

Earlier we pointed out that marginal costs on a given arc depend upon only one or two attributes of the commodities using that arc. In general, we will seek to add flows to arcs in the most efficient manner. Thus, on a dollar dominated arc, we would desire to add flow with a low inventory holding cost, but high weight and/or volume. Similarly, on a weight dominated arc, we would desire to add flow with a low weight but high volume. However, since we wish to minimize the sum of the costs on all arcs, we must weigh the potential benefits on all arcs simultaneously. As we shall see, the special cost structure of this problem makes our task quite difficult.

1.4 Thesis Objectives

We have seen that a number of tradeoffs must be considered simultaneously when solving the type of problem described earlier in this chapter. As we shall learn in the next chapter, the literature review, previous model formulations have tended to ignore at least one of the commodity attributes of weight, volume, or inventory holding cost. The main reason for this is the great difficulty of obtaining optimal or even near optimal solutions for this problem in networks of any practical size.

This thesis will attack the problem by first examining its mathematical structure. Through this examination, we hope to come to a better understanding of the underlying tradeoffs, and to find features of the problem that can be exploited by specialized solution algorithms.

We will also devise a solution algorithm that can be applied to network problems of a size that would be encountered in practical usage. This requirement will likely preclude finding a global optimum; but if we can obtain "good" locally optimal solutions with a reasonable amount of effort, this disadvantage is minimal. One means of attaining such solutions is by employing a strict standard upon the type of local optima that will be acceptable as a final solution. Due to the possibly large number of local optima to be evaluated, the algorithm must perform its search and evaluation procedures quickly and efficiently.

Finally, we will endeavor to test the methods and analyze the results by producing a workable computer code of the solution algorithm. In this way, we can assess the quality of the solutions, and determine the efficiency of the algorithm under differing problem scenarios.

1.5 Thesis Summary

Chapter 1 provided an overview of the motivation for and the primary issues involved with the logistical problem examined in this thesis. In Chapter 2, we will discuss previous research in related areas of study. The relevance of the research will be examined, providing a basis of comparison for this thesis' model formulation, analysis, and solution methods.

Chapter 3 will begin with a detailed problem statement. A mathematical programming formulation of the problem is then followed by an in-depth analysis of the cost functions associated with the arcs of the network. This analysis will determine the necessary conditions for concavity/convexity, the relevancy of nonlinearity in practice, and the effect nonlinearity has on our solution strategies.

In Chapter 4, we will explore local optimality conditions -- specifically, one set that has been in standard usage, and a stronger, more restrictive set introduced and defined in the context of this problem. Algorithms that can achieve both sets of optimality conditions will be introduced, followed by a discussion of the merits and the drawbacks of each approach.

Chapter 5 contains a series of results from computer trials on test problems using a new algorithm presented in Chapter 4. These results measure solution quality and algorithm speed for various network configurations.

Lastly, Chapter 6 provides conclusions regarding both the significance of the mathematical results of this thesis, and the utility and efficiency of the new algorithm .

2. LITERATURE REVIEW

Although the problem under consideration in this study is unique in the breadth of factors considered, it shares characteristics with a number of earlier formulations of multicommody min-cost flow problems. By placing restrictions on the cost structure, or upon the types of constraints considered, our problem can be reduced to these earlier models. Therefore, to thoroughly understand our problem, it is necesary to examine these formulations, and see how previous researchers have proposed solving them.

2.1 Linear Network Flow Models

Single Capacity

If we set vehicle frequencies beforehand, the nonlinear inventory costs become fixed, and the problem is reduced to a linear multicommodity min-cost flow problem. The vehicle frequencies, together with the vehicle capacities, would correspond to the arc capacities. If we further assume that vehicles have only one type of capacity (weight or volume), we have the standard multicommodity flow problem with one type of arc capacity. These problems have been studied extensively due to their wide application and appealing structure. As with other linear cost network problems, multicommodity flow problems can be modeled directly as a linear program, and solved by standard LP algorithms, such as the simplex method. However, the multicommodity flow problem can become very large in terms of the number of variables and constraints which must be considered, making the simplex method prohibitively slow. Thus, much of the research has centered around specialized solution techniques that attempt to capitalize upon the underlying network structure of the problem. Kennington [9] provides an extensive survey of linear multicommodity flow problems and their solution techniques. He divides these techniques into the categories of price-directive decompositions, resource-directive decompositions, and partitioning methods. Several forms of column generation algorithms are presented which have been used to solve price-directive formulations. An important example was provided by Tomlin [15], who was the first to develop a price-directive decomposition for the linear multicommodity flow problem. In his formulation, he assumes infinite arc capacities, thus reducing the subproblems of the algorithm to simple shortest path problems. Kennington also describes several resource-directive decompositions, and shows how they have been solved by tangential approximation and gradient search algorithms. Finally, partitioning techniques are presented using various methods that exploit the structure of the current basis in the simplex method.

Multiple Capacities

Further difficulty is introduced to the linear multicommodity min-cost flow problem when multiple capacity constraints on the arcs are used. Weigel and Cremeans [16] provide a formulation in which capacities are expressed in vehicles per unit time period and different commodities can be expressed in varying units of measurement. However, they neglect the possibility of combining high and low density commodities on the same vehicle to economize on vehicle capacity requirements. Swoveland [14] uses a price directive decomposition technique to solve a problem with generalized capacity constraints. The model is flexible enough to handle weight and volume constraints simultaneously, as well as other types of resource constraints.

Hall and Daganzo [8] analyze transportation costs with consideration of both the weight and volume capacities of the vehicle. The analysis is performed in the context of a "peddling" strategy used over a region of suppliers with items of varying densities. (Peddling strategies allow vehicles to make multiple pickups or dropoffs along the course of a vehicle route.) They show that the number of vehicle loads required is minimized when all vehicles are simultaneously filled to both capacities. Cases are analyzed to determine potential savings, which may be as much as 50%, when compared to solutions obtained by using only one type of capacity.

Daganzo [3] considers a two capacity network model with many origins, one destination, and a consolidation terminal. Each origin produces a unique commodity with a given density. His formulation has a linear cost structure and seeks to minimize the total vehicle-miles traveled. The solution is obtained via graphical methods derived from linear programming theory. The solution technique capitalizes upon the fact, which he proves in his paper, that costs are minimized when the flows leaving the consolidation terminal have the "ideal density"; that is, both vehicle capacities are reached simultaneously.

Mixed Integer Models

The linear multicommodity flow problem assumes that all costs are directly proportional to the flow of commodities across the arcs, and that arc capacities are fixed and known beforehand. Another class of network flow models, generally known as "network design problems", also consider the costs of optionally added arc capacity, which in our problem would correspond to solving for the vehicle frequency variables. These models usually involve finding the optimal *integer* capacities on the arcs, along with the optimal commodity flows, subject to various demand, conservation of flow, and possibly, budgetary constraints. Thus, one approach to our problem would be to restrict the vehicle frequency (capacity) variables to integer values, and then to use the standard techniques available for solving mixed integer mathematical programming models. The subproblem of such an algorithm generally finds the optimal values of the non-integer variables for given values of the integer variables. In our case, the subproblems would be linear multicommodity flow problems, which could be solved via any of numerous techniques mentioned earlier. Unfortunately, the number of possible combinations of feasible vehicle frequencies is very large, thereby limiting the size of network problem which could be solved successfully. However, our model allows the capacity variables to be non-integer by assuming that vehicles may have non-integer interdeparture times. We can then treat arc capacity as a vehicle flow (measured in vehicles per unit time) with real values. It can be shown that if we were to ignore inventory holding costs due to waiting time prior to loading, our model would become an LP relaxation of a mixed integer network design problem. Magnanti and Wong [12] provide an extensive survey of network design problems involving both linear and nonlinear cost structures.

2.2 Nonlinear Network Flow Models

The problem defined in this thesis is made more difficult by a nonlinear objective function. This nonlinearity comes about from the relationship between inventory holding costs and vehicle frequency. It is useful to review some of the methods used previously to deal with nonlinearity in multicommodity flow problems. However, none of the models that were examined reflect multiple arc capacities, being either uncapacitated or having single arc capacities. In the case of capacitated networks, it is often the case that a transformation is first applied to the network to obtain an equivalent, but larger, uncapacitated network flow model. Other strategies will be discussed with some of the models below.

Convex Network Flow Models

Convex cost network design models have received a lot of attention, partially because of the availability of numerous convex optimization techniques, and have often been used to solve traffic assignment problems. In addition, convex models have the desirable feature that a global optimum can be found through straightforward optimization techniques. LeBlanc [11] uses a branch-and-bound algorithm to solve a mixed integer model with convex routing costs. The flows are essentially single commodity in the sense that all "commodities" have a single shared flow attribute, and there are no arc capacities. The problem is reduced to solving a sequence of shortest path problems and one dimensional searches.

Steenbrink [13] uses a decomposition technique to solve a convex road expansion/traffic assignment model. He first eliminates the road expansion (capacity) variables from the main problem by placing them in a subproblem which relates optimal capacity to traffic flow on each link. The main or "master" problem minimizes a sum of the objective functions on the links; these functions being concave at low levels of flow, and convex at higher levels of flow. However, he simplifies the problem by assuming that the objective functions are strictly convex, and justifies this with the intuitive fact that links with low levels of flow are not expected to be part of the final solution. The master problem can then be solved via a stepwise loading algorithm.

Concave Network Flow Models

Many transportation related problems can be solved using concave cost flow models. For example, by using Steenbrink's decomposition technique, many fixed charge problems can be reduced to a multicommodity concave-cost flow problem. It is a characteristic of such problems that the optimal solution occurs at an extreme point of the feasible region. It is known that these extreme points correspond to arborescence flows in the network [18], but the large number of feasible extreme points considerably complicates the search for a global optimum. Therefore, many authors have proposed algorithms that converge to a *locally* optimal solution. Among these is Yaged [17], who uses both marginal cost and average cost pricing schemes to reduce the problem to a series of shortest path problems. He shows that for an important special case, his algorithm will converge at a local optimum, or will produce an infinite sequence which converges to a local optimum. Gallo and Sodini [6] define the concept of adjacent extreme flows to a feasible solution of the concave min-cost flow problem. They provide an algorithm that searches all adjacent extreme flows to find a locally optimal solution. They show that their technique can make considerable improvement over solutions obtained using Yaged's procedure.

Other authors have sought algorithms which find the global optimum to the concave min-cost flow problem. Erickson, Monma, and Veinott [4] have devised a dynamic programming technique, known as send-and-split, to find a global optimum to the problem. For certain classes of problems, their method outperforms previous global solution techniques. However, since the general concave min-cost flow problem is very hard, their technique is still limited in the size of the problem which can be solved.

Blumenfeld, Burns, Diltz, and Daganzo [1] analyze cost tradeoffs between transportation, inventory, and production costs in concave cost networks with a consolidation terminal. Strictly concave costs are obtained by assuming a fixed freight charge per shipment, independent of shipment size, up to the capacity of the vehicle. By decomposing the network into sub-networks, and capitalizing on the all-or-nothing feature of the optimal solution to concave cost network flow problems, they eliminate the need for mathematical programming techniques.

Hall [7] analyzes a special type of concave cost freight network which can be decomposed into "shared" and "exclusive" arcs. He shows that in this type of network, the optimal route for each origin-destination pair can be found from the optimal flows on the shared arcs alone. The problem is solved using a local search algorithm that relies upon the concepts of both marginal cost and "incremental cost" as optimality criteria. Computational results are provided which show time requirements to be relatively small, even for large networks.

General Nonlinear Network Flow Models

Klessig [10] provides a solution method for general nonlinear min-cost multicommodity flow models which requires only that the cost function be continuously differentiable, and that the constraint space be compact and convex. The solution algorithm, known as the conditional gradient method (CGM), involves calculating gradients and performing line searches in the direction of these gradients to achieve a local optimum. This technique builds upon a method originally proposed by Franke and Wolfe [5], through inclusion of a line search particularly well suited for multicommodity flow problems. Klessig shows that other solution techniques, including Yaged's, for solving both convex and concave cost multicommodity flow problems, are merely special cases of the CGM. Convergence is guaranteed, but no computational results are presented.

2.3 Tradeoff Analysis

Other studies have approached the min-cost flow problem using analytical frameworks other than mathematical programming. These studies have tended to concentrate on explicitly considering cost tradeoffs at a conceptual level, as opposed to complex optimization techniques with precise costs.

Burns, Hall, Blumenfeld, and Daganzo [2] compare alternative distribution strategies, such as peddling and direct shipping. economic order quantity type formulas are developed to examine the transportation cost/inventory cost tradeoff under both strategies, with a single vehicle capacity taken into consideration. The formulas require only a few easily measurable parameters, and can be solved via a hand calculator.

2.4 Summary

The model to be described later in this thesis will be shown to have a nonlinear objective function, along with the two types of vehicle capacity constraints: weight and volume. We have seen that a number of earlier model formulations capture various aspects of the above model; however, no single formulation contains all of the relevant features. For example, we have seen linear models with one or two types of capacity constraints. We have also seen nonlinear models with one type of capacity constraint; but these involved either strictly concave or strictly convex objective functions, which is not necessarily the case with our model. Mixed integer models could capture all of the necessary features, but the number of possible combinations of integer values would render all but the smallest problems unsolvable. In addition, restricting vehicle flow (capacity) variables to integer values is highly questionable in our model. Only Klessig's nonlinear model formulation appears general enough to handle all of the necessary features. However, the result of his solution algorithm is a local optimum which may or may not be very good in a global sense. Some authors [6,7] have developed techniques to achieve a better form of local optimum than one which is merely optimal in a marginal sense. However, these have only been made operational in the case of strictly concave cost single commodity networks. One can conclude that more specialized techniques are desirable.

3. MATHEMATICAL ANALYSIS

The multicommodity multiattribute network flow problem has a mathematical structure fundamentally different than other min-cost network flow problems. This is due to two related factors. The first of these is the presence of two types of commodity attributes: attributes for which costs increase linearly (weight and volume), and an attribute for which costs increase nonlinearly (inventory holding cost). The second factor is the presence of multiple commodities that may share an arc. In a single commodity environment, the first factor would lead to an arc cost function that is strictly concave with respect to a commodity flow variable. With multiple commodities, cost functions may be either concave or convex with respect to commodity flow variables, depending upon the relationships between the attributes of the incremental commodity flow, and the attributes and flow levels of the other commodities sharing the arc. Therefore, the solution techniques that apply to network flow problems with strictly linear, convex, or concave cost functions cannot be applied without modification. This section of the thesis will explore the mathematical properties of the problem in detail, giving us some insight into possible solution strategies.

3.1 Problem Description

As discussed in the Introduction, the transportation network has three sets of nodes: sources (origins), sinks (destinations), and intermediate nodes (transshipment terminals); these sets being denoted O, D, and T, respectively. Vehicles carry loads of commodities along the (one way) paths of the network. These paths start in O, end in D, and may include a maximum of one node in T. The model does not make any specific provision for vehicles once they reach the end of an arc; they might either return empty, or be rescheduled for another load. However, this will be treated as a separate problem beyond the scope of this model.

A demand will be defined as a specific number of weight loads of a commodity, to be shipped between a specific origin-destination pair, at a constant rate over time.[1] The vehicles carrying the commodities have identical capacity, and for each given arc, will have constant inter-departure times. (These assumptions on the vehicles, while not crucial to the methodology developed in this thesis, will reduce the number of variables that need to be considered.) Thus, the model has no dynamic aspects, both commodity flow and vehicle flow being constant over time. The relationship between these two flows must be such that neither the volume capacity, nor the weight capacity of the vehicles will be exceeded. The individual shipping units (e.g. boxes) of the commodities are assumed to be small in relation to either capacity of the vehicles. This allows us to further assume that commodities can be fit into a vehicle with essentially little or no wasted space, thereby eliminating the need to analyze vehicle loading from a combinatorial point of view. Assuming the items to be small in relation to vehicle capacity also allows us to model the commodity flow variable as a non-integer variable with no major loss of accuracy.

Two types of costs are involved in the model formulation: vehicle costs and inventory holding costs. Inventory holding costs are assessed on a unit of commodity as soon as it is "made available" at an

---

1. A weight load is defined as the amount of commodity that will fill a vehicle to its weight capacity.

origin, and continue until it is delivered to its final destination. Because of the non-dynamic, continuous nature of the model formulation, it is convenient to calculate average values for inventory holding costs per unit time. The average time between vehicle departures on an arc is equal to the inverse of the vehicle flow rate, y_{ij}. We are assuming commodities are made available for shipping at a continuous rate, implying that a unit of commodity will wait on the average for one half of this time, or 1/2yij. We are further assuming that inventory holding costs are no longer accrued once the commodity reaches its destination, which is reasonable in this formulation if the vehicle unloading time is small relative to the vehicle interdeparture time. Thus, if the inventory holding cost/weight load/unit time for a commodity k is h^k, the average waiting costs per weight load for travel on arc (i,j) for commodity k is $h^k/2y_{ij}$. In addition, those inventory holding costs/weight-load that relate to vehicle travel time can be expressed as $t_{ij}h^k$, where t_{ij} is the amount of time required for a vehicle to travel the length of arc (i,j). The parameter t_{ij} might also include time required for material handling. Total inventory costs for each weight load of commodity k on arc (i,j) can then be expressed as $h^k(1/2y_{ij} + t_{ij})$.

Vehicle flow rates will also be treated as a continuous variable. This provides no conceptual difficulty if we allow non-integer interdeparture times, which is equivalent to having the ability to redefine our time units on each arc so that one vehicle leaves per unit time. Vehicle costs per unit time will be obtained on each arc by multiplying the vehicle flow rate by a constant that reflects the cost of loading and driving the vehicle the length of the arc.

3.2 Mathematical Programming Formulation

First we will define the following notation:

O is the set of origin nodes

D is the set of destination nodes

T is the set of terminal nodes

the variables are defined as:

x_{ij}^{pk} - the number of weight loads per unit time of commodity k on arc (i,j) between origin-destination pair p

y_{ij} - the number of vehicles per unit time on arc (i,j)

and the parameters are:

s^k - the density of commodity k in lbs./ft^3

h^k - the inventory holding cost per unit time per weightload of commodity k

s^0 - the weight capacity of a vehicle (lbs.) / the volume capacity of a vehicle (ft^3)

t_{ij} - the time required for a vehicle to traverse arc (i,j)

d^{pk} - the demand per unit time for commodity k between origin-destination pair p

Let X and Y represent solution vectors for total network commodity flow per unit time and total network vehicle flow per unit time, respectively. Then let:

F(X,Y) = total network costs per unit time given X,Y

We will seek to minimize F(X,Y) subject to demand, capacity, and conservation of flow constraints as follows:

$$\mathbf{P}: \quad \text{Minimize } F(X,Y) = \Sigma_{i,j}[C_{ij}y_{ij} + \Sigma_k\{h^k x^k_{ij}/2y_{ij} + t_{ij}h^k x^k_{ij}\}]$$

$$\text{s.t.} \quad \Sigma_j\, x^{pk}_{ij} = d^{pk} \qquad \text{for } i \in p, \text{ for all } p,k \tag{3.1}$$

$$\Sigma_i\, x^{pk}_{ij} = d^{pk} \qquad \text{for } j \in p, \text{ for all } p,k \tag{3.2}$$

$$\Sigma_j\, x^{pk}_{ij} - \Sigma_j\, x^{pk}_{ji} = 0 \qquad \text{for each } i \in T, \text{ for all } p,k \tag{3.3}$$

$$x^k_{ij} = \Sigma_p x^{pk}_{ij} \qquad \text{for all } i,j,k \tag{3.4}$$

$$\Sigma_k\, x^k_{ij} \le y_{ij} \qquad \text{for all } i,j \tag{3.5}$$

$$\Sigma_k\, s^0 x^k_{ij}/s^k \le y_{ij} \qquad \text{for all } i,j \tag{3.6}$$

$$x^{pk}_{ij} \ge 0 \quad , \quad y_{ij} \ge 0 \qquad \text{for all } i,j,p,k \tag{3.7}$$

Constraints (3.1) and (3.2) ensure that demand is met, while constraints (3.3) provide for conservation of flow through the consolidation terminals. Constraints (3.4) are definitional constraints that relate the individual flow variables to total commodity flow on each arc. Constraints (3.5) and (3.6) are weight and volume constraints, relating commodity flow to vehicle flow. Note that the units for the left-hand side of constraints (3.5) are weight loads, and the units for the left-hand side of constraints (3.6) are volume loads. Constraints (3.7) are the standard restrictions to nonnegative flow levels.

3.3 Analysis of the Arc Cost Function

The objective function of Problem P is a sum of the cost functions on each arc of the network. To gain further understanding of the underlying mathematics, it is useful to analyze the cost function for a given arc (i,j). We will first define the arc cost function, and then show how the function may take on three different forms, with the form depending upon which attribute - weight, volume, or inventory holding cost - "dominates" the current flow on the arc. We will relate the behavior of these functions to relationships between the flow attributes and determine necessary conditions for second order behavior of the functions. It will be convenient at this time to introduce some vector notation. Let:

$\mathbf{x}_{ij} = (x^1_{ij}, x^2_{ij}, \ldots, x^K_{ij})$ - the commodity flow vector for arc (i,j)

$\mathbf{h} = (h^1, h^2, \ldots, h^K)$ - the vector of inventory holding cost parameters

$\mathbf{s} = (s^0/s^1, s^0/s^2, \ldots, s^0/s^K)$ - the vector of unit conversion parameters, converting weight loads to volume loads

$\mathbf{1} = (1, 1, \ldots, 1)$ - a vector of unitary coefficients

The arc cost function, F_{ij}, has the same form on each arc, and can be expressed as

$$F_{ij}(\mathbf{x}_{ij}, y_{ij}) = C_{ij}y_{ij} + \mathbf{h}\mathbf{x}_{ij}/2y_{ij} + t_{ij}\mathbf{h}\mathbf{x}_{ij} \qquad (3.8)$$

Decomposition

F_{ij} is a strictly convex function of y_{ij}. Therefore, for a fixed level of $\mathbf{x}_{ij}$, the optimal unconstrained value for y_{ij} can be found directly by setting the derivative of F_{ij} with respect to y_{ij} equal to zero and solving for y_{ij}:

$$dF_{ij}(x_{ij},y_{ij})/dy_{ij} = C_{ij} - hx_{ij}/2y_{ij}^2 = 0$$

which implies that the optimal value, y_{ij}^*, is

$$y_{ij}^* = (hx_{ij}/2C_{ij})^{\frac{1}{2}} \tag{3.9}$$

This is very similar to the standard Economic Order Quantity formula of inventory theory. However, the capacity constraints

$$lx_{ij} \leq y_{ij}$$

$$sx_{ij} \leq y_{ij}$$

may not allow y_{ij}^* to be physically realizable. This implies that optimization of equation (3.8), subject to its capacity constraints, will either lead to a solution given by equation (3.9), or to a solution at a boundary of the feasible region. Therefore, the optimal feasible value for y_{ij} will be

$$y_{ij} = \max\{\ lx_{ij},\ sx_{ij},\ (hx_{ij}/2C_{ij})^{\frac{1}{2}}\ \} \tag{3.10}$$

Equation (3.10) provides some insight into the multiattribute characteristic of the model. The commodity flow can be viewed as being composed of three separate "flows": weight (lx_{ij}), volume (sx_{ij}), and "dollars" (hx_{ij}), where we consider the inventory holding cost per unit time to be a dollar flow, although not in the literal sense. The vehicle flow is dependent only upon the dominating "attribute flow". This can be seen by substituting the appropriate expression for y_{ij} in equation (3.10) into equation (3.8), thus obtaining a function $F_{ij}(x_{ij})$ dependent only upon the commodity flow variables x_{ij}^k. This function has three distinct forms, one for each of the three elements of the right-hand side of equation (3.10). When y_{ij} is equal to the first term in equation (3.10), we will say that the cost function is "weight dominated", and will denote the cost function by $F^{w}_{ij}(x_{ij})$. Similarly, when y_{ij} is equal to the second term in equation (3.10), we will say that the cost function is "volume dominated", and will denote the function by $F^{v}_{ij}(x_{ij})$. Finally, when y_{ij} is equal to the third term in equation (3.10), we will say that the cost function is "dollar dominated", and will denote the cost function by $F^{d}_{ij}(x_{ij})$. These three cost functions are expressed as

$$F^{w}_{ij}(x_{ij}) = C_{ij} 1 x_{ij} + hx_{ij}/2(1x_{ij}) + t_{ij}hx_{ij} \qquad (3.11)$$

$$F^{v}_{ij}(x_{ij}) = C_{ij} s x_{ij} + hx_{ij}/2sx_{ij} + t_{ij}hx_{ij} \qquad (3.12)$$

$$F^{d}_{ij}(x_{ij}) = (2C_{ij}hx_{ij})^{\frac{1}{2}} + t_{ij}hx_{ij} \qquad (3.13)$$

The above suggests a decomposition of the problem in a manner similar to that of Steenbrink [13]. That is, one can define a "master problem" that finds a cost minimizing commodity flow subject to constraints (3.1)-(3.4) and (3.7). The subproblem for each arc would provide the appropriate function from (3.11)-(3.13) for each level of flow, in effect, "optimizing" the capacity variable y_{ij} according to the flow level. However, before we discuss specific solution algorithms for the master problem, further analysis of the arc cost functions (3.11)-(3.13) is necessary.

Parameter Space

It will often be the case that as we increase or decrease the flow of a commodity on an arc, two, or perhaps three, different cost functions (3.11)-(3.13) will apply, depending upon the relationships of the three attributes of flow: weight, volume, and dollars. It is instructive to view this process of changing flows and cost functions as movement through a 3-dimensional "parameter space", with a different axis representing weight flow (lx_{ij}), volume flow (sx_{ij}), and "dollar flow" (hx_{ij}).

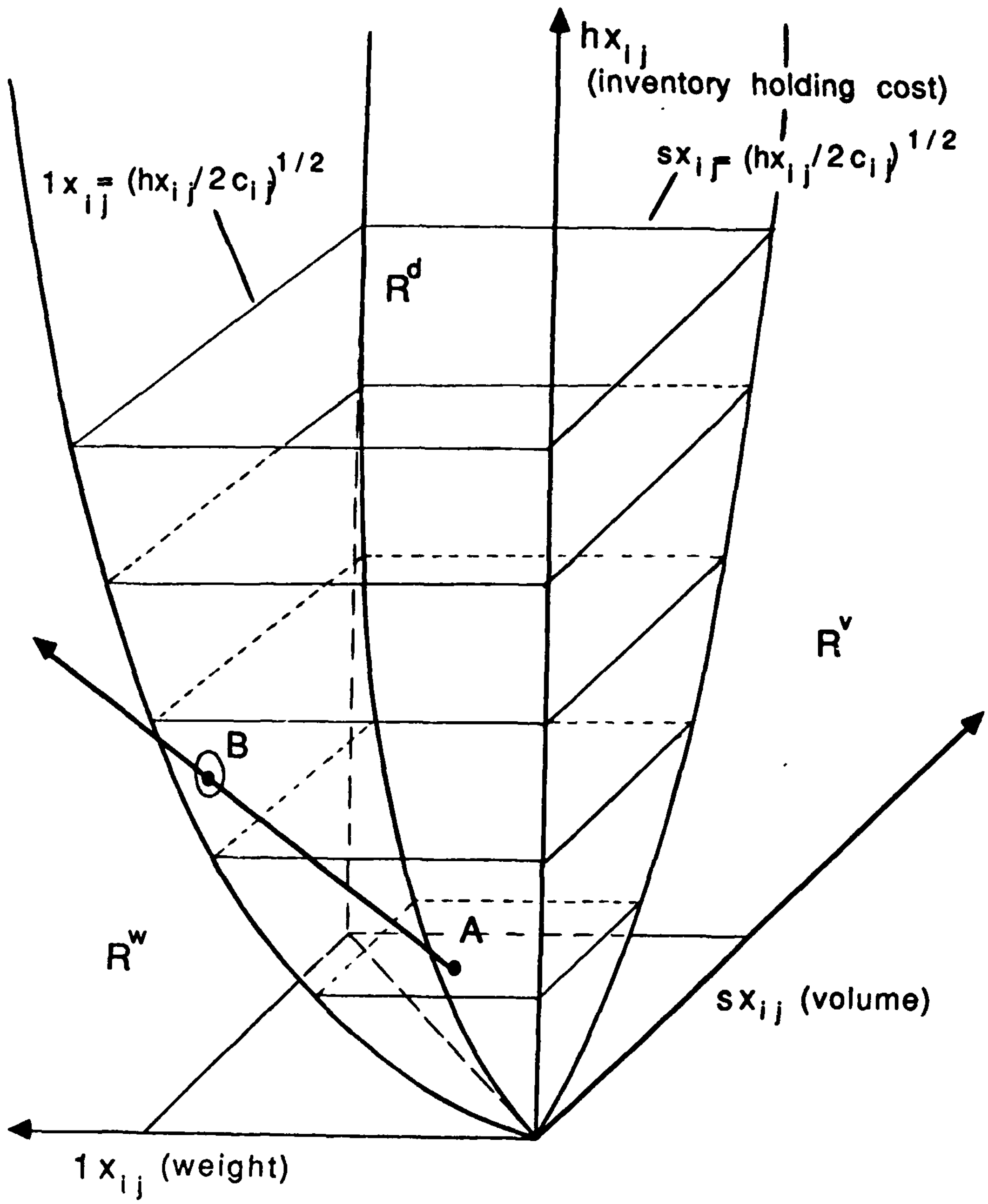
Observe the parameter space depicted in Figure 3.1. The space is divided into three regions, denoted R^w, R^v, and R^d. By knowing the values of the three flow attributes, we know in which region the current flow resides, and which of the three cost functions (3.11) - (3.13) is appropriate. That is, in region R^w, costs are weight dominated, and equation (3.11) will apply; in region R^v, costs are volume dominated and equation (3.12) will apply; while in region R^d, costs are dollar dominated and equation (3.13) will apply. The boundaries between the regions are defined by the set of points where two or more elements of equation (3.10) are equal. For example, the boundary between R^w and R^d is defined as the set of points where

$$lx_{ij} = (hx_{ij}/2C_{ij})^{\frac{1}{2}}.$$

The three flow regions have the following physical interpretation: region R^d represents combinations of flow attributes for which it is optimal to allocate more vehicle capacity than physical requirements dictate; region R^w represents combinations of flow attributes for which it is optimal to allocate vehicle capacity according to the weight of the loads; and region R^v represents combinations of flow attributes for which it is optimal to allocate vehicle capacity according to the volume of the loads. Any combination of commodity flows is represented by a point in the parameter space, while incremental changes in the

Figure 3.1

Parameter Space



flow of a given commodity can be represented by a line through this space. When this line crosses a boundary between regions, a different cost function applies. Thus, it is possible to think of $F_{ij}(x_{ij})$ as a composite cost function formed by concatenating the functions F^w_{ij}, F^v_{ij}, and F^d_{ij} at the points where the incremental flow line crosses the region boundaries in the parameter space.

As an example, observe ray $\overrightarrow{AB}$ in Fig. 3.1. It starts in region R^d, and at point B, crosses into region R^w. One possible composite cost function for this incremental flow is shown in Figure 3.2. Note how the form of the function changes when the incremental flow line reaches point B; it goes from being concave in region R^d to convex in region R^w. In the next section we will develop further understanding of this

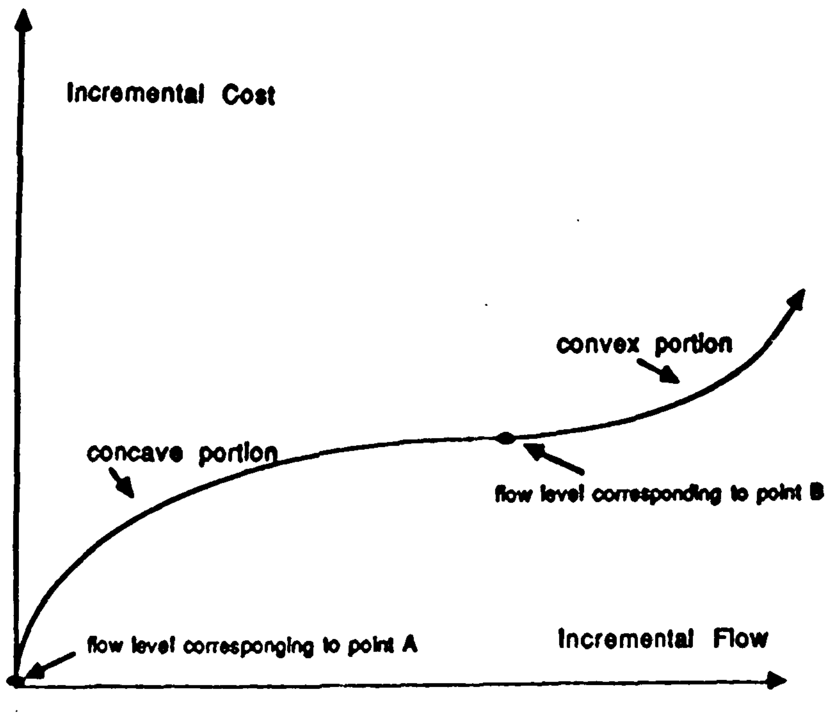


Figure 3.2

Incremental Costs Corresponding to Ray AB behavior by analyzing the derivatives of the three forms of arc cost functions.

First and Second Order Behavior of the Arc Cost Functions

We have seen how the arc cost function F_{ij} will, at any given point, have one of three forms, depending upon the relationships between the three flow attributes. In this section, we will examine the behavior of equations (3.11) - (3.13) when we allow a particular commodity flow variable, x^k_{ij} say, to vary. The flows of the other commodities will be held at some fixed level. In particular, we will derive and examine the first and second derivatives, with respect to x^k_{ij}, of the functions, define expressions for limiting values, and determine necessary conditions for concavity/convexity.

The dollar dominated function, F^d_{ij}, is simplest to analyze; being a sum of linear and square root terms, it will always be strictly concave; thus, there is no need to analyze the second derivative. In addition, limiting behavior is not particularly relevant. This is easily seen by recalling our parameter space of Figure 3.1. Suppose that our current flow is defined by point A in R^d. Any incremental flow with $h^k < \infty$ defines a ray, similar to ray $\overrightarrow{AB}$, that must eventually cross over into R^w or R^v. Therefore, for large values of the incremental flow variable, x^k_{ij}, the functions F^w_{ij} or F^v_{ij} will apply.

The situation is less clear cut when analyzing F^w_{ij} or F^v_{ij}. We will begin with the weight dominated function, F^w_{ij}. First, we will define the following notation. Let:

$$(F^w_{ij})'_k = \frac{dF^w_{ij}(x_{ij})}{dx^k_{ij}}, \text{ and}$$

$$x^{\backslash k}_{ij} = x_{ij} \quad \text{with } x^k_{ij} = 0 \quad (\text{i.e. } x^{\backslash k}_{ij} = \{x^1_{ij}, x^2_{ij}, \ldots, x^{k-1}_{ij}, 0, x^{k+1}_{ij}, \ldots x^K_{ij}\})$$

Then we calculate:

$$(F^w_{ij})'_k = C_{ij} + (h^k 1x_{ij} - hx_{ij})/2(1x_{ij})^2 + t_{ij}h^k,$$

and by cancelling the $h^k x^k_{ij}$ terms we have:

$$(F^w_{ij})'_k = C_{ij} + (h^k 1x^{\backslash k}_{ij} - hx^{\backslash k}_{ij})/2(1x_{ij})^2 + t_{ij}h^k \qquad (3.14)$$

A few observations can be made at this time. Note that x^k_{ij} only appears in the denominator of the second term of (3.14). This implies that the second term will get increasingly smaller as x^k_{ij} gets larger, which leads to

$$\lim_{x^k_{ij} \to \infty} (F^w_{ij})'_k = C_{ij} + t_{ij}h^k \qquad (3.15)$$

Since this is a constant, F^w_{ij} will tend to become linear for large flow values. Observe also, that in the single commodity case, the numerator of the second term in equation (3.14) will always be 0. This implies that, in this case, F^w_{ij} will be linear for all values of x^k_{ij}.

In investigating second derivative behavior, we define the additional notation:

$$(F^w_{ij})''_k = \frac{d^2F^w_{ij}(x_{ij})}{dx^k_{ij}}, \text{ and}$$

$$\bar{h}_w = hx^{\backslash k}_{ij}/1x^{\backslash k}_{ij} .$$

The term $\bar{h}_w$ is the ratio of dollar flow to weight flow of the fixed commodities, and can be interpreted as the average dollar value per weight load. Now,

$$(F^{w}_{ij})''_{k} = (hx^{\backslash k}_{ij} - h^{k}1x^{\backslash k}_{ij})/(1x_{ij})^{3}, \tag{3.16}$$

which implies that when

$$\boxed{\begin{array}{ll} h^{k} \leq \bar{h}_{w}, & F^{w}_{ij} \text{ is convex in } x^{k}_{ij}, \\ h^{k} = \bar{h}_{w}, & F^{w}_{ij} \text{ is linear in } x^{k}_{ij}, \text{ and} \\ h^{k} \geq \bar{h}_{w}, & F^{w}_{ij} \text{ is concave in } x^{k}_{ij}. \end{array}} \qquad (\,*\,)$$

This is a very useful set of results in that the whole question of second derivative behavior is reduced to a comparison of two simple ratios. Recall that the relationships (*) apply to an incremental cost function, where we vary the level of only one commodity. In flow region R^{w}, we then have the following physical interpretation: when the dollar value per weight load of a commodity is less than or equal to the average dollar value per weight load of the current flow on an arc, the arc cost function will be convex with respect to incremental changes in the flow of the added commodity. Similar statements can be made regarding linear and concave cost functions for the cases where the dollar value per weight load of a commodity is equal to, or greater than or equal to, the average dollar value per weight load. However, *the convexity condition is unique to the multicommodity, multiattribute formulation, and will not occur in single commodity problems, nor in multicommodity problems that don't involve inventory costs and capacity constraints.*

The volume dominated function, F^v_{ij}, can be analyzed in a similar fashion. Let:

$$(F^v_{ij})'_k = \frac{dF^v_{ij}(x_{ij})}{dx^k_{ij}}, \qquad (F^v_{ij})''_k = \frac{d^2F^v_{ij}(x_{ij})}{dx^k_{ij}}, \text{ and}$$

$$\bar{h}_v = hx^{\backslash k}_{ij}/sx^{\backslash k}_{ij} .$$

The term $\bar{h}_v$ is the ratio of dollar flow to volume flow of the fixed commodities, and can be interpreted as the average dollar value per volume load. In this case, it can be shown that

$$(F^v_{ij})'_k = C_{ij}s^0/s^k + (h^k sx^{\backslash k}_{ij} - s^k hx^{\backslash k}_{ij})/2(sx_{ij})^2 + t_{ij}h^k \quad (3.17)$$

$$\lim_{x^k_{ij} \to \infty} (F^v_{ij})'_k = C_{ij}s^0/s^k + t_{ij}h^k \quad (3.18)$$

$$(F^v_{ij})''_k = s^0/s^k((s^0/s^k)hx^{\backslash k}_{ij} - h^k sx^{\backslash k}_{ij})/(sx_{ij})^3 \quad (3.19)$$

which implies that when

$h^k/(s^0/s^k) \le \bar{h}_v$,	F^v_{ij} *is convex in* x^k_{ij},	
$h^k/(s^0/s^k) = \bar{h}_v$,	F^v_{ij} *is linear in* x^k_{ij}	(**)
$h^k/(s^0/s^k) \ge \bar{h}_v$,	F^v_{ij} *is concave in* x^k_{ij}.	

These conditions have a physical interpretation analogous to the case with F^w_{ij}. Here we can say that, in flow region R^v, if the dollar value per volume load of an added commodity is less than or equal to the average dollar value per volume load of the current flow on the arc, the arc cost function will be convex with respect to incremental changes in the flow of the added commodity. As before, similar statements can be made regarding linear and concave costs for the cases where the dollar value per volume load of the commodity is equal to, or greater than or equal to the average dollar value per volume load of the current flow.

Graphical Analogies to Function Behavior

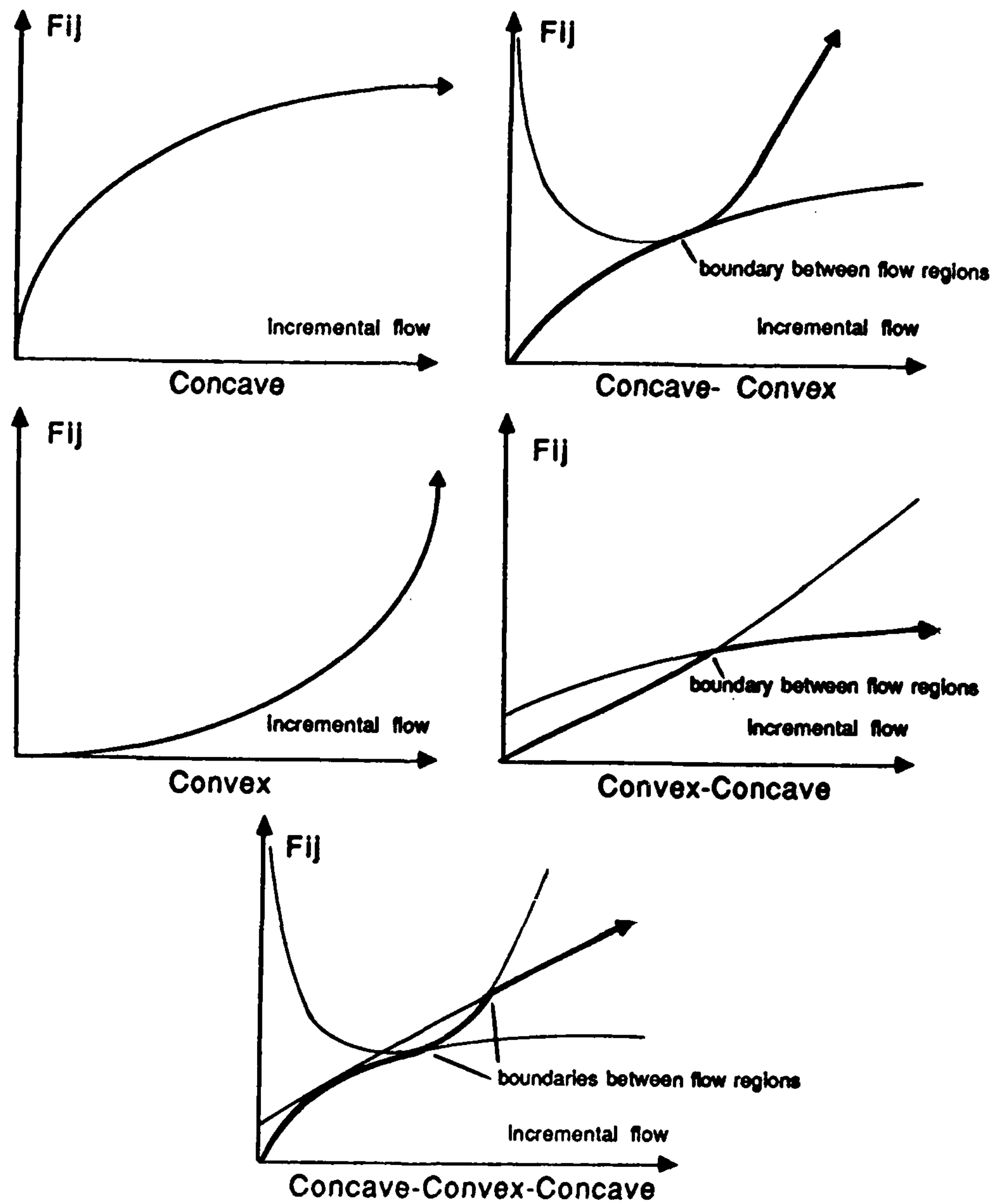
We have seen how the nonlinearity of F^w_{ij} and F^v_{ij} depends upon the relationships between the parameters of current and incremental flow. These relationships can be easily captured in the framework of the parameter space of Figure 3.1 by noting that h^k, $h^k/(s^0/s^k)$, $\bar{h}_w$, and $\bar{h}_v$ can be represented by the slopes of certain lines through the parameter space. For example, suppose that a line connects the origin of the space with the point defined by the current commodity flow (e.g. point A in Figure 3.1). The slope of this line with respect to the dollar-weight axes is $\bar{h}_w$, while the slope of this line with respect to the dollar-volume axes is $\bar{h}_v$. We can interpret the line representing incremental flow ($\overrightarrow{AB}$ in Figure 3.1) in a similar fashion: the slope of the incremental flow line with respect to the dollar-weight axes is h^k, while the slope of the incremental flow line with respect to the dollar-volume axes is $h^k/(s^0/s^k)$. Thus, the second order behavior of the arc cost function can be determined by comparing the slopes of the lines associated with current and incremental flows.

Note that our composite cost function, $F_{ij}(x_{ij})$, may consist of up to three function segments, with a separate segment for each region in which the incremental flow line lies. There may be one segment that is concave, corresponding to $F^d_{ij}(x_{ij})$, and up to two other segments that may be linear, convex, or concave, corresponding to $F^w_{ij}(x_{ij})$ and $F^v_{ij}(x_{ij})$. Figure 3.3 illustrates the five basic forms of F_{ij} that can be obtained by extending a line from one point in the parameter space to another. The form of the arc cost function will depend upon the point defining the initial flow, the region boundaries crossed by the incremental flow line, and the slope of the incremental line. For example, an incremental flow line that begins in R^d, ends in R^w, and has slope $h^k < \bar{h}_w$ will correspond to a composite arc cost function that is first concave and then convex. However, if $h^k \geq \bar{h}_w$, the line would correspond to a concave arc cost function.

The parameter space concept is also useful in determining the precise range of x^k_{ij} values over which each function will apply. First note that the sign of the relevant inequality of the necessary conditions will not change as we move along the incremental flow line in a particular flow region. For example, if $h^k < \bar{h}_w$, increasing the proportion of commodity k flow (moving outward on the incremental flow line) will decrease the average, $\bar{h}_w$, but can never make it become less than or equal to h^k. Therefore, the inequality will remain unchanged, and the second derivative condition will remain in effect throughout a given flow region. *The only points at which the second derivative condition will change is where the incremental flow line enters another region of the parameter space.* When this occurs, a different set of

FIGURE 3.3

The Five Basic Forms of the Arc Cost Function Fij



ratios must be compared. This greatly simplifies the problem of determining the precise range of values over which a functional form will apply. Instead of resorting to complicated numerical analysis, *the inflection points of the arc cost function F_{ij} can be determined through simple geometric calculations in the parameter space.*

3.4 Bounds on the Degree of Nonlinearity

We have seen that the functions F^w_{ij} and F^v_{ij} may be nonlinear, but it is unclear at this point if this poses any practical problems. For example, we have seen that the degree of nonlinearity is determined largely by the differences between values of inventory holding cost parameters. We might wish to know how much deviation in these values is allowed before nonlinearity becomes significant. Perhaps by assuming that these parameters fall within a certain range, we could assume that F^w_{ij} and F^v_{ij} are linear, especially if the flow values are large enough. If this proved to be the case, our composite cost function would be a combination of concave (F^d_{ij}) and linear functions and would then itself be concave. Standard techniques of concave cost network flow models could then be applied.

This section will develop a means of analyzing the cost functions to determine if, in practice, F^w_{ij} or F^v_{ij} have significant levels of nonlinearity. We will confine our analysis to F^w_{ij} since less parameters are involved, thereby making the calculations less cumbersome. However, the conclusions we obtain can be applied to F^v_{ij} as well. We will proceed from the viewpoint of having a known collection of commodities, with given parameters, that are to be sent through the transportation network. We wish to determine if the set of parameters is such that some feasible combinations of flow would produce significant nonlinearity. We will only analyze the situation where F^w_{ij} is convex, that is, $h^k \leq \bar{h}_w$. This implies no loss of generality since, on each arc, at least one inventory holding cost parameter must be less than or equal to the average inventory holding cost on that arc. The analysis will consider two separate cases, one corresponding to the situation where the current flow on an arc is in region R^d, and the other where the current flow is in region R^w. The reason for this distinction will be made clear in the analysis to follow.

Case 1: current flow in region R^w $(1x^{\backslash k}_{ij} \geq (hx^{\backslash k}_{ij}/2C_{ij})^{\frac{1}{2}}$, $sx^{\backslash k}_{ij})$

In this situation the addition of commodity flow, x^k_{ij} say, will cause costs to increase in a strictly convex fashion. That is, our composite cost function for incremental flow will consist only of $F^w_{ij}(x_{ij})$, which will be strictly convex by our initial assumption that $h^k \leq \bar{h}_w$. The point on this function with greatest convexity (largest second derivative) will be where the incremental flow is set to 0. It is at this point that we wish to establish a bound on nonlinearity.

The bound to be used is derived from the first derivative. Let $(F^w_{ij})'_k|_\infty$ be the limiting value of the first derivative defined by equation (3.15). Our bound, B^w, will be expressed as:

$$B^w = 1 - [(F^w_{ij})'_k / (F^w_{ij})'_k|_\infty] \tag{3.20}$$

Since F^w_{ij} is convex and strictly increasing, $(F^w_{ij})'_k < (F^w_{ij})'_k|_\infty$, which implies that $0 < B^w < 1$. If B^w is "small", then we can say that

F^w_{ij} is "nearly linear". The largest value of B^w is found where $x^k_{ij} = 0$. Thus, by inserting $x^k_{ij} = 0$ into (3.14), and substituting (3.14) and (3.15) into (3.20) we have

$$B^w \leq 1 - [C_{ij} + (h^k 1x^{\setminus k}_{ij} - hx^{\setminus k}_{ij})/2(1x^{\setminus k}_{ij})^2 + t_{ij}h^k] \;/\; [C_{ij} + t_{ij}h^k]$$

In appendix 1A, simplification of the above equation results in:

$$B^w < (hx^{\setminus k}_{ij} - h^k 1x^{\setminus k}_{ij}) \;/\; 2C_{ij}(1x^{\setminus k}_{ij})^2$$

(This bound is nearly exact at $x^k_{ij} = 0$ if we assume that $t_{ij}h^k << C_{ij}$) Suppose that we wish B^w to be no greater than some value α. This can be accomplished by setting

$$B^w < (hx^{\setminus k}_{ij} - h^k 1x^{\setminus k}_{ij}) \;/\; 2C_{ij}(1x^{\setminus k}_{ij})^2 \leq \alpha$$

which leads to the following compact but important result:

$$B^w \leq \alpha \quad \equiv \quad h^k/\bar{h}_w \geq (1 - \alpha(r^{kw}_{ij})^2) \tag{3.21}$$

where $$r^{kw}_{ij} = 1x^{\setminus k}_{ij}/(hx^{\setminus k}_{ij}/2C_{ij})^{\frac{1}{2}} \geq 1 .$$

By our earlier assumption that $h^k < \bar{h}_w$, we know that h^k/h_w must be less than 1, and is therefore bounded on either side. Note, however, that for some combinations of α and r^{kw}_{ij}, the right hand side of (3.21) will be negative. This implies that assuming linearity at the "α level" does not restrict inventory holding cost parameters. This is more likely to occur when r^{kw}_{ij} is large, which can be physically interpreted as a "heavy" flow of low dollar value commodities. On the other hand, many "reasonable" values of these parameters will be such that only very small differences in inventory holding cost parameters would be tolerated. For example, observe the situation depicted in Figure 3.4, a two-dimensional cross section of the parameter space of Figure 3.1. Here r_{ij}^{kw} is equal to 2.0; that is, the number of weight loads is twice that given by the R^d-R^w boundary at the current dollar flow level. When α is set to .05, the left hand side of equation (3.21) is constrained to be greater than .8, but less than 1.0. Equivalently, for r_{ij}^{kw} = 2.0, no commodity on arc (i,j) may have a dollar value per weight load that differs from the average dollar value per weight load of the other commodities on the arc by more than 20%. On the other hand, if r_{ij}^{kw} was set at 5.0, and α remained the same, there would be no implied restriction on inventory holding cost parameters. In summary,

equation (3.21) becomes more relevant in network flow problems where solutions with low values of r_{ij}^{kw} have high probability. This is most likely to be the case when commodity weights are low, inventory holding cost parameters are high, and demands are small.

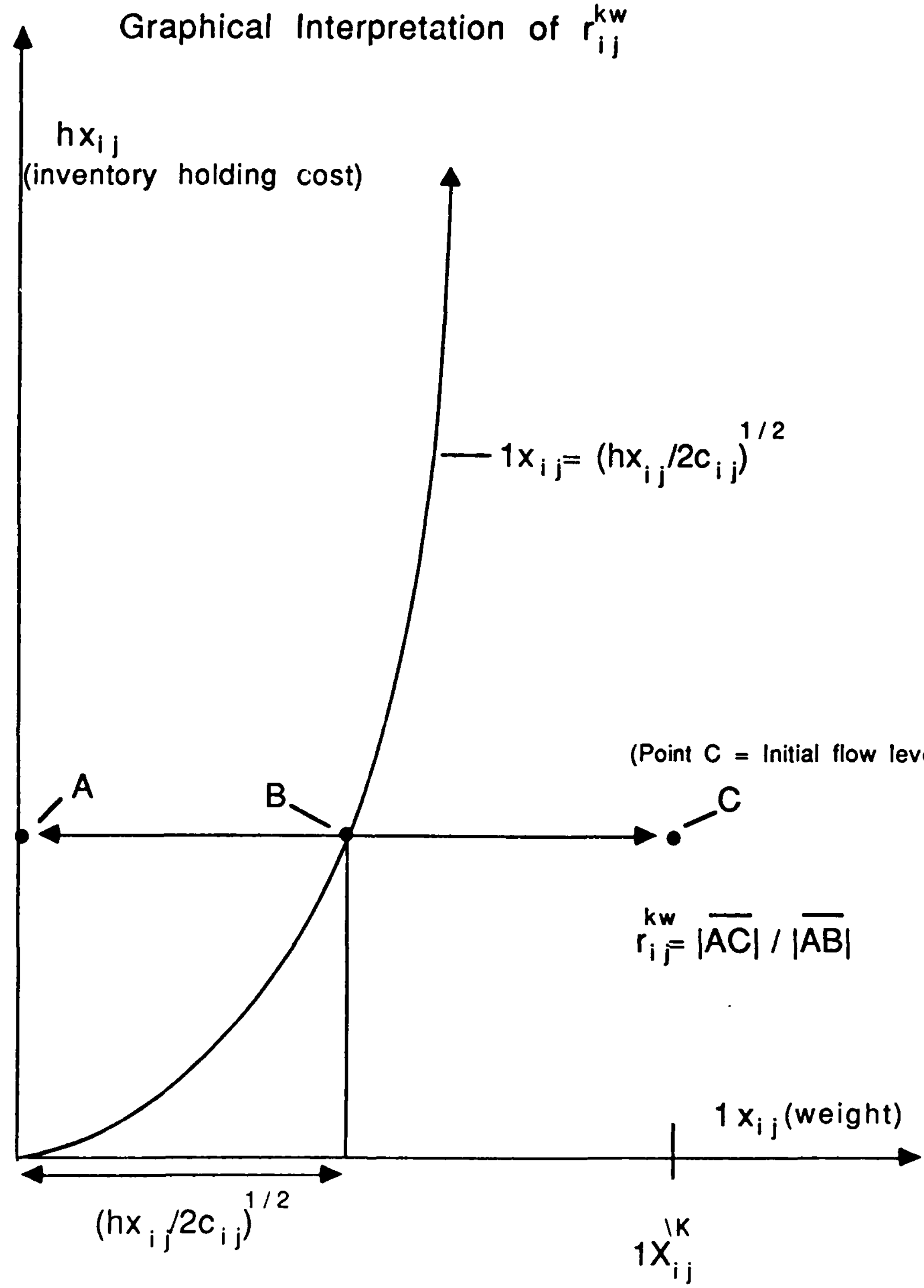
Case 2: current flow in region R^d $(1x_{ij}^{\backslash k},\ sx_{ij}^{\backslash k} \leq (hx_{ij}^{\backslash k}/2C_{ij})^{\frac{1}{2}})$

In this situation our current flow is in R^d, but by adding enough incremental flow, x_{ij}^k, we will cross the boundary into R^w. The composite cost function for incremental flow will consist of portions of F_{ij}^d and F_{ij}^w. The point on the composite function with greatest convexity will be at the boundary between the two cost regions. Therefore, we must evaluate the bound B^w at the point where

$$1x_{ij}^{\backslash k} + x_{ij}^k = [(hx_{ij}^{\backslash k} + h^k x_{ij}^k) / 2C_{ij}]^{\frac{1}{2}}$$

Figure 3.4

Graphical Interpretation of r_{ij}^{kw}



In this case the bound is considerably more difficult to derive since x_{ij}^{k} must now be solved in the above expression through the quadratic formula. In appendix 1B, the above expression is shown to imply

$$1x_{ij}^{\backslash k} + x_{ij}^{k} > [\ (h^{k}/2C_{ij})^{2} - 1x_{ij}^{\backslash k}/2C_{ij}(h^{k} - \bar{h}_{w})\ ]^{\frac{1}{2}},$$

which after simplification is shown to further imply

$$B^{w} < [\ 1x_{ij}^{\backslash k}(\bar{h}_{w} - h^{k})\ ] \ / \ [\ (h^{k})^{2}/2C_{ij} - 1x_{ij}^{\backslash k}(h^{k} - \bar{h}_{w})\ ] \qquad (3.22)$$

Suppose that, as before, we wish to limit B^{w} to being no greater than α . In this case:

$$\boxed{B^{w} \leq \alpha \ \equiv \ h^{k}/\bar{h}_{w} \geq [\ 1 - (\alpha/(1-\alpha))(r_{ij}^{kd})^{2}\ ]} \qquad (3.23)$$

where

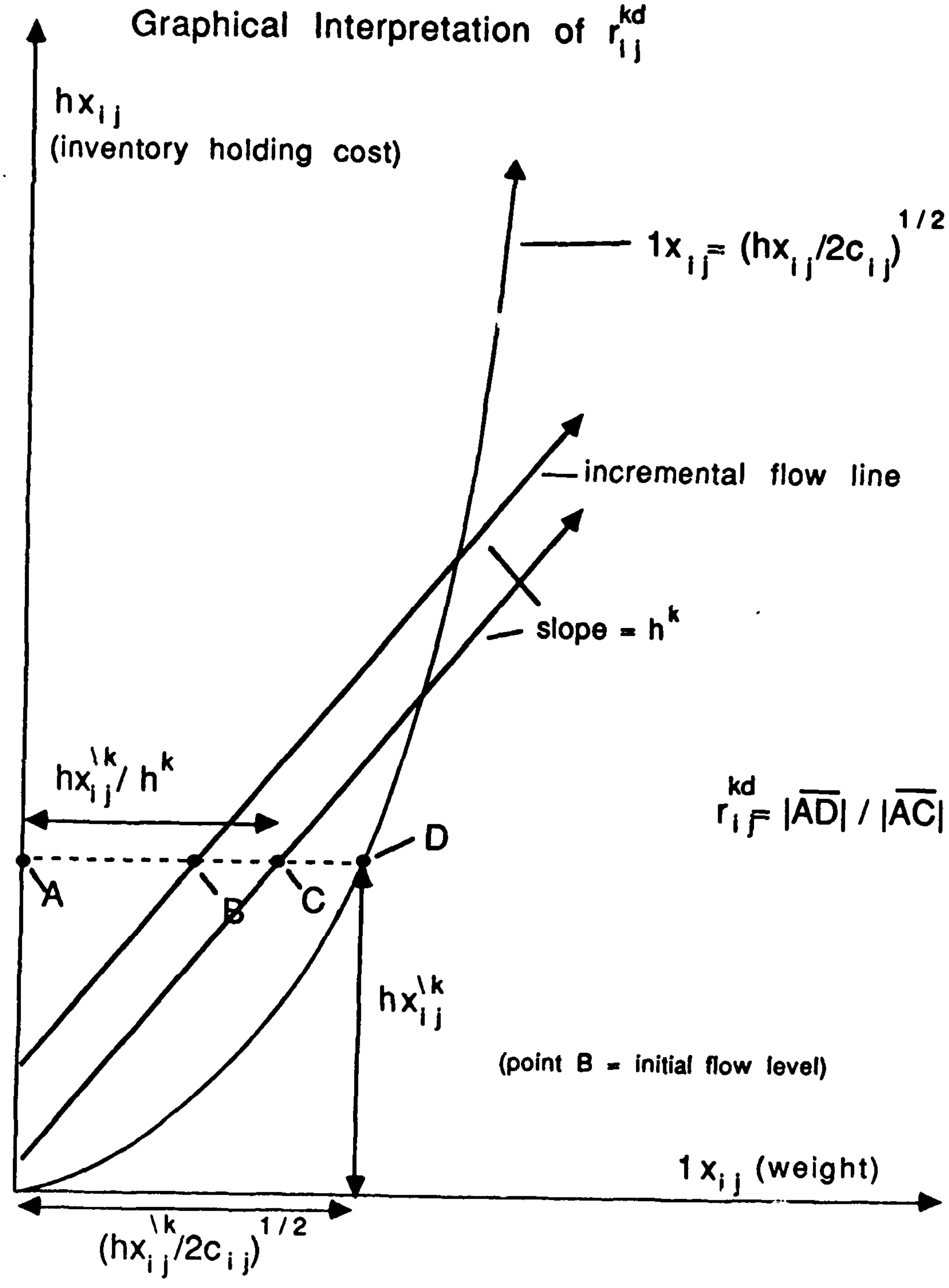
$$r_{ij}^{kd} = (hx_{ij}^{\backslash k}/2C_{ij})^{\frac{1}{2}} \ / \ (hx_{ij}^{\backslash k}/h^{k})$$

In this case the "r term" is quite different, but can also be expressed graphically in our parameter space. In Figure 3.5 we see the R^{d}-R^{w} boundary point, D, for our current dollar flow level. The term h^{k} is the slope of the line representing incremental flow, while the term $hx_{ij}^{\backslash k}$ represents the initial dollar flow level. The r term can again be seen to represent a multiple of the R^{d}-R^{w} boundary. In this case, however, r may be above or below the value 1.0, and will decrease with h^{k}. Thus, the effects of deviation in inventory holding cost parameters become more difficult to avoid.

A final note is in order regarding the relationship between r_{ij}^{kd} and r_{ii}^{kw} . It is apparent from Figure 3.5 that the limiting value of

Figure 3.5

Graphical Interpretation of r_{ij}^{kd}



$hx_{ij}^{\backslash k}/h^k$ is $lx_{ij}^{\backslash k}$, corresponding to the case where $h^k = \bar{h}_w$. In this situation, r_{ij}^{kd} has the same form as the reciprocal of r_{ij}^{kw} . Further comparisons can not be made, however, since here we are assuming that $lx_{ij}^{\backslash k} \leq (hx_{ij}^{\backslash k}/2C_{ij})^{\frac{1}{2}}$ whereas in case 1 we assumed the opposite.

3.5 Summary

The multiattribute multicommodity network flow problem was formulated as a mathematical program, Problem P, with certain assumptions regarding the network structure. The objective function of Problem P is comprised of a summation of the individual arc cost functions, F_{ij}, which are actually composites of up to three different functions: F_{ij}^w, F_{ij}^v, and F_{ij}^d. Each of these three separate functions apply within a particular region of the parameter space; this space being three-dimensional, and each point representing a possible commodity flow per unit time of a given weight, volume, and dollar value.

The function F_{ij}^d is strictly concave, but F_{ij}^w and F_{ij}^v may be either concave, linear, or convex, depending upon the relative values of easily derived parameter ratios. Thus F_{ij} is a general nonlinear function, but with a constant functional form for incremental flow within a particular region of the parameter space. The convexity condition is unique to the multicommodity multiattribute formulation, and will not occur in single commodity problems, nor in multicommodity problems that don't involve inventory holding costs and capacity constraints. Since incremental flow can be represented by a line through the parameter space, the inflection points of the function F_{ij} may be determined through straightforward geometric calculations.

We have also seen that nonlinearity can not be dismissed as only a theoretical problem. Assuming F^{w}_{ij} or F^{v}_{ij} to be linear could lead to quite inaccurate results and suboptimal solutions.

4. LOCAL OPTIMA

It was shown in Chapter 3 that a commodity's incremental arc cost function may contain concave and/or convex portions, depending upon the attributes and current flow values of the commodities on the arc. This feature implies that the problem defined in this thesis is at least as difficult as finding the global optimum in strictly concave networks -- a cost structure in which one could at least be assured that the optimal solution would be of the "all-or-nothing" type. Unfortunately, the number of possible solutions of this type grows exponentially with the size of the network. With the presence of arc cost functions that may contain both convex and concave portions, the all-or-nothing necessary condition need not apply, thus enlarging the set of candidate solutions even further.

The mathematical difficulties discussed above imply that finding (or verifying) a global optimum for our problem would be very difficult in a network of any practical size. One strategy, used previously to attack large min-cost concave network flow problems, has been to decompose the network into smaller independent subnetworks, for which the min-cost flow problem can be optimally solved much more readily (Blumenfeld et.al.[1], Hall [7]). However, this strategy depends upon certain assumptions that may not apply in our problem, such as assuming that cost is linear on arcs that normally carry large flows. Chapter 3 demonstrated that nonlinearity is significant in our problem, even at relatively high flow levels. In addition, the decomposition strategy also relies upon there being a finite set of discrete choices of flow values on arcs that share flow from more than one origin-destination pair. This is only true when all the arc cost functions are concave, but this is clearly not the case in our problem when more than one commodity is involved. Other means by which to decompose the problem of this thesis, that don't rely upon the above assumptions, have not as yet been discovered.

In Chapter 2 we saw that a number of researchers have considered it sufficient, especially in larger networks, to find a local optimum to the minimum-cost concave network flow problem that is "good" in some sense. To find such local optima, the basic strategy usually involves starting with some initial feasible solution, and then making a series of local improvements until some final criterion of local optimality is met. This strategy avoids the computational difficulties discussed above as inherent to a global optimization strategy. Therefore, since no decomposition techniques have been determined to be appropriate, the research effort in this thesis has also been directed towards finding "good" local optima.

Three issues are central to the local optimization process: finding an efficient means of converging to a local optimum, choosing a "good" starting solution, and finding a means of verifying that the local optimum is truly "good" in some sense. Though the issues are not separable, we will initially discuss some of the aspects of the first issue -- finding an efficient means of converging to a local optimum. We will then relate local optima to the concept of a "good" local optimum through the use of what shall be referred to as "adjacent concave flows". An algorithm will be presented that has the capability to efficiently find such "good" local optima for the problem defined in this thesis. The issue of a "good" starting solution will be dealt with in Chapter 5 of this thesis, Computational Results.

4.1 Marginal Definition of a Local Optimum

To formalize the concept of a local optimum, it is necessary to introduce some additional notation. Let:

$F(X) = \Sigma_{ij}F_{ij}(x_{ij},y_{ij}(x_{ij}))$ - the objective function of the network flow problem expressed as a function of the commodity flow vectors x_{ij}, and the vehicle flow vectors, $y_{ij}(x_{ij})$, as defined in equation (3.10)

$F'(X)$ = the gradient vector of $F(X)$, representing the partial derivatives of $F(X)$ with respect to the commodity flow variables on each arc

Ω = the set of constraints (3.1)-(3.4),(3.7)

Note that since y_{ij} can be defined in terms of x_{ij}, we can eliminate the y_{ij} variables from our formulation in a fashion similar to Steenbrink's decomposition technique [13]. The problem P of Chapter 3 can then be rewritten as:

$$\mathbf{P'}: \quad \text{Minimize } F(X) \quad \text{s. t. } X \in \Omega$$

$F(X)$ is a summation of functions F_{ij}, each of which, as was shown in Chapter 3, is a composite of arc cost functions of the type (3.11) - (3.13). The arc cost functions for the weight and volume dominated situations, (3.11) and (3.12) respectively, were shown to have continuous first derivatives (equations (3.14) and (3.17)) and the same can easily be shown for (3.13). The optimality condition that we shall define shortly requires that $F(X)$ itself be continuously differentiable in the sense that a first derivative is defined at every point.[1]

The constraints Ω ensure that the feasible region for X is bounded. More specifically, it is a flow polyhedral, and is thus compact and convex. This fact, combined with our assumption of continuous differentiabilty, allows us to define the following necessary condition of local optimality:

Proposition: If $\hat{X}$ solves P', then

$$\min \{ \langle F'(\hat{X}), X-\hat{X} \rangle \mid X \in \Omega \} = 0 \quad (4.1)$$ [2]

If F(X) consisted only of convex arc cost functions, there would be only one local optimum, the global optimum, and the necessary condition (4.1) would then become sufficient for global optimality.

The Conditional Gradient Method

In Klessig [10], a version of the Franke-Wolfe algorithm called the Conditional Gradient Method (CGM) is described which seeks a local optimum meeting the condition (4.1). The CGM is meant to be applied to nonlinear multicommodity flow problems meeting the two conditions above: 1) the objective function is continuously differentiable, and 2) the constraint space is compact and convex. Therefore, the CGM could be applied to P' to find a local optimum of the form (4.1).

---

1. The points at which the functions (3.11)-(3.13) intersect in forming each F_{ij}, though technically discontinuities, present us with no difficulty in this regard. We can merely assume that at these intersection points, each F_{ij} is defined by the cost function (3.11)-(3.13) in the flow region having higher costs.

2. The notation <-,-> will denote the inner product of the two vectors indicated within the brackets.

The CGM is an iterative algorithm that maintains feasibility with successively lower cost at each iteration. The algorithm moves from one solution to the next by moving in the direction defined by the current gradient vector. A line search sequentially examines solutions in the direction of the gradient, starting with the solution at the limit of the feasible region, and moving back towards the current solution, until one is found that meets specific cost reduction criteria. This then becomes the current solution. The algorithm halts when condition (4.1) is satisfied. Below is a formal statement of the algorithm as applied to problem P'.[1]

Algorithm I

Step 0 Select any X_0 belonging to Ω . Set $i = 0$. Select values for α and β such that $0 < \alpha, \beta < 1$. (α and β are tuning parameters that can be adjusted as the problem requires.)

Step 1 Compute any $\tilde{X}_i$ belonging to Ω satisfying

$\langle F'(X_i), \tilde{X}_i \rangle = \min \{ \langle F'(X_i), X \rangle \mid X \in \Omega \}$

($\tilde{X}_i$ defines the limit of the feasible region in the direction of the gradient.)

Step 2 If $\langle F'(X), \tilde{X}_i - X_i \rangle = 0$, STOP ; X_i satisfies the necessary condition of (4.1) . Otherwise compute the smallest nonnegative integer k such that

$$F(X_i + \beta^k(\tilde{X}_i - X_i)) - F(X_i) - \alpha\beta^k \langle F'(X_i), \tilde{X}_i - X_i \rangle \leq 0 \qquad (4.2)$$

---

1. The parameter α used in Algorithm I should not be confused with the α used in Chapter 3, they are not related in any way.

Step 3 Set $X_{i+1} = X_i + \beta^k(\tilde{X}_i - X_i)$

Step 4 Set i = i + 1 and go to *Step 1*

A few observations can be made about this algorithm. In the case of Problem P', Step 1 reduces to a simple shortest path problem, with arc costs given by the gradients when evaluated at the current flow levels. In a model with a more complex constraint space, such as those models involving fixed arc capacity constraints, this step would require solving a linear program. This also would have been the case in this model had we not decomposed the problem into the form P'.

The line search of Step 2 is of finite duration in that there will always be some finite integer k that satisfies the inequality (4.2). This line search procedure is desirable in that it does not require the storage of solutions or a time consuming one dimensional minimization. On the other hand, although solution improvement is guaranteed, the effects of the β^k factor cause the line search to have the property that a solution may be selected that has a higher objective value than solutions examined and discarded earlier in the line search. The performance of the line search will also depend upon the values selected for the parameters α and β. Klessig [10] suggests values of $\alpha = \beta = \frac{1}{2}$. In general, larger values of β will cause the line search to *examine* solutions that are further from the current solution than would be the case for smaller values of β. On the other hand, larger values of α will make it more likely that the line search will *accept* solutions that are further from the current solution than would be the case for smaller values of α. Consequently, the selection of "correct" parameter values involves weighing the tradeoff between the number of iterations of the CGM and the work per iteration.

Convergence will occur in that either the algorithm will generate a finite series of solutions X_i whose last element is a local optimum, or the algorithm will generate an infinite series of solutions X_i that will, in the limit, approach a local optimum. [10 pg. 347]

Klessig provides no performance data for the algorithm, but its performance is likely to be highly dependent upon the particular problem, and the structure of its objective function. We could apply the algorithm directly to P' and obtain local minima. However, this would not capitalize upon any of our knowlege regarding the structure of F(X). In addition, there would be no guarantee of achieving a "good" local optimum.

4.2 Adjacent Concave Flows

Algorithm I of the previous section is sufficient for finding a local optimum of the form (4.1). However, we are interested in obtaining not just any local optimum, but one that is "good" in relation to other nearby local optima. For this reason, we will introduce the concept of "adjacent concave flows". This concept is similar to Gallo and Sodini's [6] "adjacent extreme flows" and Hall's [7] "incremental flows"; both of which are applied to concave cost networks. This section will define the concept of adjacent concave flows and give the rationale for its use.

Hall defines an optimality criterion that requires that cost can not be reduced by switching flow between an origin-destination pair from its assigned path to an alternative path. The attainment of this state occurs by searching over all possible extreme flows. This is possible in certain network structures by examining only those arcs with "shared flow". Gallo and Sodini use a similar concept, but in a more generalized network structure. They provide an algorithm that, starting from an extreme flow, finds all adjacent extreme flows. These adjacent extreme flows are the result of rerouting the flow between certain paths, or equivalently, a basis can be found for the old and new flow matrices that differ by only one column. If no adjacent extreme flow has a lower cost, the solution is a local optimum.

A common element of both of the above is the definition of local optimality criteria, based upon rerouted flows, that are in some cases stronger than the marginal definition of local optimality. This is true for the situation in [7] and the uncapacitated single commodity example in [6]. There are difficulties in accomplishing the same thing for our model. A major reason is that we do not have a one-to-one correspondence between extreme flows and local optima. This is due to both the presence of more than one commodity, and the generality of the arc cost functions.

We can analyze the effects of this problem, and examine the validity of testing adjacent extreme flows, by dividing the commodities into three categories. These categories correspond to the functional form of the incremental cost functions over the feasible range of a commodity's flow, on all feasible origin destination paths. The categories are: strictly concave, strictly convex, and mixed convex and concave. The mixed situation will occur if a single functional form does not apply over the entire feasible range of an arc cost function, the arc cost functions along a particular path have mixed functional forms, or the paths differ between each other in functional form.

Strictly Concave

In the strictly concave situation, there is ample justification for having a search algorithm examine adjacent extreme flows for that commodity. The reasoning for this is straightforward. Suppose that we hold the flows of all commodities but one fixed, and that given this fixed flow, the incremental cost functions for the free commodity are concave over the feasible region on all of its possible flow paths. As in ordinary concave cost networks, the all-or-nothing necessary condition will apply to the free commodity, and the optimal flow will be along a single path. To find the overall optimum flow for that commodity (given the fixed flow), we can evaluate F(X) for each alternative flow path, and choose the path corresponding to the smallest F(X).

Strictly Convex

In the case of commodities with strictly convex cost functions, we can do no better than a solution that is optimal in a marginal sense. This can be seen by again holding all commodities but one at some fixed level. Now suppose that, given the fixed flow, the incremental cost functions for the free commodity are convex over the feasible region on all possible flow paths for that commodity. The overall optimum solution for the flow of the free commodity (given the fixed flow) can be found by a gradient search technique such as Algorithm I. Thus, we could do no better by shifting flow of the free commodity from its originally assigned path(s).

Mixed Concave and Convex

Unfortunately, no simple argument exists for commodities whose incremental cost functions are a mixture of concave and convex, as the optimal solution for that commodity may have split flows and/or involve numerous local optima. In addition, the local optima do not necessarily correspond to any extreme flow of the commodities in question. The implication of these facts is that, in the mixed situation, we no longer have a finite set of local flow modifications to test for cost reductions. Therefore, adjacent flow concepts have much less relevance in the mixed situation.

In summary, we see that there is justification for searching adjacent extreme flows of commodities with strictly concave incremental cost functions. On the other hand, there is no reason to search adjacent flows of commodities with strictly convex incremental cost functions. In the case where a commodity has mixed concave and convex incremental cost functions, the situation is ambiguous; extreme flows may or may not be optimal. Adjacent flow modifications of these commodities could only be performed on a heuristic basis, without any underlying theory to guide us. For this reason, we shall employ the restriction that any algorithm that searches adjacent flows will only examine the adjacent extreme flows of commodities in the "strictly concave" category. These types of flows will be given the name of "adjacent concave flows".

4.3 A Stronger Definition of a Local Optimum

Condition (4.1) was the basis for an earlier definition of local optimality. If we wish to use adjacent concave flows as the basis for a stronger definition of local optimality, we must still take account of the fact that shifting any type of flow may cause us to move to a solution that is not optimal in the earlier defined marginal sense. One way to deal with this is to use each adjacent concave flow as the initial solution to a gradient based local optimization routine, such as Algorithm I. The solution then produced would have cost less than or equal to the adjacent concave flow, and would also meet condition (4.1). The general framework for an adjacent flow testing algorithm (which will be denoted Type A) would be:

Type A

1. Use Algorithm I to find an initial solution that meets condition (4.1). Define the set of adjacent concave flows corresponding to this solution.
2. Select an untested adjacent concave flow. If none exists, then STOP, the current solution is considered to be locally optimal.
3. a. Using the selected adjacent concave flow as a starting solution, perform Algorithm I.

   b. If the solution to Algorithm I has cost less than the current solution, go to Step 4; else, go to Step 2.
4. Replace the current solution with the new solution. Define the new set of adjacent concave flows corresponding to this solution and go to Step 2.

Of course, this algorithm lacks the elegant simplicity of merely evaluating the cost of all adjacent extreme flows; even worse, the repeated use of Algorithm I makes it extremely time consuming.

A less time consuming approach would be a similar algorithm that differs in that testing of adjacent concave flows for cost reductions occurs without performing Algorithm I. Only adjacent concave flows that are found to have cost lower than that of the current solution would be refined by Algorithm I to achieve optimality condition (4.1). The new local optimum (in the sense of condition (4.1)) would then become the starting point for again testing adjacent concave flows. This procedure would continue until we achieved a local optimum from which no further cost reductions could be made by reassigning a concave flow to an adjacent path. The general framework of this algorithm (which will be denoted Type B) would be:

Type B

1. Use Algorithm I to find an initial solution that meets condition (4.1). Define the set of adjacent concave flows corresponding to this solution.
2. Select an untested adjacent concave flow. If none exist, then STOP, the current solution is considered to be locally optimal.
3. If the solution corresponding to the selected adjacent concave flow has cost less than that of the current solution, go to Step 4; else, go to Step 2.
4. Perform Algorithm I upon the adjacent concave flow. Replace the current solution with the solution to Algorithm I. Define the new set of adjacent concave flows corresponding to this solution, and go to Step 2.

The final solution of this algorithm would then be optimal in a marginal sense, and also in a stronger adjacent flow sense. In addition, such an algorithm would be much faster than the Type A algorithm, which optimizes each adjacent concave flow. However, the solutions obtained by the Type B algorithm will often be worse than those obtained by the Type A Algorithm. In any case, they will be optimal in a weaker sense than those produced by the first type of algorithm; that is, it might be possible to achieve cost reductions by applying the Type A algorithm to a "final solution" produced by the Type B algorithm. This is due to the fact that each iteration of Algorithm I provides a solution with cost less than or equal to that of the previous iteration.

A better approach would be to somehow combine the quality of solutions of the Type A algorithm with the relative speed of the Type B algorithm. This is indeed possible by using the maximum number of iterations of Algorithm I as a control parameter. The rationale for such an approach is based upon the empirical observation that most of the cost reduction achieved by Algorithm I occurs in its first few iterations. This alternative approach would test the cost reduction achieved by shifting a concave flow to an adjacent path, and then performing at most n iterations of Algorithm I. (Less iterations would be required if local optimality is achieved earlier.) If this solution is then found to have cost less than that of the original flow, it is then refined by running Algorithm I to its completion, and becomes the new current solution. In this way, our solutions still meet condition (4.1).

An additional desirable feature would be the ability to choose the adjacent concave flow with *greatest* cost reduction after n iterations, not just the first flow found with reduced cost. However, this procedure might greatly decrease the speed of the algorithm without necessarily finding better solutions. Therefore, a hybrid approach has been chosen, whereby only the adjacent concave flows obtainable by shifting the flow of a single commodity being shipped between a given origin-destination pair are compared for the greatest cost reduction. A more formal statement of such an algorithm follows:

Algorithm II

Step 0 Select any X_0 belonging to Ω. Set i=0 . Select values for α and β such that $0 < \alpha,\beta < 1$. Set n = the maximum number of iterations of Algorithm I for testing adjacent concave flows.

Step 1 Using X_i as a starting solution, find a local optimum $\bar{X}_i$, using Algorithm I. Set {C} = the set of commodities that, given $\bar{X}_i$, have concave incremental path cost functions over the feasible range on their possible flow paths. Set $\bar{F}_i$ = the total cost of solution $\bar{X}_i$.

Step 2 If $\{C\} = \phi$, then STOP, $\bar{X}_i$ is the final solution. Else do: Select commodity k from the set {C}. For each of k's alternative flow paths;

a. Reroute the entire flow of commodity k to the alternate flow path.

b. Using the network with rerouted flow as a starting solution, perform n iterations of Algorithm I, or until condition (4.1) is satisfied, whichever occurs first.

Step 3 Set $\hat{X}_i$ = the solution in Step 2b with the lowest cost, $\hat{F}_i$. If $\hat{F}_i \geq \bar{F}_i$, set $\{C\} = \{C\} \setminus k$ and go to Step 2; Else, go to Step 4.

Step 4 Set $X_{i+1} = \hat{X}_i$, $i = i + 1$; go to Step 1.

Definition: A solution to Algorithm II using parameters n, α, and β will be denoted as (level n, α, β) locally optimal.

The solutions to Algorithm II have several interesting features that merit discussion. The most important feature is that any solution will have a form of local optimality that is stronger than the local optimality condition (4.1); this is ensured by Step 1. Also note the control parameter n. The algorithm has been structured such that any solution that is (level n, α, β) locally optimal will also be (level n^0, α, β) locally optimal for any $n^0 < n$. However, this does not imply that a higher value for n will necessarily ensure a lower cost final solution.

Recall the first two types of algorithms discussed in this section. The Type A algorithm involved achieving a local optimum of the form (4.1) when testing each adjacent flow. This fits into the framework of Algorithm II by setting $n = \infty$. The Type B algorithm involved testing adjacent flows without optimizing. This is equivalent to Algorithm II with n = 0. We see that both of these algorithms are actually special cases of Algorithm II; however, setting n to some small integer value will likely be more efficient.

It is also interesting to note that, when only one commodity is considered, the n=0 case of Algorithm II reduces to a concave programming algorithm, roughly similar to that developed by Hall [7]. This is because, in the single commodity case, the arc cost functions are all concave, which further implies that each extreme flow will also be a local optimum in the marginal sense. Algorithm II is then reduced to a two phase procedure: 1) obtain a marginally based local optimum of the form (4.1) by applying Algorithm I to the initial solution (which in the concave single-commodity case requires only a single iteration of Algorithm I), 2) search over all adjacent flows, moving to successively lower cost solutions, until no further cost reductions are possible.

The operation of Algorithm II is facilitated in solving Problem P' by results from the analysis of Chapter 3. In particular, the ratio tests, (*) and (**), of that chapter, provide simple means to evaluate whether an arc cost function will be concave or convex with respect to the incremental flow of a commodity. However, this is useful for only partially determining which incremental path cost functions are concave in Step 1. There will be some cases where the path cost function (defined by the summed arc cost functions along a particular origin-destination path) will be concave, even though one of the arcs on the path has a convex arc cost function. The arc oriented tests, (*) and (**), are not able to test for this possibility; thus their use would imply that some adjacent concave flows would be missed in Step 1 of the algorithm. The implementation of Algorithm II that provides the results of Chapter 5 uses the aforementioned ratio tests under the assumption that the run time computational requirements of a path oriented concavity test would be unjustifiably high.

4.4 Summary

The mathematical complexity of the arc cost functions causes previously developed techniques for solving non-linear min-cost flow problems to be inappropriate for the problem defined in this thesis. This is especially true if one considers finding a local optimum, as defined in the marginal sense (4.1), to be insufficient. By using a decomposition to transform Problem P, as defined in Chapter 3, to an equivalent form, P', we first simplified the constraint space such that finding a local optimum in the marginal sense was reduced to solving a series of shortest path problems. This was accomplished by devising a variation upon Klessig's algorithm [10] called Algorithm I. Then, to refine such solutions we proposed a search over all "adjacent concave flows" for possible cost reductions. These adjacent concave flows correspond to rerouted flow of those origin-destination-commodity triplets for which all path cost functions are concave.

The search procedure for cost reducing flows was made more complicated by reouiring that any new solution must also be locally optimal in the earlier defined marginal sense, implying a stronger form of local optimality, which we denoted (level n,α,β) local optimality. A procedure known as Algorithm II was defined that produces these (level n,α,β) locally optimal soloutions. Making the search process for these stronger local optima algorithmically operational required tradeoffs between solution quality and computational requirements. This was accomplished in Algorithm II by the "level n" parameter above, which controls the amount of computational effort expended in testing adjacent concave flows for cost reductions. Without this control parameter, it is likely that the algorithm would require inordinate amounts of time to achieve a final solution meeting the necessary local optimality requirements. Computational testing of Algorithm II is necessary to determine the effects of varying the level parameter, n, and to judge the algorithm's overall performance.

5. COMPUTATIONAL RESULTS

This chapter will provide a detailed examination and discussion of three major aspects of the performance of Algorithm II as applied to Problem P'. In the first section, we will examine the quality of the solutions obtained by the algorithm in relation to solutions obtained through alternate formulations and solution techniques. The following section will discuss the relationships between the initial and final solutions of Algorithm II. The data for these first two sections will be provided by a set of test problems involving relatively small networks, to be described more fully below. The third section of this chapter will evaluate the "level-n optimality" concept introduced in the preceding chapter. In this case, Algorithm II solves a series of test problems with n set at various levels. For reasons to be explained later in this chapter, these test problems will be performed on considerably larger networks than those of the first two sections of this chapter. The chapter will conclude with a final summary of the three sections and observations regarding the overall performance of Algorithm II.

5.1 Quality of Solutions

It is important that there be some means of verifying that the solutions produced by Algorithm II are "good" relative to solutions that can be obtained through alternative techniques or formulations. The ideal situation would be if the Algorithm II solution for a given problem could be compared to the global optimum for that problem; however, the complexity of the model prevents one from obtaining (or verifying) the global optimum in all but the simplest of cases. For this reason, our standards for comparison in judging the quality of the solutions will be somewhat less direct, and will be provided by solving three different forms of relaxations of the problem. These relaxations will involve eliminating different features of the model formulations relating to problems P or P', which in two cases allows us to use currently available alternative solution techniques: linear programming and concave programming. In this way, we can view the results of this analysis from the standpoint of comparison to alternative solution techniques, as well as from that of the appropriateness of modeling various commodity attributes. _The emphasis of this section will be upon the objective function values of the final solutions to Algorithm II._ Other features, such as run time performance, will be evaluated in later sections.

The _first relaxation_ involves eliminating the inventory holding costs related to vehicle frequency from the objective function of Problem P. The result is a linear objective function in a model with linear constraints, thus giving us a problem that is solved to optimality via standard linear programming techniques. This formulation was discussed in the literature review of Chapter 2 as an example of current modeling techniques in multicommodity network flow problems. Note that the final solution to this LP provides a lower bound to the optimal solution of Problem P'. As we shall observe, for certain problems, the solution to this relaxation provides a reasonably close approximation to the solution of the full problem. We can then obtain the true cost of the optimal solution to the linear programming relaxation by evaluating it under the objective function of the original problem. This cost is then an upper bound to the optimal solution of Problem P'.

The second relaxation involves reducing the set of commodities to a single "average commodity"[1] to be shipped through the network. This provides us with a single commodity network flow problem with all concave arc cost functions. Previously developed concave programming techniques could be applied to solve this problem; however, we will not have to resort to a separate algorithm. As noted in Chapter 4, Local Optima, in the single commodity case, Algorithm II reduces to a concave optimization algorithm similar to that devised by Hall [7]. The final solution to the relaxed problem will then be evaluated under the objective function of the original problem and the full commodity set, thus providing us with an additional upper bound to Problem P'. As with the first relaxation, this formulation also represents one of the means by which multi-attribute network flow problems are currently modeled.

A third relaxation comes about from assuming that all commodities have identical "ideal" densities, thus eliminating the need for one of the forms of physical capacity constraints, (3.5) or (3.6). Unfortunately, this modification does not alter the characteristic of arc cost functions having both concave and convex portions, making the degree of difficulty close to that of the original problem. Therefore, solving this problem also requires the full machinery of Algorithm II. The solution to this relaxation is relevant in that it forms an additional upper bound to the optimal solution of the original problem. Of further importance is that it provides a measure of the costs had we

---

1. In accordance with the variable definitions in Problem P (Chapter Three), a commodity refers to a distinct item type in the sense of having a unique weight, volume and inventory holding cost; however, multiple origin-destination pairs may be involved.

ignored one of the two capacity attributes: weight or volume. Doing so would have eliminated many technical details in our mathematical analysis of the problem and streamlined the operation of Algorithm II.

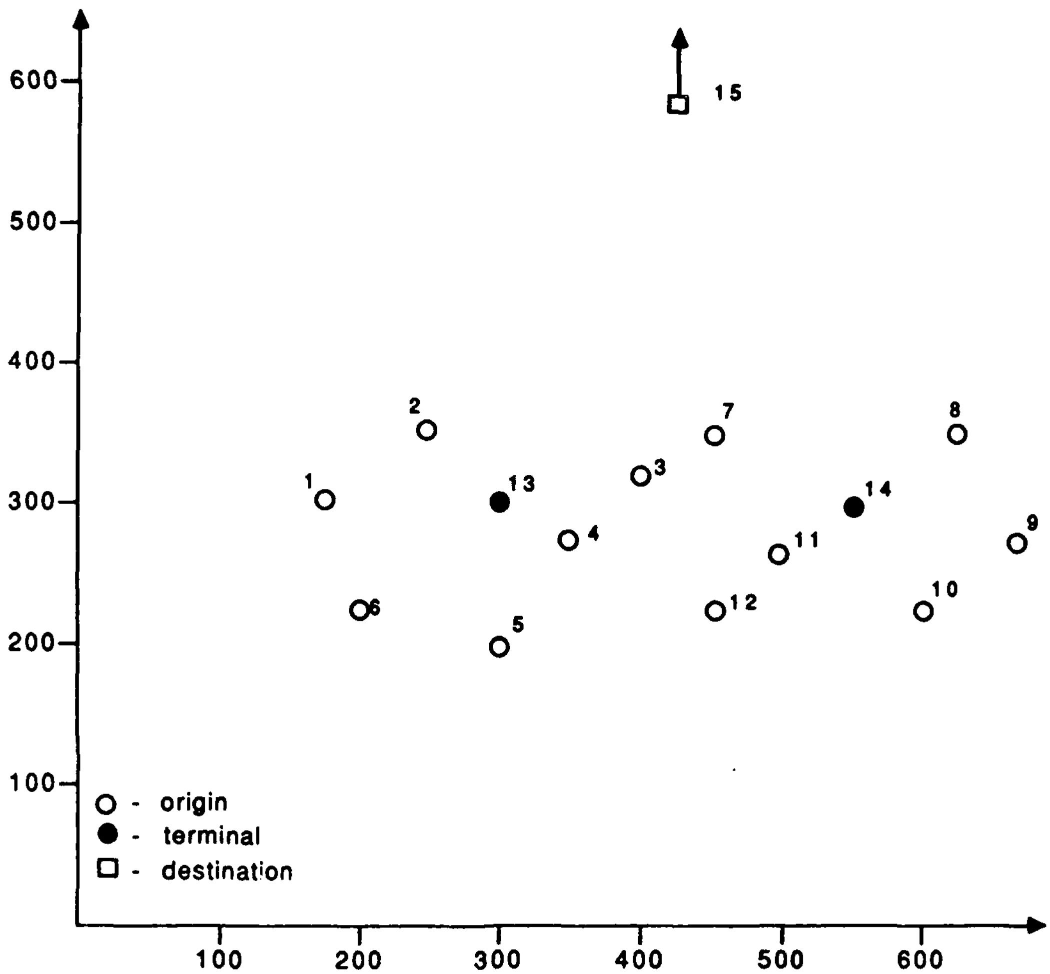
Description of Test Problems

A series of test problems was constructed to evaluate solution quality under varying circumstances of demand and travel distance. As noted earlier in this thesis, the set of network nodes consists of origins, destinations, and one level of transshipment terminals. In particular, there are 12 origins, two transshipment terminals, and one destination, located on an L2 metric grid. This particular size of network was chosen in an effort to balance two requirements: the need for sufficient problem complexity, and the desire to reduce the effort required to set up the problem constraints and transfer the solutions obtained in the linear programming relaxations. The initial configuration of the network is shown in Figure 5.1.

Four different network configurations are obtained by moving the destination node closer or further from the origins and transshipment terminals. By doing this, we change the relative desirability of routing commodities via a transshipment terminal, as opposed to routing them directly. One can achieve the same effect by moving the origins relatively closer or further from the transshipment terminals; however, this would have greatly increased the number of offline recalculations required to set up new network configurations.

Figure 5.1

Network for Solution Quality Testing



Each network configuration is tested under a total of four randomly generated demand sets, where each set is created by using a different random number seed in the following procedure: 1) each origin of the test network is randomly assigned one of eight possible commodities (for which the attributes are given in Table 5.1) , and 2) the amount of commodity flow is selected as a random quantity between 1/4 and 1 full truckload per unit time, as determined by the dominating attribute of that commodity.

Table 5.1
Commodity Attributes

Commodity	Weight (lbs.)	Volume (ft^3)	Inv.Hldg. Cost ($/weight-load/ unit time)
1	500.0	50.0	800
2	10.0	5.0	400
3	50.0	4.0	16
4	100.0	25.0	800
5	20.0	.5	8000
6	150.0	5.0	533
7	25.0	1.0	32
8	50.0	1.0	160

The final solution produced by Algorithm II, including the concave version used in the second relaxation, may be varied by using different initial solutions. Therefore, an attempt was made in this section to find a "good" initial solution for each problem instance by initializing Algorithm II with three different types of solutions. These were: 1) The optimal solution to the linear programming relaxation, 2) evenly dividing each commodity's flow over all possible paths, and 3) routing the flows through the transshipment terminal closest to each origin. In this section of the chapter, we will provide Algorithm II results for only the best solution (one with least cost) obtained via the three alternative starting solutions. The next section of this chapter will discuss in detail the effects of using different types of initial solutions under varying circumstances.

Table 5.2 provides a summary of the values of parameters used in the test problems.

Table 5.2
Parameter Values for Test Problems

Parameter	Value
Vehicle Weight Capacity	80,000 lbs.
Vehicle Volume Capacity	4,200 $ft.^3$
Vehicle Travel Time	.002 days x # of miles traveled
Vehicle Cost	$1.35/mile x # of miles traveled
Tuning Parameter (α)	.5
Tuning Parameter (β)	.5

Test Results

Algorithm II was coded in FORTRAN IV and implemented on an IBM 3081. The linear programming relaxations were solved via the simplex method using the LINDO package. Tables 5.3 a) - 5.3 d) summarize the results of testing the alternate formulations and solution techniques, where the objective function value listed is that obtained by evaluating the respective final solutions under the objective function of the original problem formulation. The cost of the solution with all flows routed directly is also provided, as an additional benchmark in our analysis. The relative rank of each formulation, in relation to the others in that problem instance, is given in parenthesis.

Table 5.3

Objective Function Values and Rankings of Alternative Model Types
($/unit time)

5.3 a) Demand Set 1

Solution Type	Coordinate of Destination (miles)							
	587		875		1162		1450	
Algorithm II (Problem P')	19333	(2)	31328	(1)	43133	(1)	54988	(1)
Linear Program	20696	(5)	33773	(5)	44817	(5)	55844	(4)
Concave Program	19149	(1)	31332	(2)	43133	(1)	54988	(1)
Algorithm II (One Capacity)	19463	(4)	31529	(4)	43636	(3)	55808	(3)
Direct Routing	19333	(2)	31462	(3)	43656	(4)	55914	(5)

5.3 b) Demand Set 2

Solution Type	Coordinate of Destination (miles)							
	587		875		1162		1450	
Algorithm II (Problem P')	4340	(1)	7189	(1)	9913	(1)	12671	(1)
Linear Program	4567	(5)	7265	(2)	9960	(2)	12717	(2)
Concave Program	4408	(2)	7641	(4)	10889	(4)	14187	(4)
Algorithm II (One Capacity)	4408	(2)	7800	(5)	11304	(5)	14552	(5)
Direct Routing	4408	(2)	7539	(3)	10789	(3)	14088	(3)

5.3 c) Demand Set 3

Solution Type	Coordinate of Destination (miles)							
	587		875		1162		1450	
Algorithm II (Problem P')	6270	(1)	10181	(1)	14249	(1)	18166	(1)
Linear Program	6539	(4)	10569	(3)	14435	(3)	18349	(2)
Concave Program	6308	(3)	10574	(4)	15092	(5)	18900	(4)
Algorithm II (One Capacity)	6672	(5)	10678	(5)	14790	(4)	18959	(5)
Direct Routing	6297	(2)	10287	(2)	14415	(2)	18605	(3)

5.3 d) Demand Set 4

Solution Type	Coordinate of Destination (miles)							
	587		875		1162		1450	
Algorithm II (Problem P')	10297	(1)	16485	(1)	22366	(1)	28257	(1)
Linear Program	11209	(5)	16968	(3)	22851	(3)	28916	(3)
Concave Program	10882	(4)	17347	(5)	23762	(5)	30147	(4)
Algorithm II (One Capacity)	10327	(2)	16562	(2)	22366	(1)	28316	(2)
Direct Routing	10327	(2)	16974	(4)	23593	(4)	30185	(5)

The most noticeable feature of the results of Table 5.3 is that Algorithm II, when used to solve the formulation P', in almost every case produces a lower cost final solution than any other technique or formulation. This is not too surprising, since only formulation P' takes full account of all three commodity attributes: weight, volume, and inventory holding cost. However, in many cases, the differences in cost are significant.

In the case of the optimal solution to the linear relaxation, there are consistently large differences between it and the Algorithm II solution to Problem P'. However, the size of this difference decreases as the destination becomes further from the origins and terminals; at the shortest distance, the average difference is 6.3%, while at the longest distance, the average difference is 1.3%. This is due to the fact that as distance increases, the C_{ij} terms increase, which in turn decreases the second derivatives of the arc cost functions. In effect, the arc cost functions become more linear, and the linear programming relaxation becomes closer to the costs of the original formulation.

The magnitude of the distance effect will vary greatly from one set of demands to another, depending upon the amount of linearity induced by the feasible combinations of commodity flows. For example, we would expect that demand sets containing large quantities of commodities with high inventory holding costs would be more difficult to approximate with a linear programming relaxation. This can be illustrated by observing Table 5.4 . The attributes of all commodity flows have been summed within each demand set to obtain the total amounts of weight, volume, and inventory holding cost. The last columns of the table provide the average percentage difference in objective function values occurring between the Algorithm II solution and the linear and concave solutions. Although this provides only a rough comparison, we can see that the rankings of these average reductions for the optimal linear solution fall in the same order as the size of the total inventory holding cost.

Table 5.4
Total Attribute Levels vs. % Reduction

Demand Set	Weight Loads	Volume Loads	Inv. Hldg. Cost/ Unit Time	Avg. % Reduction-Alg.II vs.- Opt. Linear	Concave
1	6.06	5.93	15,221.88	5.1	- 0.2
4	4.60	6.56	5,892.60	4.1	6.0
3	4.36	7.76	1,635.78	2.6	3.6
2	4.44	6.21	634.56	1.5	7.1

Inventory holding cost considerations may in some cases prove more important than accounting for differences in commodity densities. Observe in Table 5.3 that for the two demand sets with highest total inventory holding cost/unit time shown above, sets 1 and 4, the Algorithm II solution to the single capacity formulation also produces lower objective function values than the optimal linear solution. This occurs even though set 1 is "heavy" and set 4 is "light", while the single capacity formulation had assumed uniformly ideal densities.

In general, the concave programming algorithm did not perform particularly well when compared to Algorithm II. However, we can observe from Table 5.4, that in the case of Demand Set 1, the concave programming algorithm performed slightly better on average. Here, the unusually large inventory holding costs caused the arc cost functions to be highly concave, thus allowing a reasonably close functional approximation. This is particularly true when the travel distances are relatively short, or, equivalently, the C_{ij} terms are relatively small. This distance effect also causes the arc cost functions to become more concave. In general, the percentage difference in objective function values between Algorithm II and the concave programming algorithm increases as the destination is moved further away from the origins and terminals -- from an average difference of 1.7% at 587 miles, to an average difference of 6.1% at 1450 miles.

5.2 Dependence Between Initial and Final Solutions

The previous section demonstrated that demands and network structure affect the linearity of the arc cost functions. This in turn affected the quality of solutions produced by initializing the algorithm with the optimal solution to a linear programming relaxation of the problem. Similarly, these factors will affect the ability of various types of initial solutions to lead to a final solution with least cost in Algorithm II. In this next phase of the computational study, the dependence between initial and final solutions of Algorithm II is investigated. Our source of data will be the same test problems used in the previous section. Further insight is gained by providing two different sets of commodity attributes for each problem instance.

The first set of commodities is the one used previously, and contains large variations in density between commodities. The second set of commodities has a much narrower range of density. It was constructed by generating densities randomly in an interval of $\pm$ 10 lbs./ft.^{3} about the ideal density of 19.05 lbs./ft.^{3}. However, the ordering of commodities by density is maintained by assigning the random densities to the appropriate commodity.

The results of this investigation have been summarized in Table 5.5 . The type of initial solution which led to the least cost final solution of Algorithm II is given for each problem instance using the following code:

L - the optimal solution to the linear relaxation,
E - evenly dividing the flow between an origin-destination pair over all possible paths,
C - routing flows through the terminal closest to an origin, and
D - routing all flows directly.

The second place initial solution(s) are enclosed in parenthesis immediately after the first, followed by the percentage difference between first and second place objective functions.

Table 5.5
Initial Solution Providing Lowest Cost Final Solution

Demand Set	Variation in Density	Y Coordinate of Destination (miles) 587	875	1162	1450
1	High	D,(C)-.55%	L,E,C-0%	L,E,C-0%	L,E,C-0%
1	Low	L,(E,C)-.01%	L,(E,C)-.04%	L,(E)-.03%	L,E,C-0%
2	High	E,(D)-.19%	L,(E)-1.47%	L,(E)-.42%	L,(E)-.7%
2	Low	L,E,C-0%	L,(D)-1.12%	L,(C)-.93%	L,(C)-1.15%
3	High	L,(D)-.42%	L,(D)-1.0%	E,(L)-.56%	L,(E)-.5%
3	Low	D,(L)-.02%	L,E-0%	L,E-0%	L,E-0%
4	High	L,(E)-.11%	L,(C)-.46%	C,(L)-.88%	C,(L)-.87%
4	Low	D,(L)-.25%	C,(L)-1.0%	C,(E)-.81%	C,(E)-1.33%

We see above that the optimal linear solution generally provides the best initial solution for both sets of commodities. A likely explanation for this is that the linear relaxation is solved to optimality, thus accounting for all possible savings due to commodity density considerations. On the other hand, the remaining initial solutions are approximate and heuristic in nature. That is, direct routing seeks to minimize item-miles traveled; the closest terminal strategy attempts to capitalize on the economies of scale effect of concave functions; and the even flow strategy, by initially sending flow on all paths, is able to sample the marginal costs on all paths at some non-zero flow level. However, one major drawback in using an LP optimal initial solution is the considerable effort required to set up the formulation and transfer the solution to Algorithm II in the proper format. In practice, this process would have to be automated via specially tailored preprocessing routines.

One of the most striking features of the above results is the consistently small difference between first and second place final solutions: usually less than 1%, and often, 0%. This result is encouraging in that it shows that the quality of the Algorithm II final solutions is nearly independent of the starting solutions. However, it would be desirable to test this result more rigorously by randomly generating many different initial solutions and analyzing the variations in final solutions. Unfortunately, time and computer resource constraints prevented the inclusion of such a test in this thesis. On the other hand, the extreme difference in form of the initial solutions that were examined makes this relatively limited test more credible.

There appears to be few differences between the type of results achieved with the two different sets of commodities. One might expect the optimal linear solution to perform relatively better as an initial solution when density variations are high, implying high potential consolidation savings. However, this is not the case here. It appears that Algorithm II is also able to capitalize on density variations to achieve consolidation savings.

In some of the above problem instances, routing all flows direct was a better solution than any that could be achieved via Algorithm II. This situation becomes more likely when the distance from the terminals to the destination is small relative to the distance from the origins to the destination. This is more a characteristic of the network than any failing of Algorithm II. However, it is worth noting that Algorithm II can not make any improvement to an initial solution with all flows routed direct. The reason for this is in the way that Algorithm II tests adjacent flows. Recall that the source of cost savings in our network model is the consolidation of two or more commodities at a transshipment terminal. Algorithm II operates by shifting only one commodity flow at a time, and then evaluating marginal costs to determine the extent of any further flow shifts. The triangle inequality ensures that marginal costs will always be lower on the direct route than on the terminal route for any single commodity flow.

5.3 Level n Optimality

In this section, we will test the performance of the part of Algorithm II that searches over adjacent concave flows, and observe how this performance varies with changes in the value of the control parameter n. Performance will be judged in terms of the reduction in costs below those of the initial solution and those of the initial local optimum. We will also examine the execution time of the algorithm for different levels of n, and contrast the time requirements with the cost reductions at each level.

This phase of the computational testing required a more complicated series of test problems. The test problems discussed previously were unsuitable, due to the fact that the final solutions to Algorithm II were achieved at level n=0; that is, setting n to a higher level had no effect upon the final solution value. This situation will always occur when there is only one flow from each origin, and the local optimum does not have any split flows. These conditions imply that all flows will follow a single path from origin to destination, and the unused paths will have at least one arc with zero flow. Then, since the gradients on the zero flow arcs will have a value of infinity, the gradient based shortest path algorithm imbedded in Algorithm II will never select an unused path over the current path. Thus, flow changes will only occur as a result of shifting adjacent concave flows, not through the gradient based optimization of Algorithm I.

The test networks in this section were generated randomly, instead of having preassigned node locations as before. However, as we shall describe, this did not entail distributing the nodes completely independent of each other. In the procedure that was used, three terminal nodes and five destination nodes were first assigned to random locations on a 1000 mile by 1000 mile grid. For each terminal, a random number, uniformly distributed between five and 20, was generated to determine the number of origins to be assigned to the region within a 200 mile radius (within the limitations of the grid space) of the terminal. By locating each of the the origins close to at least one terminal we increase the likelihood that a reasonably interesting amount of consolidation will occur in the final solutions. However, this "assignment" of origin to terminals is not meant to imply any permanent restriction of flows to a particular terminal during the course of the algorithm.

Commodities were randomly assigned to the origins such that each origin shipped a commodity to each of the five destinations. The quantity of each commodity shipped between a given origin-destination pair was a random number uniformly distributed between 1 unit and .1 truckload. The attributes of the commodities are the same as those used earlier in this chapter, and can be found in Table 5.1.

Each problem instance was initiated by two types of initial solutions: one with flow evenly divided over all possible paths, and one with flow routed through the terminal closest to each origin. The optimal linear solution was not used here as an initial solution, due to the large time requirements to manually formulate and transfer such solutions for networks of this size. However, our analysis of the previous section suggests that this will not have a large effect upon the quality of our solutions. The costs of direct routing have been included in the table for comparative purposes.

Each network was solved using Algorithm II with n set at 0,1 and 2, with no further cost reductions found above this level. Budgetary constraints on computer time prevented running more than five distinct network/demand configurations for each level of n, and each initial solution. Table 5.6 summarizes the network characteristics while Table 5.7 summarizes the results of these tests. The parameters of the model are those used earlier, and are given in Table 5.2. The cpu times result from using an IBM 3081, and include I/O time.

Table 5.6
Summary of Network Characteristics for Level n Analysis

origins	Uniform (5,20) at each terminal
terminals	3
destinations	5
origin-dest. pairs	5 per origin (one for each destination)
commodities	1 per origin-destination pair, chosen randomly from the set of 8 commodities in Table 5.1

Table 5.7
Solution Values and CPU Times for Varying Levels of n
($/unit time)
(CPU seconds given in parenthesis)

No. of Orgns	Initial Solution	Starting Cost	Marginal Optimum	Level n=0	n=1	n=2
21 1	Even Flow	45903	23429	22961 (5.51)	22961 (15.31)	22961 (24.36)
21	Closest Terminal	23520	23520	22961 (3.82)	22961 (17.40)	22961 (27.53)
21	Direct Routing	22961				
36 2	Even Flow	67232	32068	32007 (5.58)	31872 (36.78)	31872 (57.14)
36	Closest Terminal	32482	32482	31978 (5.89)	31872 (35.91)	31872 (55.86)
36	Direct Routing	41489				
36 3	Even Flow	121465	62673	62403 (3.62)	62388 (18.36)	62349 (40.05)
36	Closest Terminal	64013	64013	62591 (4.33)	62412 (18.70)	62412 (30.72)
36	Direct Routing	76768				
45 4	Even Flow	90004	39583	39576 (8.62)	39492 (41.52)	39492 (67.22)
45	Closest Terminal	41239	41239	39492 (15.73)	39492 (81.04)	39492 (131.69)
45	Direct Routing	52583				
49 5	Even Flow	84152	40853	40748 (10.57)	40729 (84.05)	40729 (107.39)
49	Closest Terminal	40897	40897	40781 (10.17)	40778 (52.52)	40730 (112.68)
49	Direct Routing	57501				

We see that in each of the ten network/initial solution combinations, the final solution value appears to "bottom out" by the time that n reaches 2, if not sooner. In fact, the average percentage reduction in solution values from n=0 to n=2 is only slightly more than .1%. We also observe large increases in cpu requirements as n becomes higher than zero. These two factors suggest that it may not be worthwhile to set n higher than 2, and arguably, not higher than 0. On the other hand, the n=0 level produces solutions that are reduced an average of 1.4%, and as high as 4.2% in one case, from the initial marginally based local optimum. It would be premature to generalize from these observations; but it does suggest that available computer time might be better spent generating a number of initial solutions and solving them with n set to a low value, 0 say, than by solving a single problem with n set higher.

The majority of the cost reduction achieved by the algorithm occurs in the process of reaching an initial local optimum (Algorithm I). This would tend to indicate that, in this particular problem structure, a marginally based local optimum is often a relatively "good" solution. On the other hand, it is interesting to note that there appears to be some tradeoffs involved in solving a problem via Algorithm II versus Algorithm I alone. For example, by initializing Algorithm I with different starting solutions, we can obtain a number of final solutions (marginally based local optima), with a wide range of costs. Additional cost reduction can then be achieved by proceeding with the remainder of Algorithm II. This additional cost reduction tends to be relatively small when the Algorithm I solution is "good", and relatively large when the Algorithm I solution is "poor". In effect, the adjacent concave flow search allows the poorer Algorithm I solutions to "catch-up" to better Algorithm I solutions. Thus, it is necessary to go beyond the marginally based local optima of Algorithm I to *ensure* a "good" final solution.

The solutions produced at the higher levels are less dependent upon the initial solution than those at the lower levels. In three of the five problems, the final results are identical over each initial solution type, while the other two problems have only insignificant differences. This is particularly interesting in light of the major differences in form and cost of the starting solutions. This would suggest a certain amount of robustness on the part of Algorithm II.

5.4 Summary

In the first section of this chapter, we saw how solving Problem P', via Algorithm II, compared to solving relaxed versions of Problem P' via linear programming, concave programming, or Algorithm II itself. It was shown in a series of test problems that Algorithm II, operating on the full version of Problem P', was consistently superior to other solution techniques or alternate formulations. This is largely due to the fact that only Algorithm II has the methodology to deal with all three of the commodity attributes: weight, volume, and inventory holding cost.

The second section demonstrated that Algorithm II is not overly dependent upon its initial solution. This was true over a wide variety of demands, network structures, and commodity attributes. Such a characteristic indicates a certain amount of robustness on the part of Algorithm II, and provides some initial evidence that near optimality is being achieved.

The final section explored the behavior of Algorithm II as the level-n parameter was varied. We saw that n need not be set higher than 2 for the algorithm to achieve its maximal cost reductions. In fact, the relatively small amount of cost reduction achieved by increasing n from 0 to 2 may not justify the relatively large additional cpu time requirements.

6. CONCLUSIONS

This thesis has demonstrated the advantages of using the multiattribute multicommodity network flow formulation to model load planning problems in transportation networks. Previously, research has focused upon techniques for solving problems with less complex formulations, neglecting one or more facets of the general problem. These shortcomings can lead to improper cost estimations, and thereby, to suboptimal routings of vehicles and commodities in the transportation network.

Solving the multiattribute multicommodity formulation in reasonably large networks required developing new methodology to deal with arc cost functions that were shown to have both concave and convex portions. This nonlinear feature results in the presence of numerous local optima, over which an exhaustive search is computationally impractical. To overcome this difficulty, it was first necessary to analyze the structure of the cost functions and the constraint space. This analysis provided several new and powerful results, which were then incorporated into the solution algorithm presented in this thesis.

The three-dimensional parameter space concept developed in Chapter 3 is an intuitively appealing means of representing multiattribute flow across an arc. Incrementing commodity flows can be represented by movement in a straight line through the space. Inflection points of the commodity's incremental arc cost function can then be found at the intersections of this line with the boundaries between the flow regions R^w, R^v, and R^d.

In Chapter 3, a simple ratio test was derived which indicates whether an arc cost function will be concave or convex with respect to an incremental flow. This result can be used in the development of efficient solution algorithms by eliminating the need for numerical analysis techniques to determine the second derivatives of the arc cost functions. Recall that the ratios can also be represented in the parameter space by the slopes of the lines defined by average and incremental flow.

The computational results in Chapter 5 demonstrated the importance of accounting for all three of the commodity attributes: weight, volume, and inventory holding cost. Algorithm II is currently unique in its ability to effectively operate in a multicommodity environment that includes these three attributes, and appears most applicable to situations with multiple commodities and high inventory costs. In contrast, linear models are incapable of handling the effects of high inventory costs, while previously developed nonlinear models neglect the possibility of capitalizing upon the commodity density variations often present in multicommodity situations. On the other hand, the single commodity case may be adequately handled through already established concave programming techniques (to which Algorithm II reduces in the single commodity situation); furthermore, in situations without high inventory holding costs, a linear model may be more appropriate. In fact, preliminary trials of Algorithm II in instances where all commodities have low inventory holding costs indicate poor performance; run times are very long, and the algorithm has trouble converging at all.

Some of the generalizations above regarding model appropriateness have been summarized in Table 6.1 below. These generalizations may be applied to min-cost network flow problems whose network structure is like that described in this thesis.

Table 6.1
Generalizations of Appropriate Modeling Techniques

	High Density Variation		Low Density Variation	
	Inventory Costs		Inventory Costs	
	High	Low	High	Low
Single Commodity[1]	-	-	C	D
Multicommodity	AII	LP	AII	D

AII - Algorithm II
C - Concave Programming
D - Direct Routing (little motivation for consolidation)
LP - Linear Programming

Algorithm II demonstrated the capability to efficiently produce "good" local optima, with costs significantly lower than known alternatives. The final solutions generated are largely independent of the form of the initial solutions; that is, nearly the same final solution is attained, regardless of the starting solution. This partially suggests that the final solutions may be nearly optimal, or at the very least, that Algorithm II is relatively robust. The level n control parameter proved to be an effective means of holding down computational requirements while still allowing significant cost reductions.

Further testing of Algorithm II upon larger and more complex network structures is desirable. It would be particularly interesting

1. Recall that in this thesis the term, "single commodity", implies the existence of only one commodity in the sense of having a unique weight, volume, and inventory holding cost; however, multiple origin-destination pairs may be involved.

to test Algorithm II in problems that allow commodities more than one stop on paths between their origin and their destination. This would be the case in a problem which modeled consolidation as well as break-bulk activities at the terminals. The ability of Algorithm II to efficiently find low cost solutions in this multi-level terminal situation is still an open question.

REFERENCES

[1] Blumenfeld, Burns, Diltz, and Daganzo, "Analyzing Tradeoffs Between Transportation, Inventory, and Production Costs on a Freight Network", *Trans. Res.* , 19B(5), 1985, pp. 361-380

[2] Burns, Hall, Blumenfeld, and Daganzo, "Distribution Strategies That Minimize Transportation and Inventory Costs", *Operations Research*, 33(3), 1985, pp. 469-490

[3] Daganzo, C.F., "Shipment Composition Enhancement at a Consolidation Center", Unpublished Paper, Institute of Trans. Studies, U.C. Berkeley, Berkeley ,CA, 1987

[4] Erickson, Monma, Veinott, "Send-and-Split Method for Minimum-Concave-Cost Network Flows", Tech. Report 33, Department of Operations Research, Stanford University

[5] Franke, M. and Wolfe, P.,"An Algorithm for Quadratic Programming", *Nav. Res. Log. Quart.*, 3, 1956, pp. 95-110

[6] Gallo, G. and Sodini, C., "Adjacent Extreme Flows and Application To Min-Concave Cost Flow Problems", *Networks*, 9, 1979, pp. 95-121

[7] Hall, R.W., "Route Choice on Freight Networks with Concave Costs and Exclusive Arcs", Unpublished Paper, Dept. of IEOR and the Institute of Trans. Studies, U.C. Berkeley, Berkeley, CA 1987

[8] Hall, R.W. and Daganzo, C.F., "Vehicle Miles for a Freight Carrier With Two Capacity Constraints", *Trans. Res. Record*, 1038,pp. 39-40

[9] Kennington, J.L.,"A Survey of Linear Multicommodity Network Flows", *Operations Research*, 26(2), 1978, pp. 209-236

[10] Klessig, R.W., "An Algorithm for Nonlinear Multicommodity Flow Problems", *Networks*, 4, 1974, pp. 343-355

[11] LeBlanc, L.J., "An Algorithm for the Discrete Network Design Problem", *Trans. Sci.*, 9, 1975, pp.183-199

[12] Magnanti, T.L. and Wong, R.T., "Network Design and Transportation Planning: Models and Algorithms", *Trans. Sci.*, 18, 1984, pp. 1-55

[13] Steenbrink, P.A. ,"Transport Network Optimization in the Dutch Integral Transportation Study", *Trans. Res.*, 8, 1974, pp. 11-27

[14] Swoveland C. ,"A Two-Stage Decomposition Algorithm for a Generalized Multicommodity Flow Problem", *INFOR*, 11, 1973, pp. 232-244

[15] Tomlin, J. A.,"Mathematical Programming Models for Traffic Network Problems", unpublished dissertation, Department of Mathematics, University of Adelaide, Australia, 1967

[16] Weigel, H.S. and Cremeans, J.E., "The Multicommodity Network Flow Model Revised to Include Vehicle Per Time Period and Node Constraints", *Naval Res. Log. Qrtly.*, 19, 1972, pp. 77-89

[17] Yaged, B., Jr., "Minimum Cost Routing for Static Network Models", *Networks*, 1, 1971, pp. 139-172

[18] Zangwill, W.I., "Minimum Concave Cost Flows in Certain Networks", *Management Science*, 14, 7, 1968, pp. 429-450

Appendix 1A

Derivation of Expressions for Bounds on Nonlinearity,

Case 1 : $1x_{ij}^{\backslash k} \geq (hx_{ij}^{\backslash k}/2C_{ij})^{\frac{1}{2}},\ sx_{ij}$

Definition: $B^{W} = 1 - (F_{ij}^{W})'_{k}/(F_{ij}^{W})'_{k}\big|_{\infty}$

By inserting the expressions 3.14 and 3.15 for first derivative and limiting derivative into the above expression we obtain:

$$B^{W} = 1 - [\ C_{ij} + (h^{k}1x_{ij}^{\backslash k} - hx_{ij}^{\backslash k})/2(1x_{ij}^{\backslash k})^{2} + t_{ij}h^{k}\ ] / [\ C_{ij} + t_{ij}h^{k}\ ]$$

Because $x_{ij}^{k} = 0$, the term $1x_{ij}$ can be set to $1x_{ij}^{\backslash k}$, and if we eliminate the term $t_{ij}h^{k}$ by assuming $t_{ij}h^{k} << C_{ij}$, we can say

$$B^{W} < 1 - [\ C_{ij} + (h^{k}1x_{ij}^{\backslash k} - hx_{ij}^{\backslash k})/2(1x_{ij}^{\backslash k})^{2}\ ] / C_{ij}.$$

Continuing with the simplification:

$$B^{W} < [hx_{ij}^{\backslash k} - h^{k}1x_{ij}^{\backslash k}] / [2C_{ij}(1x_{ij}^{\backslash k})^{2}] .$$

Suppose now that B^{W} must be less than some α , which we can accomplish by setting

$$[hx_{ij}^{\backslash k} - h^{k}1x_{ij}^{\backslash k}] / [2C_{ij}(1x_{ij}^{\backslash k})^{2}] \leq \alpha$$

which is equivalent to

$$(hx_{ij}^{\backslash k} - h^{k}1x_{ij}^{\backslash k}) \leq 2\alpha C_{ij}(1x_{ij}^{\backslash k})^{2} .$$

By dividing through by $hx_{ij}^{\backslash k}$ we obtain:

$$(\ 1 - h^{k}/\bar{h}_{W}\ ) \leq 2\alpha C_{ij}(1x_{ij}^{\backslash k})^{2}/hx_{ij}^{\backslash k} .$$

Then by rearranging terms:

$$h^k/\bar{h}_w \geq (1 - \alpha(r_{ij}^{kw})^2)$$

where $$r_{ij}^{kw} = 1x_{ij}^{\backslash k}/(hx_{ij}^{\backslash k}/2C_{ij})^{\frac{1}{2}}$$

Appendix 1B

Derivation of Expressions for Bounds on Nonlinearity,

Case 2: $1x_{ij}^{\backslash k}$, $sx_{ij} \leq (hx_{ij}^{\backslash k}/2C_{ij})^{\frac{1}{2}}$

Definition : $B^{W} = 1 - (F_{ij}^{W})_{k}'/(F_{ij}^{W})_{k}'|_{\infty}$

By inserting the expressions 3.14 and 3.15 for first derivative and limiting derivative into the above expression we obtain:

$$B^{W} = 1 - [\ C_{ij} + (h^{k}1x_{ij}^{\backslash k} - hx_{ij}^{\backslash k})/2(1x_{ij})^{2} + t_{ij}h^{k}\ ]\ /\ [\ C_{ij} + t_{ij}h^{k}\ ]$$

In this case we set $1x_{ij}$ to $1x_{ij}^{\backslash k} + x_{ij}^{k}$ such that

$$1x_{ij} = 1x_{ij}^{\backslash k} + x_{ij}^{k} = [(hx_{ij}^{\backslash k} + h^{k}x_{ij}^{k})/2C_{ij}]^{\frac{1}{2}}.$$

Putting this equality into standard quadratic form we have

$$[x_{ij}^{k}]^{2} + [2(1x_{ij}^{\backslash k}) - h^{k}/2C_{ij}]x_{ij}^{k} + [(1x_{ij}^{\backslash k})^{2} - hx_{ij}^{\backslash k}/2C_{ij}] = 0$$

Thus $a = 1$

$$b = [2(1x_{ij}^{\backslash k}) - h^{k}/2C_{ij}]$$

$$c = [(1x_{ij}^{\backslash k})^{2} - hx_{ij}^{\backslash k}/2C_{ij}]$$

It is convenient to first solve for $b^{2} - 4ac$, which is equivalent to

$$(h^{k}/2C_{ij})^{2} - 2(1x_{ij}^{\backslash k})/C_{ij}(h^{k} - \bar{h}_{w})$$

We can then say that

$$x_{ij}^{k} = \tfrac{1}{2}\{[h^{k}/2C_{ij} - 2(1x_{ij}^{\backslash k})]\ (+/-)\ [(h^{k}/2C_{ij})^{2} - 2(1x_{ij}^{\backslash k})/C_{ij}(h^{k} - \bar{h}_{w})]^{\frac{1}{2}}\}$$

Since we have assumed that $h^k \leq \bar{h}_w$ the term in the square root is positive and x_{ij}^k will have real roots. Also, because we are finding the solution for an increasing x_{ij}^k, we need only find the upper root. Thus by substituting the expression above for x_{ij}^k into $(1x_{ij}^{\backslash k} + x_{ij}^k)^2$ and by cancelling terms we obtain

$$(1x_{ij}^{\backslash k} + x_{ij}^k)^2 = h^k/4C_{ij}(b^2 - 4ac)^{\frac{1}{2}} + \tfrac{1}{2}(h^k/2C_{ij})^2 - 1x_{ij}^{\backslash k}/2C_{ij}(h^k - \bar{h}_w)$$

But since $h^k < \bar{h}_w$, $(b^2 - 4ac)^{\frac{1}{2}} > h^k/2C_{ij}$, we can say

$$(1x_{ij}^{\backslash k} + x_{ij}^k)^2 > (h^k/2C_{ij})^2 - 1x_{ij}^{\backslash k}/[2C_{ij}(h^k - \bar{h}_w)]$$

Using the right hand side above to replace $(1x_{ij})^2$ in the expression for B^w we obtain:

$$B^w < 1 - \{C_{ij} + [1x_{ij}^{\backslash k}(h^k - \bar{h}_w)]/[2(h^k/2C_{ij})^2 - 1x_{ij}^{\backslash k}(h^k - \bar{h}_w)] + t_{ij}h^k\}/\{C_{ij} + t_{ij}h^k\}$$

And if we eliminate the $t_{ij}h^k$ term by assuming that $t_{ij}h^k \ll C_{ij}$

$$B^w < [1x_{ij}^{\backslash k}(\bar{h}_w - h^k)] \;/\; [(h^k)^2/2C_{ij} - 1x_{ij}^{\backslash k}(h^k - \bar{h}_w)]$$

Suppose now that B^w is required to be less than some α, which we can accomplish by setting

$$[1x_{ij}^{\backslash k}(\bar{h}_w - h^k)] \;/\; [(h^k)^2/2C_{ij} - 1x_{ij}^{\backslash k}(h^k - \bar{h}_w)] \leq \alpha$$

which after simplification is equivalent to

$$(\bar{h}_w - h^k) \leq [(\alpha/(1-\alpha))(h^k)^2] \;/\; [2C_{ij}1x_{ij}^{\backslash k}]$$

Dividing through by $\bar{h}_w$ we obtain

$$1 - h^k/\bar{h}_w \leq [(\alpha/(1-\alpha))(h^k)^2] \;/\; [2C_{ij}hx_{ij}^{\backslash k}]$$

or

$$h^k/\bar{h}_w \geq [\; 1 - (\alpha/(1-\alpha))(r_{ij}^{kd})^2 \;]$$

where $r_{ij}^{kd} = [hx_{ij}^{\backslash k}/2C_{ij}]^{\frac{1}{2}}/[hx_{ij}^{\backslash k}/h^k]$