**THE AVERAGE NETWORK FLOW
PROBLEM: SHORTEST PATH AND
MINIMUM COST FLOW FORMULATIONS,
ALGORITHMS, HEURISTICS, and
COMPLEXITY**

DISSERTATION

Jeremy D. Jordan, Captain, USAF

AFIT/DS/ENS/12-09

**DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY**

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

THE AVERAGE NETWORK FLOW PROBLEM: SHORTEST PATH AND
MINIMUM COST FLOW FORMULATIONS, ALGORITHMS, HEURISTICS,
AND COMPLEXITY

DISSERTATION

Presented to the Faculty

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

in Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

Jeremy D. Jordan, B.A., M.S

Captain, USAF

August 2012

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

AFIT/DS/ENS/12-09

THE AVERAGE NETWORK FLOW PROBLEM: SHORTEST PATH AND

MINIMUM COST FLOW FORMULATIONS, ALGORITHMS, HEURISTICS,

AND COMPLEXITY

Jeremy D. Jordan, B.A., M.S
Captain, USAF

Approved:

//SIGNED//	11 SEPT 2012
Dr. Jeffery D. Weir	Date
Dissertation Advisor	

// SIGNED //	11 SEPT 2012
Dr. Doral E. Sandlin	Date
Committee Member	

// SIGNED //	11 SEPT 2012
Dr. John M. Colombi	Date
Committee Member	

Accepted:

// SIGNED //	11 SEPT 2012
Marlin U. Thomas	Date
Dean, Graduate School of Engineering and Management	

AFIT/DS/ENS/12-09

Abstract

Transportation mode selection is becoming increasingly popular in the field of logistics and operations research. Several modeling challenges exist, one being the consideration of multiple factors into the transportation decision. While the Analytic Hierarchy Process is quite popular in the literature, other multicriteria decision analysis methods such as value focused thinking (VFT) are used sparingly, as is the case across the entirety of the supply chain literature. We provide a VFT tutorial for supply chain applications and a general transportation mode selection value hierarchy. The transportation environment lends itself naturally to network optimization; we therefore integrate multicriteria decision analysis (MCDA), specifically VFT, with the shortest and longest path problem. Since a decision maker desires to maximize value with these techniques, this creates the Multiobjective Average Longest Path (MALP) problem. The MALP allows multiple quantitative and qualitative factors to be captured in a network environment without the use of multicriteria methods, which typically only capture 2-3 factors before becoming intractable. The MALP (equivalent to the average longest path) and the average shortest path problem for general graphs are NP-complete, proofs are provided. The MALP for *directed acyclic graphs* can be solved quickly using an existing algorithm or a dynamic programming approach. The existing algorithm is reviewed and a new algorithm using DP is presented. We also create a faster heuristic allowing solutions in $O(m)$ as opposed to the $O(n^3)$ and $O(nm)$ solution times of the optimal methods. This scaling heuristic is empirically investigated under a variety of conditions and easily modified to approximate the longest or shortest average path problem for directed acyclic graphs. Furthermore, a decision maker may wish to make tradeoffs between increasing value

in the network and decreasing the number of arcs used. We show the DP algorithm and scaling heuristic automatically generate the efficient frontier for this special case of the more general bicriteria average shortest path problem, thus providing an efficient algorithm for this multicriteria problem. Since most organizations ship multiple products, the clear extension to the average shortest path is the average minimum cost flow. Similar to the average shortest path problem, we implement MCDA into the minimum cost flow problem; this creates a multiobjective average minimum cost flow (MAMCF) problem, a problem equivalent to the average minimum cost flow problem. We show the problem is also NP-complete. However, for directed acyclic graphs efficient pseudo-polynomial time heuristics are possible. The average shortest path DP algorithm is implemented in a successive shortest path fashion to create an efficient average minimum cost flow heuristic. Furthermore, the scaling heuristic is used successively as an even faster average minimum cost flow heuristic. Both heuristics are then proven to have an infinitely large error bound. However, in random networks the heuristics generate solutions within a small percentage of the optimal solution. Finally, a general bicriteria average minimum cost flow (BAMCF) problem is given. In the case of the MAMCF, decision makers may choose to minimize arcs in a path along with maximizing multiobjective value. Therefore, a special case of the BAMCF is introduced allowing tradeoffs between arcs and value. This problem is clearly NP-hard, however good solutions are attainable using the information gained from the average minimum cost flow heuristics.

Table of Contents

List of Figures

xiii

List of Tables

THE AVERAGE NETWORK FLOW PROBLEM: SHORTEST PATH AND

MINIMUM COST FLOW FORMULATIONS, ALGORITHMS, HEURISTICS,

AND COMPLEXITY

I. Introduction

1.1   Transportation Mode Selection

Interest in transportation mode selection is increasing as the world becomes more interconnected and supply chains expand globally. Operations research techniques are becoming more popular in logistics, yet much work is still needed. Modeling tradeoffs between cost, speed, and reliability functions is especially important. A popular method of combining qualitative and quantitative factors is through multiobjective decision analysis (MODA), specifically through the use of utility or value functions. The concept of combining a multiobjective programming problem with an additive utility or value function is not new [2, 3, 4, 5], however it remains to be integrated in the context of network optimization. In this dissertation, we combine MODA and network optimization problems and create new formulations not yet discussed in the literature.

Due to the inherent structure present in transportation problems, a minimum cost flow (MCF) optimization model best captures the dynamics of transportation systems. Accounting for multiple objectives in the MCF problem allows conflicting criteria to be represented in transportation planning problems, this problem is called the multiobjective minimum cost flow (MMCF) problem [47]. We take this a step further and create a unique problem we call the multiobjective *average* minimum

cost flow (MAMCF) problem. Hamacher et al [47] present a review of the MMCF including theory and algorithms for solving; their all inclusive reference list doesn't include any papers formulating this problem. Furthermore, very few authors have even combined multiobjective programming and the MCF, although transportation planning frequently requires multiple objective functions such as minimizing cost, minimizing arrival time, minimizing deterioration of goods, and maximizing safety [47].

The difference between the (MAMCF) and the multiobjective minimum cost flow (MMCF) problem is in the average calculation, simple minimum cost algorithms cannot solve MAMCF due to several inherent reasons discussed in the following chapters. An average network flow algorithm is available for the shortest path problem, a special case of the MCF problem, but not for the minimum cost flow problem in general. In addition to the average shortest path algorithm, non-additive shortest path algorithms are available yet are not very efficient. There needs to be an exclusive study of the average shortest path and average minimum cost flow problem.

In addition to the MAMCF, another contribution is the application of value focused thinking to the transportation mode selection problem. Previous research incorporating multiple objectives is limited to the analytic hierarchy process (AHP), an alternative focused method. Alternative focused thinking is concerned with choosing the best alternative among some group of alternatives. AHP is the original and current method of choice for transportation mode selection [69], value focused thinking (VFT) hasn't been used to solve this problem in the literature. Value-focused thinking is different in that it chooses the best alternative *and* determines the value this alternative has in satisfying the problem objectives. This is especially important in transportation mode selection where hundreds of alternatives potentially exist. Rather than choosing the best alternative from a small group, such as that offered by

2

a third party logistics (3PL) provider, value focused thinking (VFT) assigns values to each alternative. If alternatives are low valued, better alternatives can be requested from the 3PL.

Transportation mode selection needs more quantitative methodologies; this is the motivating factor behind the new methodology. While the additive value model from VFT and AHP are quite popular in the literature, no studies exist of its implementation into the shortest path problem or the minimum cost flow problem. A simple substitution into the cost function is not sufficient as we show in the subsequent chapters.

1.2 The Multiobjective Average Shortest Path Problem

The multiobjective average shortest path problem is equivalent to the average shortest path problem; both are unique formulations. In this paper, we combine multiobjective techniques such as the analytic hierarchy process [89] and value focused thinking [61],[60] with the shortest path problem for directed acyclic graphs. The shortest path problem is thus transformed into a highest or longest average path problem in order to maximize value within the network setting, allowing multiple qualitative and quantitative factors to be captured without the use of multicriteria optimization. Shortest and longest path algorithms cannot solve the average shortest or longest path problem. Since the problem encompasses a non-linear cost function, any shortest path algorithm simply chooses a longest or shortest path and ignores the non-linearity of the average function. We prove the average longest and shortest path problems for general graphs is NP-complete; this is shown through a reduction from the Exact Cover by 3-sets problem which is already proven NP-complete. For directed acyclic graphs, an average shortest path algorithm already exists and solves in $O(n^3)$ [108]. A faster algorithm solving in $O(nm)$ is possible using a dynamic programming

approach [81]. The method finds the shortest distance $d(j, k)$ from node s to node j in exactly k arcs. Similarly, it uses a DP approach called the reaching algorithm to obtain the maximum number of arcs to each node. An arc is then scaled by the rank of the incoming node subtracted from the rank of the outgoing node and divided by the rank of the source node. The reaching algorithm is used a second time to solve the scaled network to optimality. Small errors are possible when using the heuristic due to the scaling effect, however theoretically its proven no bound exists on the worst case error unless the arc costs are bound. In practice, the 95% confidence interval on the error bound is 13%, much better than its theoretical bound. Finally, a decision maker may choose to consider the number of arcs in a chosen solution. To accomplish this, a biobjective average shortest path problem is formulated allowing tradeoffs between increasing value and decreasing arcs. This extends the well known biobjective shortest path problem to include an non-linear average objective function.

1.3 The Multiobjective Average Minimum Cost Flow Problem

We extend the multiobjective average shortest path problem to the minimum cost flow. Combining VFT or AHP with the minimum cost flow problem results in a multiobjective average minimum cost flow problem, which is equivalent to the average minimum cost flow problem. In the paper, we formulate these problems and prove them NP-complete by reduction from the exact cover by 3-sets problem. Current minimum cost flow algorithms cannot solve the MAMCF or the average minimum cost flow problem (AMCF) as the problems are non-additive in nature. The average minimum cost flow problem is a special case of the non-additive minimum cost flow problem; specialized heuristics can therefore exploit the problems structure for efficiency. Recall the general additive value model

4

$$v_j(x) = \sum w_i v_i(x_{ij}) \tag{1}$$

for $i = 1, 2, ..., n$ measures and $j = 1, 2, ..., m$ alternatives, where $v_i(x_{ij})$ is the single dimension value function of measure i for alternative j, w_i is the weight of measure i, and $v_j(x)$ is the multiobjective value for alternative j.

Given a graph $G = (V, E)$, the general minimum cost flow (MCF) problem is:

$$\min z = \sum_{(i,j) \in E} c_{i,j} x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = s(i) \ \ \forall \ i \in N$$

$$l_{i,j} \leq x_{i,j} \leq u_{i,j} \ \ \forall \ (i,j) \in E$$

$$\tag{2}$$

where $x_{i,j}$ is the flow from vertex i to j, $c_{i,j}$ is the cost of a unit of flow from i to j, $l_{i,j}$ is the lower bound on the flow from i to j, $u_{i,j}$ is the upper bound on the flow from i to j, and $s(i)$ is the supply or demand at node i. The multiobjective average minimum cost flow formulation combines these two methods.

Similar to the multiobjective average shortest path problem, a bicriteria average minimum cost flow problem is formulated allowing tradeoffs between total arc supply and overall overage value.

1.4 Contributions Summary

1. Value Hierarchy of Transportation Mode Selection - Uses Value Focused Thinking (VFT) to build a construct capable of incorporating any factors deemed important to transportation mode selection.

5

2. A tutorial describing the necessary steps to use VFT for supply chain applications

3. Formulation of the multiobjective average shortest path problem - combines VFT and the shortest path problem to create the multiobjective average shortest path (MASP) problem or multiobjective average longest path (MALP) problem.

4. Algorithm to solve the MASP or MALP in $O(nm)$.

5. Heuristic to estimate the MASP or MALP very accurately in $O(m)$.

6. Formulation of the multiobjective average minimum cost flow problem - combines VFT and the minimum cost flow problem to create the multiobjective average minimum cost flow (MAMCF) problem.

7. Two heuristics to estimate the average minimum cost flow problem - These heuristics combine both an average shortest path algorithm and scaling heuristic with a maximum flow algorithm and solve in $O(nmC)$ and $O(mC)$, respectively. This mimics the well known successive shortest path algorithms for solving the minimum cost flow problem.

8. Formulation and estimate of a bicriteria average minimum cost flow problem - The algorithm solves the MAMCF while using the weighted sum method to make tradeoffs between value and the number of arcs traversed. Pareto optimal solutions are generated allowing the decision maker to tradeoff increases in the value obtained versus the number of arcs used.

9. Computational complexity proofs for the MASP, MALP, and MAMCF problems showing instances where the problems are NP-complete and when solvable in polynomial time.

1.5 Organization

The dissertation is organized into 7 chapters. Chapter 2's literature review covers multiobjective decision analysis, network optimization, multicriteria optimization, computational complexity, and transportation mode selection. Chapter 3 is a tutorial on Value Focused Thinking for Supply Chain Applications, targeted publication is the Journal of Business Logistics. Chapter 4 is an article defining and solving the multiobjective average shortest path problem. Targeted journal is the European Journal of Operational Research. Chapter 5 is a journal article about the multiobjective average minimum cost flow problem. Chapter 6 is an extension of these concepts describing variants and further complexity proofs for the average minimum cost flow problem in general. Finally, Chapter 7 summarizes and describes future research areas. The appendices include a conference proceedings presented at the military applications track at the 2011 Western Decision Sciences Conference discussing potential uses in United States Transportation Command (USTRANSCOM), a conference proceedings from the 2011 Industrial Engineering Research Conference presented in the Multicriteria Optimization track, and a 2012 Industrial & Systems Engineering Research Conference proceedings presented in the Military Applications track.

II. Literature Review

The literature review begins with multiobjective decision analysis and a focus on value focused thinking. Next, network optimization is expounded with a focus on the minimum cost network flow problem and the shortest path problem. We also discuss research from the average shortest path problem and multiobjective minimum cost flows and shortest paths. This leads into a discussion on multiobjective programming, also called multicriteria optimization. Computational complexity theory is then briefly described. Finally, we thoroughly examine measures used to make transportation mode selection decisions since the methodology in this dissertation applies directly to this problem.

2.1 Multiobjective Decision Analysis

Two of the most popular methods of capturing multiple qualitative and quantitative measures are the Analytic Hierarchy Process and Value Focused Thinking. Decision analysis [16] is a powerful and widely used technique in Operations Research ([52],[19]), but recently has been defined further [59] as "a set of quantitative methods for analyzing decisions based on the axioms of consistent choice. This excludes techniques such as AHP, fuzzy sets, MCDM, traditional math programming, and other useful decision making techniques. Value-Focused Thinking and decision analysis seek instead to aid in human decision making, not model the human decision making process. The justification for this purpose is decision makers should desire to make rational choices given any situation. The axioms of consistent choice from [65] where $\succ$ means some consequence c is preferred are:

1. (Transitivity) If $c_i \succ c_j$ and $c_j \succ c_k$, then $c_i \succ c_k$

2. (Reduction) If the rules of probability can be used to show two alternatives have the same probability for each c_i, then the two alternatives are equally preferred.

3. (Continuity) If $c_i \succ c_j \succ c_k$, then a p exists such that an alternative with a probability p of yielding c_i and a probability of 1-p of yielding c_k is equally preferred to c_j.

4. (Substitution) If two consequences are equally preferred, one can be substituted for the other in any decision without changing the preference ordering of alternatives.

5. (Monotonicity) For two alternatives that each yield either c_i or c_j, where $c_i \succ c_j$, the first alternative is preferred to the second if it has a higher probability of yielding c_i.

These axioms hold true for many decisions, it can even be argued they hold for any decision.

2.1.1 Value Focused Thinking.

2.1.1.1 VFT Overview.

VFT has several important benefits as shown in Figure 1 from [61], three of which are uncovering hidden objectives, creating alternatives, and improving communication; these don't commonly come to mind in most decision support studies. The pervasiveness of multiple alternatives lures decision makers away from thinking about fundamental objectives, and traps them in paradigmatic thought processes. By improving communication through the VFT modeling process, new ideas emerge, objectives are uncovered, and alternatives are generated.
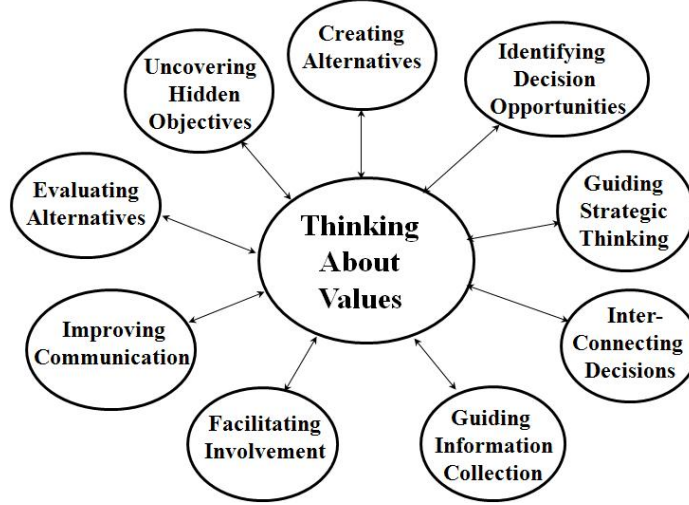
Figure 1. FIGURE: VFT BENEFITS

A list of suggested implementation procedures is always useful. [65] and [61] describe the steps in the VFT process rather generally, a more specific declaration of events is given by [96] in Figure 2. This detailed process shows 10 essential steps in a value model study. Problem identification (Step 1) can be more challenging than it seems. Creating the value hierarchy (Step 2), developing measures (Step 3), creating value function (Step 4), and weighting the hierarchy (Step 5) are critical to model success. Following model construction is the generation (Step 6) and scoring (Step 7) of alternatives through deterministic analysis (Step 8). Because of the subjectiveness in defining values and weights, a proper sensitivity analysis (Step 9) is essential for a good analysis. Finally, conclusions and recommendations (Step 10) are help communicate the results to decision makers.
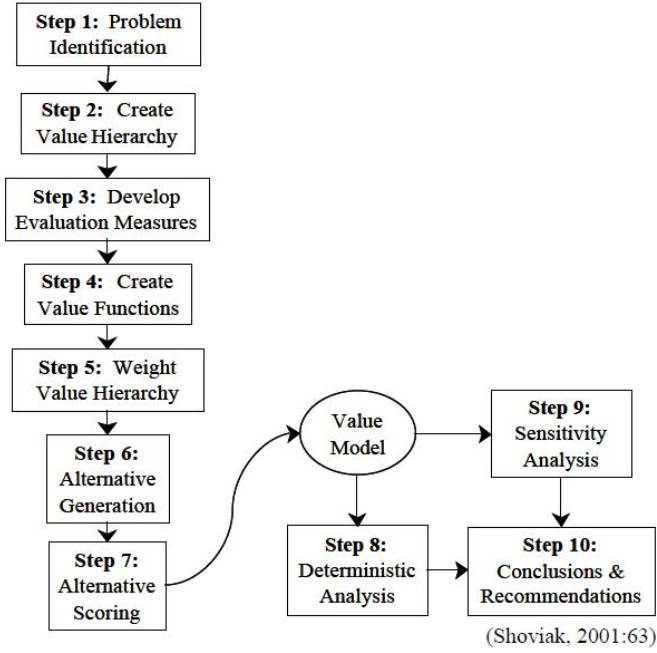
Figure 2. Steps in VFT Study

Two approaches are possible when building a hierarchy, bottom-up and top-down. When alternatives to a decision problem are previously known, a bottom-up method is used to establish ways they are different. The important ways in which alternatives differ end up being the measures used to compare them. This "bottom up" approach is called such because the value hierarchy is built from the measures up to the objectives. Conversely, a top-down approach begins with the fundamental objectives and works down to the measures.

Keeney [61] defines the properties of a good value hierarchy as completeness, non-redundancy, independence, operability, and small size. Completeness ensures that every important objective and measure valued by the decision maker is accounted for. A good hierarchy should have mutually exclusive or non-redundant measures; large overlaps between measures are not preferred. This is inevitable in some cases however. If measures must overlap, its important the decision maker is aware of the implications on the model, this being the combined weights of the measures

11

in the overall decision. In addition to non-redundancy and completeness, a value hierarchy should be operable or understandable by all interested parties and small in design. Another property is small size. This may seem non-intuitive as people normally enjoy building complex models, even though model sparsity usually results in a better solution. Smaller hierarchies are easier to communicate and usually have enough information to make good decisions. The art of value modeling lies in the ability to balance the defensibility and practicality of a model. The final property, independence, requires the satisfaction of several mathematical assumptions. If the assumptions fail, other models such as multiplicative or multilinear models may be used. We only discuss 3 of the important mathematical requirements of the additive value model in this section, more info can be found in [65].

Definition 1 *A function $v(x)$ is a value function if $v(x') > v(x'')$ if and only if $x' \succ x''$, where x' and x'' are specified but arbitrary levels of x.*

In order to use this additive value model, measures must be preferentially independent. We first discuss the corresponding tradeoffs condition that must hold when dealing with two measures. Two measures X and Y hold to the corresponding tradeoffs condition if: for any levels x_1, x_2, y_1, and y_2 of the measures, if $(x_1, y_1) \sim (x_1 - a, y_1 + b)$ and $(x_2, y_1) \sim (x_2 - d, y_1 + b)$, then for c such that $(x_1, y_2) \sim (x_1 - a, y_2 + c)$ it is true that $(x_2, y_2) \sim (x_2 - d, y_2 + c)$. This condition must hold for any x_1, x_2, y_1, and y_2. Thankfully, for 3 or more measures, this need not be shown, only mutual preferential independence is needed.

Definition 2 Preferential independence: *Suppose that Y and Z are a partition of $X_1, X_2, ..., X_n$, each X_i being in exactly one of Y or Z. Then Y is preferentially independent of Z is the rank ordering of alternatives that have common levels for all attributes in Z does not depend on these common levels. (The common levels do not*

have to be the same for different attributes, but the level of each X_i in Z is the same for all alternatives.)

Given this, we can now define mutual preferential independence.

Definition 3 Mutual Preferential independence: *A set of attributes $X_1, X_2, ..., X_n$ displays mutual preferential independence if Y is preferentially independent of Z for every partition Y, Z of $X_1, X_2, ..., X_n$.*

For proofs, see [65].

2.1.1.2 Measures and Objectives.

Determining good objectives is critical to the accuracy of the model. Initially, the fundamental objectives of the problem must be identified. For instance, in a logistics application such as supplier selection, the overall objective of one organization may be to minimize costs while another seeks to maximize customer satisfaction. These differing views on the purpose of supplier selection may affect the weighting and value function discussed in the next section. It's important to ask the decision maker why he feels that an objective is important and what is trying to be accomplished through that objective. Doing this ensures the decision maker thinks through the problem completely, and drills down to the actual overall objective. Attainment of the fundamental objective is achieved through further objectives called means objectives, while achievement of the means objectives is gauged through measures. Measures themselves are generally quite easy to generate; the difficulty comes in deciding which measures should be included in the value hierarchy. Including every possible measure ensures completeness but increases difficulty in weighting. Less measures are preferred given they adequately represent the decision problem. When deciding what measures to include, use the "test of importance" [60], evaluations should only be included if for

two given alternatives, a change in the measure could change the preference between the two.

Types of measures are natural or constructed, and direct or proxy. Natural scales need not be produced, that is they are "naturally" occurring. Examples include cost in dollars, container loads shipped, and time after due date. A constructed scale on the other hand does not exist and must be developed for a measure if a natural scale isn't available or practical. Natural and constructed scales can be either direct or proxy. Direct scales measure the direct attainment of an objective whereas proxy scales measure the degree of attainment of an objective. Natural scales are clearly objective in that a clairvoyant, able to see the future, will score an alternative identically, now and in the future, unless the levels of that measure change with time. Constructed scales are different. There needs to be a test for clairvoyance when setting levels on these scales, allowing subjectivity into the model introduces noise. Simply allowing an individual scorer to assign a value of high or medium based on personal preferences creates a poor model, taking away from the advantages of using VFT.

A general hierarchy is provided in Figure 3 showing the breakdown of objectives, measures, tiers and branches of the hierarchy, and the global and local weights discussed in Section **??**.
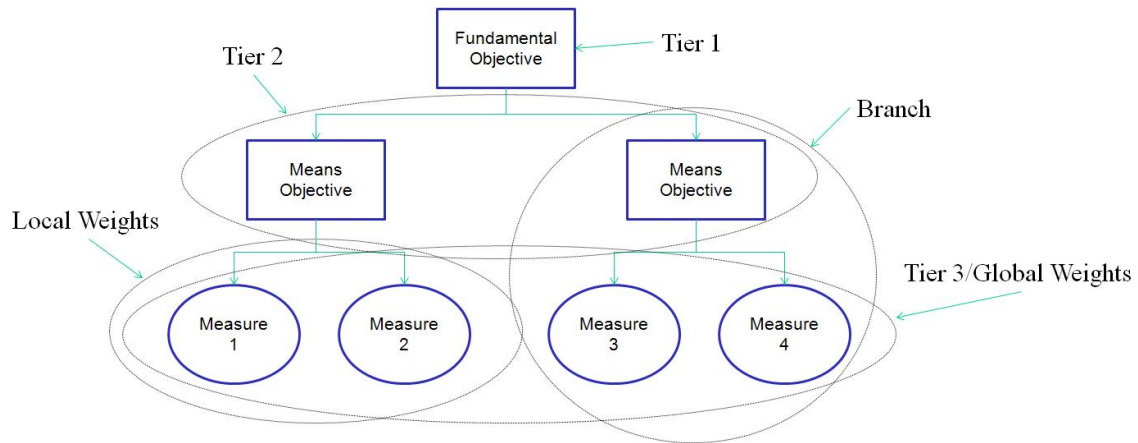
Figure 3. General Value Hierarchy

2.1.1.3 Assigning Values to Measures.

Eliciting values from a decision maker can be complicated. Single dimension value functions (SDVF) are normally used to capture preferences for varying levels of the measures, assigning each level of the function a value between 0 to 1 . These functions can be monotonically increasing or decreasing. For instance, performance is a monotonically increasing function because higher values are more desirable, whereas cost is monotonically decreasing, because higher levels are less desirable. Two popular SDVF's [65] are piecewise linear and exponential. For an in depth look at values and preference functions see Keeney and Raiffa [60]. Continuous measures are captured by either the piecewise linear or exponential functions, preferable the exponential, depending on the preferences of the decision makers. For cases where measures are not continuous, or preferences do not match the exponential function, [65] suggests using the piecewise linear SDVF.

When measure levels are discrete, a piecewise linear SDVF is best. Values are assigned to different levels of the measure. See Figure 5 for an example with of 5 levels and values. Setting the scales up is simple as well using three steps:

1. Place each value increment in order - smallest to largest value increment

2. Scale each value increment as a multiple of the smallest value increment (eg. 2:1 or 10:1 or 4:1)

3. Sum the value increments to one and solve for the smallest value increment

Many of the measures we encounter are continuous in nature. The exponential SDVF is equipped to handle preferences on a continuous scale and is easy to explain to a decision maker. Initially, as with the piecewise linear SDVF, high and low levels need definition. For monotonically increasing measures, the continuous exponential function is
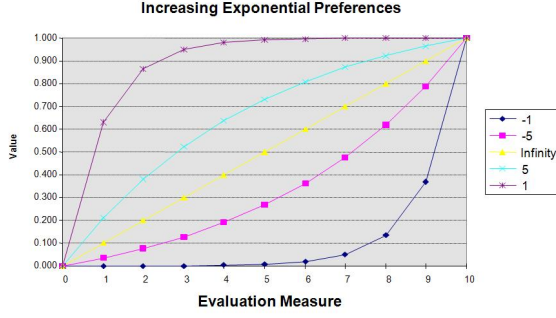
Figure 4. Exponential Value Function
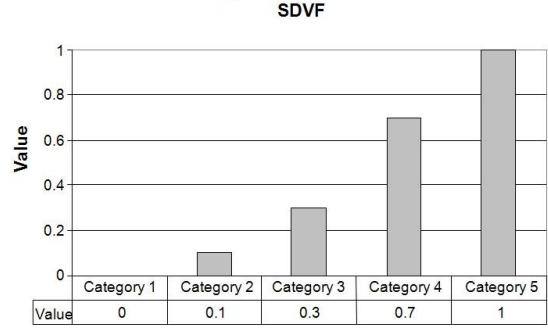
Figure 5. Piecewise Linear Value Function

$$v_i(x_i) = \begin{cases} \frac{1-e^{-(x_i-x_L)/\rho}}{1-e^{-(x_H-x_L)/\rho}} & \rho \neq \text{Infinity;} \\[2ex] \frac{x_i-x_L}{x_H-x_L}, & \text{otherwise.} \end{cases} \tag{3}$$

where x_H is the most preferred level (assigned a value of 1), x_L is the least preferred level (assigned a value of 0), x_i is the level of the ith measure, ρ is an unknown parameter, and $v_i(x_i)$ is the value of the ith measure at level x_i. With this equation, any continuous measure is valued, however the equation must be solved for the unknown parameter ρ. Unfortunately, no closed form solution exists. In practice, ρ can be estimated if the decision maker can assign a value to a mid-level between x_H and x_L. Figure 4 shows the shape of the exponential value function for differing levels of ρ. See p. 68 in [65] for methods to find ρ.

2.1.1.4 Determining Weights.

Next, weights must be assigned to each of the measures. There are several means to determining weights, AHP, swing weights, direct assessment, or group weights to name a few. Note, AHP can be used to determine weights for a value hierarchy but this differs from using it to evaluate alternatives. An actual AHP model requires the decision maker to make pairwise comparisons across all combinations of the possible

16

alternatives, whereas VFT provides a value for each of any large number of alternatives. The AHP technique in this case is used only for weights. Measures can weighted globally or locally; the former compares all measures simultaneously, the latter looks at each measure in the context of its means objectives. Local weighting is preferred as it tends to be easier for decision makers to weight measures within objectives rather than across all objectives. In turn, this provides more accurate weights, and global weights are easily calculated from local weights. Either way, the global weights are needed to calculate the overall value function. w_i is the notation used for the weight of measure i.

An easy method to implement for determining weights is direct assessment. This is accomplished through examining the measures from one means objective and weighing tradeoffs between them. The least important measure is assigned a 1, the remainder of the measures are assigned numbers based on how much more important they are than the least important measure. A measure that is twice as important as the least important measure is given a value of 2, call it r_i for measure i, and so on. Since the weights must sum to one, that is $\sum w_i = 1$, these values must be scaled to a decimal between 0 and 1. To determine a weight w_i for measure i, take its successive r_i and divide by the sum of all the ranks, that is

$$w_i = \frac{r_i}{\sum r_i}. \tag{4}$$

. This works well for most cases, if consensus isn't reached, a more robust method should be used, either swing weights or AHP.

The standard swing weights method for determining decision maker weights is popular. Advanced swing weight methods have been developed as well but are not covered here, see for instance Parnell's method [83] which uses a matrix like that in Figure 7. The standard method begins by building a table like that in Figure 6, where

17

each of the measures 1 through n are set at high to create a list of n hypothetical alternatives. Where one measure is set to high, the rest of the measures are set to their lowest level. Next, each of these alternatives are ranked from 1 to n, with 1 representing the best alternative and n the worst case.

Following the ranking assignments, each measure is assigned a rate. The baseline, or worst alternative, is assigned a rate of 0 and the highest ranking measure is assigned a weight of 100. The decision maker is then asking the following questions: how much less satisfaction do you get from swinging a lower ranked measure versus swinging the highest ranked measure?; If swinging the top ranked measure from low to high gives 100% satisfaction, what percentage satisfaction do you get from swinging the lower ranked measure?. After assignment of all rates, the weight is calculated as a ratio of

$$\frac{Rate_i}{\sum_{i=1}^{n} Rate_i}.$$
(5)

Figure 6. Swing Weights Standard Method

Figure 7. Swing Weights Matrix Method

The AHP method of determining weights is not covered here, as applications are plentiful for logistics. Application is similar to the methods used in a popular AHP supplier selection application [45], however the AHP is NOT used to rank alternatives.

For en extensive list of application papers see [50]. With weights and values for each of the measures defined, we can calculate the overall value of each of the alternatives.

2.1.1.5 Mathematical Formulation of the Model and Scoring.

The value of an alternative is given by inserting levels of the weights w_i and the values of the levels of each measure x_i, or $v_i(x_i)$. Doing this for each alternative creates the additive value model:

$$v_j(x) = \sum w_i v_i(x_{ij}) \tag{6}$$

for $i = 1, 2, ..., n$ measures and $j = 1, 2, ..., m$ alternatives, where $v_i(x_{ij})$ is the single dimension value function of measure i for alternative j, w_i is the weight of measure i, and $v_j(x)$ is the multiobjective value for alternative j.

After inserting all inputs into the equation above, or performing Deterministic Analysis as referred to in Figure 2, an overall score is obtained for each alternative being compared. The meaning of the final score is straightforward, it's the amount of value the alternative provides as a solution to the problem.

2.1.1.6 Sensitivity Analysis.

Because of the subjectivity of the VFT modeling process, a requisite sensitivity analysis is needed to reveal the effects of changes in value functions and weighting schemes, primarily weighting schemes as these produce greater changes. The idea is to vary the weights w_i of Equation 6 for each evaluation measure and determine at what point changes in the most valued alternative occur. Several techniques are currently used for sensitivity analysis including math programming [79], algorithms [11], and simulation [4], [55], [73]. Historically, sensitivity analysis has been limited to one-way analysis, changing only one weight at a time [15]. However, new techniques

are emerging (TRIAGE Method, COSA method, VBR Method) that advance the ability to conduct sensitivity analysis.

2.2   Network Optimization

Modeling an optimization problem as a network allows the exploitation of special structures in order that faster solutions be found. Several structures exist, this section provides background on the shortest path and minimum cost flow problems. All minimum cost flow and shortest path problems are simply special cases of the linear program, however in many cases faster algorithms can be used in place of general LP methods.

Network optimization in the modeling of a transportation network is common, many textbooks use transportation as an application [3], [98]. Multiple articles apply network optimization to the intermodal transportation problem as well. [46] models the multimodal multiproduct network for strategic planning and minimizes cost, yet they do not to consider multiple objectives, especially intangibles. Many papers call for increases in the number of factors incorporated into the transportation mode selection decision.

2.2.1   Shortest Path Problem.

Assume a graph G $= (V, E)$ to be a directed network with $|V| = n$ vertices and $|E| = m$ edges. Each edge or arc $(i, j) \in E$ has some cost $c_{i,j}$ and flow $x_{i,j}$ associated with its use. The source and sink vertices are designated s and t, respectively.

2.2.1.1   Linear Programming Formulation.

The general shortest path problem is:

$$\min z = \sum_{(i,j) \in E} c_{i,j} x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{j,i} = \begin{cases} 1, & \text{if i=s;} \\ -1, & \text{if i=t;} \\ 0, & \text{otherwise} \quad \forall i \end{cases}$$

$$x_{i,j} \geq 0 \quad \forall \ (i,j) \in E$$

$$(7)$$

where $x_{i,j}$ is the flow from vertex i to j and $c_{i,j}$ is the cost of a unit of flow from i to j. This is easily solved using the simplex method or any other LP algorithm. However, since it maintains a unique structure, faster algorithms are available. Label setting algorithms are most efficient in solving shortest path problems, the most popular being Dijkstra's algorithm [3]. Even faster algorithms exist if the graph has the property of being acyclic, or without cycles.

2.2.1.2 Reaching Algorithm for Directed Acyclic Graphs.

A dynamic programming algorithm that solves the directed acyclic graph (DAG) in linear time $O(m)$ is called the Reaching Method. It begins with a topological sort of the nodes and labels each vertex successively. Easily adaptable to the longest or shortest path, the algorithm configured for the longest path:

Step 1: Topologically order the DAG G

Step 2: For $i = 1, ..., n$, set dist(i)=0

Step 3: For $i = 1, ..., n-1$, for each edge $V(i), u$ outgoing from $V(i)$, if dist(V_i) + $G(V_i, u) >$ dist(u), then set dist$(u) =$ dist$(V_i) + G(V_i, u)$

Step 4: dist(t) is the longest path to the sink t

The algorithm above is easily modified to gather the shortest path to every node as well. This is the underlying dynamic programming approach behind the proposed scaling heuristic.

2.2.1.3 Non-Additive Shortest Paths.

The idea of non-additive paths is a relatively recent development in the network optimization literature. Research motivation comes from the fact that not all network paths are additive in nature, that is a path cost may be some function other than simply the addition of all the arcs costs. The methods were sparked by the traffic equilibrium problem and are discussed in [40], [1], and [41]. There are very few citations to these articles, likely because motivation is lacking. The average path is actually just a special case of the non-additive shortest path discussed in [40] and [1], where the function is simply the sum of arc costs of the path divided by the number of arcs. Because of its simple structure, faster solutions should be attainable.

2.2.1.4 Average Shortest Path Length.

[108] lays out the foundation for determining the optimal average path length through the path length minimization algorithm. This algorithm pursues the optimal path by determining the best average path of cardinality j at each node. Note, the algorithm is only useful in directed acyclic graphs.

First, each vertex is assigned a rank according to its maximum cardinality (number of arcs) of a path from s to the vertex. The source, s, is obviously assigned a rank of 0 and the sink, t has the highest rank. Vertices are numbered according to their rank, starting with s, and numbering vertices with equal ranks arbitrarily. So, s is

numbered 1, t is numbered $|U|$ (number of vertices) and for every arc $e(u,v)$ the vertex u is assigned a smaller number than the vertex v.

Define $G = (U, E)$. Let u be a vertex on some path from source s to sink t. Only the shortest path with cardinality j can be part of the shortest average arc length path from s to t. Each vertex $u \in U$ is next assigned a vector $L(u)$ of length $|U|$. The jth element of $L(u)$, $L_j(u)$ with $(0 \leq j \leq |U|-1)$, is the minimum length of any path from s to u with cardinality j. $\Pi_j(u)$ denotes the minimum length path or paths. Since $G(U, E)$ is acyclic, the cardinality of a path cannot be greater than $|U| - 1$. If no path exists for a cardinality, the path is assigned ∞. Another vector P_u of length $|U|$ is associated with u, whose jth element $P_j(u)$ is the last vertex preceding u on $\Pi_j(u)$. This is the vertex v for which $L_{j-1}(v) + l(e(v,u)) = L_j(u)$. If $L_j(u) = \infty$, then $P_j(u) = 0$

Starting at s, a new vertex is marked at each iteration until t is reached. Once a vertex u is labeled, the length of the shortest path from s to u is known for every cardinality between 0 and $|U| - 1$. The sets of arcs entering and leaving $u \in U$ are denoted $\Gamma^{in}(u)$ and $\Gamma^{out}(u)$. The algorithm is as follows:

Step 0: Initialization. Set $L_0(s) = 0$ and $L_j(s) = \infty$, $1 \leq j \leq |U| - 1$. Mark s and set $T = U - \{s\}$. For every $u \in T$ set $L_j(u) = \infty$, $0 \leq j \leq |U| - 1$. For every $u \in U$ define $P_j(u) = \emptyset$, $0 \leq j \leq |U| - 1$.

Step 1: New Vertex Selection. Find a vertex $u \in T$ for which all the tail vertices of the arcs in $\Gamma^{in}(u)$ are already marked. Such a vertex must exist since $G(E, U)$ is an acyclic digraph with a single source and a single sink whose vertices are numbered as described above.

Step 2: Updating the minimum path lengths. Determine the shortest path length

23

vector L_u by considering every vertex v for which $e(v, u) \in \Gamma^{in}(u)$ as follows.

$$L_j(u) = min\{L_{j-1}(v) + l(e(v, u))|e(v, u) \in \Gamma^{in}(u)\}, \quad 1 \leq j \leq |U| - 1. \qquad (8)$$

Let v^* be the vertex obtained when solving 8 for given u and j. Then, set $P_j(u) = v^*$.

Step 3: Updating the set of marked vertices. Mark u and set $T = T - \{u\}$.

Step 4: Termination Test. If $u = t$ then go to Step 5, else go to Step 1.

Step 5: Retrieving the minimum average arc length path. Upon termination, every $L_j(t) \leq \infty$ is the length of the shortest path from s to t among all the paths of cardinality j. For every j satisfying $L_j(t) = \infty$ there exists no path of cardinality j from s to t. Evidently, $min\{L_j(t)/j|1 \leq j \leq |U| - 1$ yields the minimum average arc length for any path from s to t. Let j^* be the cardinality of the path for which the minimum average arc length was obtained. Then, the desired path is retrieved by traversing backwards from t to s as follows. We start from t and go backwards to the vertex stored in $p_{j^*}(t)$. We then go backwards to the vertex stored in $P_{j^*-1}[P_{j^*}(t)]$ and continue in the same manner until s is reached.

2.2.1.5   Discussion of Outputs of Wimer's Algorithm.

As noted above, the algorithm produces several matrices, we add information in this section for clarity. The first matrix is an L matrix as shown in Table 1, where $L_j(u)$ is the shortest path from the source s to vertex u of cardinality j, assigned some value or ∞ if the path doesn't exist ($-\infty$ for the max problem).

A matrix $\prod_j(u)$, representing the path that yields the elements of the L matrix in Table 1, is shown in Table 2. Table 3 shows an example $\prod$ matrix.

Next, a P matrix is generated indicating the vertex preceding u for each node u for each cardinality j. Table 4 gives the P matrix. An example is given in Table 5

Table 1. L Matrix produced by Wimer's Algorithm

		# of Arcs j				
		1	2	3	...	q
	2	$L_1(2)$	$L_2(2)$	$L_3(2)$	...	$L_{|U|-1}(2)$
	3	$L_1(3)$	$L_2(3)$	$L_3(3)$		$L_{|U|-1}(3)$
Node # u	4	$L_1(4)$	$L_2(4)$	$L_3(4)$		$L_{|U|-1}(4)$
	⋮	⋮			⋱	⋮
	N	$L_1(N)$	$L_2(N)$	$L_3(N)$	...	$L_{|U|-1}(N)$

Table 2. $\prod$ Matrix produced by Wimer's Algorithm

		# of Arcs j				
		1	2	3	...	q
	2	$\prod_1(2)$	$\prod_2(2)$	$\prod_3(2)$	...	$\prod_{|U|-1}(2)$
	3	$\prod_1(3)$	$\prod_2(3)$	$\prod_3(3)$		$\prod_{|U|-1}(3)$
Node # u	4	$\prod_1(4)$	$\prod_2(4)$	$\prod_3(4)$		$\prod_{|U|-1}(4)$
	⋮	⋮			⋱	⋮
	N	$\prod_1(N)$	$\prod_2(N)$	$\prod_3(N)$	...	$\prod_{|U|-1}(N)$

Table 3. Example $\prod$ Matrix

		# of Arcs j				
		1	2	3	...	q
	2	1-2	∞	∞	...	∞
	3	1-3	1-2-3	∞		∞
Node # u	4	1-4	1-2-4	1-2-3-4		∞
	⋮	⋮			⋱	⋮
	N	1-N	1-5-N	1-4-8-N	...	1-3-5-7-...-N

that follows the example $\prod$ matrix in Table 3.

Table 4. P Matrix produced by Wimer's Algorithm

		# of Arcs j				
		1	2	3	...	q
	2	$P_1(2)$	$P_2(2)$	$P_3(2)$	...	$P_q(2)$
	3	$P_1(3)$	$P_2(3)$	$P_3(3)$		$P_q(3)$
Node # u	4	$P_1(4)$	$P_2(4)$	$P_3(4)$		$P_q(4)$
	⋮	⋮			⋱	⋮
	N	$P_1(N)$	$P_2(N)$	$P_3(N)$	...	$P_q(N)$

Assign another matrix Z, call each of its elements $Z_j(u)$, where each element is

Table 5. Example P Matrix

		# of Arcs				
		1	2	3	...	q
	2	1	0	0	...	0
	3	1	2	0		0
Node #	4	1	2	3		0
	⋮	⋮			⋱	⋮
	N	1	5	8	...	7

the arc $x_{i,j}$ combining the vertex preceding $u(P_j(u)$ from Table 4) with u. This gives $x_{i,j} = x_{(P_j(u),(u))}$. Refer again to the example $\prod$ matrix in Table 3, an example Z matrix is given in Table 6.

Table 6. Example Z Matrix

		# of Arcs j				
		1	2	3	...	q
	2	$x_{1,2}$	∞	∞	...	∞
	3	$x_{1,3}$	$x_{2,3}$	∞		∞
Node # u	4	$x_{1,4}$	$x_{2,4}$	$x_{3,4}$		∞
	⋮	⋮			⋱	⋮
	N	$x_{1,N}$	$x_{5,N}$	$x_{8,N}$	...	$x_{7,N}$

Every remaining arc in Z from Table 6 is part of at least one optimal path of some cardinality j. One final perturbation to the L matrix in Table 1 is needed, divide each element by its cardinality j. Table 7 gives the resulting matrix, call it LJ, the best average value from the source s to the node u.

Table 7. LJ Matrix produced by Wimer's Algorithm

		# of Arcs j				
		1	2	3	...	q
	2	$\frac{L_1(2)}{1}$	$\frac{L_2(2)}{2}$	$\frac{L_3(2)}{3}$	...	$\frac{L_q(2)}{q}$
	3	$\frac{L_1(3)}{1}$	$\frac{L_2(3)}{2}$	$\frac{L_3(3)}{3}$		$\frac{L_q(3)}{q}$
Node # u	4	$\frac{L_1(4)}{1}$	$\frac{L_2(4)}{2}$	$\frac{L_3(4)}{3}$		$\frac{L_q(4)}{q}$
	⋮	⋮			⋱	⋮
	N	$\frac{L_1(N)}{1}$	$\frac{L_2(N)}{2}$	$\frac{L_3(N)}{3}$	...	$\frac{L_q(N)}{q}$

With this algorithm we've essentially removed any arcs that are NOT part of any of the optimal paths for each cardinality j. The final row of the Matrix in Figure 7 is the shortest average path to node N in some number of arcs.

2.2.1.6 Multiobjective Shortest Paths.

Research in multiobjective optimization for the shortest path has traditionally been limited to problems with 2 or 3 objectives, as more objectives make the problem intractable. This is the beauty of the additive value function, many objectives are incorporated into one function. A downfall most authors allude to is the decision makers' value function must be known a priori. Several authors [27] [100] [72] [26] relay the importance and scarcity of treatment in the literature of the multiobjective shortest path. See [35] for complexity proofs and further discussion.

2.2.2 Minimum Cost Network Flow Problem.

Assume a directed acyclic graph $G = (V, E)$. The linear programming formulation of the general minimum cost flow problem is:

$$\min z = \sum_{(i,j) \in E} c_{i,j} x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = s(i) \ \ \forall \ \ i \in N$$

$$l_{i,j} \leq x_{i,j} \leq u_{i,j} \ \ \forall \ \ (i,j) \in E$$

$$(9)$$

where $x_{i,j}$ is the flow from vertex i to j, $c_{i,j}$ is the cost of a unit of flow from i to j, $l_{i,j}$ is the lower bound on the flow from i to j, $u_{i,j}$ is the upper bound on the flow from

i to j, and $s(i)$ is the supply or demand at node i. Just as with the shortest path problem, the simplex algorithm or any other LP algorithm will solve the minimum cost flow problem. However, due to its special structure, much faster algorithms have been developed.

2.2.2.1 Solution Algorithms.

The linear structure of the minimum cost network flow (MCNF) problem allows the use of the Simplex method. However, when dealing with large scale problems, the simplex method is inefficient as its theoretical run time is exponential. Speedier adaptations of the simplex method exist for the MCNF problem. More popular are successive shortest path algorithms, where an implementation of Dijkstra's algorithm is run perhaps, and this path is assigned flow up to its capacity. The shortest path algorithm is repeated while updating the arc capacities. For directed acyclic graphs, the reaching algorithm from Section 2.2.1.2 can be used successively. Because the constraint set of the MCNF problem is unimodular in nature, any LP algorithms produce integer solutions.

2.2.2.2 Average Minimum Cost Network Flow Problem.

Although this problem is never formally introduced in the literature, a few articles mention it in passing. [39] creates algorithms to find the weighted minimal cost flows, which could be useful for computing the minimum average cost but is computationally expensive. Additionally, it's only cited 3 times and only cited by the authors in their other papers. Another minimal average cost flow variant was first formulated to minimize a total cost, consisting of a fixed cost of using a network plus a variable cost per unit of flow, divided by the total flow [12]. These do not address the average minimum cost flow problem in this dissertation.

2.2.2.3 Multiobjective Minimum Cost Network Flow Problem.

A comprehensive review of algorithms and research on the multiobjective minimum cost flow (MMCF) problem reveals a threshold of two objectives for most algorithms [47]. Hamacher [47] unveils the need for algorithms solving more than 2 objectives.

2.2.2.4 Transportation Problem.

Transportation problems are a special case of the minimum cost flow problem, where each of the nodes can be classified into one of two sets, call them N_1 and N_2, with m and n nodes respectively. Let x_{ij} and c_{ij} represent the amount shipped and cost of shipment, respectively, from a supply location i to a demand location j. Assuming supply equals demand, the basic LP formulation of transportation problem is

$$\min \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\text{s.t.} \sum_{j=1}^{n} x_{ij} = a_i \quad \forall i$$

$$\sum_{j=1}^{n} x_{ij} = b_j \quad \forall j$$

$$x_{ij} \geq 0$$

(10)

where a_i is the total supply and b_j is the total demand.

2.2.3 Fixed Charge Problems.

The Fixed Charge Problem, initially formulated by Hirsch and Dantzig [49] [97] and solved using a variety of approaches [78] [18] [31] [106], is a generalization of the fixed charge transportation problem (FCTP) and the fixed charge network flow problem (FCNFP). The FCTP and FCNFP are also a special case of the minimum cost flow problem (MCFP), see [48]. Several other names exist as well and there are multiple variations of the FCNFP.

The capacitated multicommodity fixed-charge network design (CMND) (or capacitated single-facility network design problem) problem is similar to the fixed charge network flow problem (FCNFP) [13], however multiple commodities are considered. Due to the similarity of their structure, solution techniques may complement one another. The methods developed for network design problems are similar to our multicriteria formulations [70] [21] [22].

Costa's paper [20] summarizes applications of Bender's decomposition on the fixed charge network design problem and all its variants. Its application to network design problems is rare but successful in the reviewed applications. She discusses applications of Benders to the slightly more complicated capacitated single-facility network design problem with multiple commodities, but not to the fixed charge network flow problem.

2.3 Other General Optimization Areas

This section discusses other relevant used throughout the dissertation including general multiobjective programming, large scale optimization, and sensitivity analysis.

2.3.1 Multiobjective Programming.

Multiobjective programming, also called multicriteria optimization, multiobjective optimization, and multicriteria programming, gives the added benefit of examining pareto optimal or efficient solutions [35]. Decision makers are able to make trade-offs between competing objectives and see the outcomes of these tradeoffs. [35] defines the types of efficiency for a feasible solution $\hat{x} \in X$:

- weak efficiency - no other $x \in X$ such that $f(x) < f(\hat{x})$

- strict efficiency - no other $x \in X$, $x \neq \hat{x}$ such that $f(x) \leq f(\hat{x})$

- proper efficiency - there is a real number $M > 0$ such that $\forall$ i and x $\in$ X satisfying $f(x) < f(\hat{x})$, there exists an index j such that $f(\hat{x}) < f(x)$ and $\frac{f(x)-f(\hat{x})}{f(\hat{x})-f(x)} \leq$ M.

The most popular method of generating the efficient set or set of pareto optimal solutions are though scalarization techniques, the most common being:

- Weighted sum method - good for convex problems

- Epsilon Contraint, Hybrid, Elastic Constraint - good in the absence of convexity

- Benson's method, compromise programming, achievement function method

The general multiobjective program is:

$$\min \ (f_1(x), ..., f_k(x))$$
$$\text{st } g_j(x) \leq 0$$

for $j = 1,...,$m constraints and $k = 1,...,$p objective functions.

2.3.1.1 Multiobjective Linear Programming.

For the multiobjective linear program (MOLP), the objective functions from the general multiobjective problem in Formulation 2.3.1 become:

$$f_k(x) = c_k^T x \text{ for } k = 1,...,\text{p.}$$

Adding slack variables to the constraints gives:

$$g_j(x) + s_j = 0 \text{ for } j = 1,...,\text{m,}$$

and the constraints can now be written in matrix form:

$$Ax + Is = b$$

Thus, the general multiobjective linear program is:

$$\min Cx$$
$$\text{st } Ax + Is = b$$
$$x \geq 0$$

where

A - $m \times n$ matrix

I - $m \times m$ identity matrix

$x \in \mathbb{R}^n$

$s \in \mathbb{R}^m$

C - $p \times n$ matrix, that is $C = \begin{pmatrix} c_1^T \\ c_2^T \\ \vdots \\ c_p^T \end{pmatrix}$

$X = \{x \in \mathbb{R}^n : Ax + Is = b, x \geq 0, s > 0\}$ is the feasible set in decision space

$Y = \{Cx : x \in X\}$ is the feasible set in objective space

The main assumption with multiobjective linear programming is if $X_k := \{\hat{x} \in X : c_k^T \hat{x} \leq c_k^T x \ \forall \ x \in X\}$, then no $\hat{x} \in X$ such that $\hat{x} \in X_k \ \forall \ k = 1, 2, ..., p$. That is, no unique solution minimizes all k objective functions or the intersection of the solution sets is $\emptyset$,

$$\bigcap_{k=1}^{p} X_k = \emptyset.$$

Therefore, a true MOLP possesses competing objectives, so y^I, the ideal point, is not in the objective space feasible set Y.

2.3.1.2 MOLP Efficiency.

Definition 6.2. Let $\hat{x} \in X$ be a feasible solution to a MOLP and let $\hat{y} = C\hat{x}$.

1. $\hat{x}$ is called weakly efficient if there is no $x \in X$ such that $Cx < C\hat{x}$; $\hat{y} = C\hat{x}$ is called weakly nondominated.

2. $\hat{x}$ is called efficient if there is no $x \in X$ such that $Cx \leq C\hat{x}$; $\hat{y} = C\hat{x}$ is called nondominated.

3. $\hat{x}$ is called properly efficient if it is efficient and if there exists a real number $M > 0$ such that for all i and x with $c_i^T x < c_i^T \hat{x}$ there is an index j and $M > 0$ such that $c_j^T x > c_j^T \hat{x}$ and
$$\frac{c_i^T \hat{x} - c_i^T x}{c_j^T x - c_j^T \hat{x}} \leq M$$

   .

Lemma 6.4. The feasible sets X in decision space and Y in objective space of a MOLP are convex and closed.

2.3.1.3 Weighted Sum Method.

Letting $\lambda \in \mathbb{R}^p_{\geq}$, the weighted sum linear program LP(λ) is

$$\min \lambda^T C x$$
$$\text{subject to } Ax = b$$
$$x \geq 0.$$

Theorem 6.6. Let $\hat{x} \in X$ be an optimal solution of the weighted sum LP.

1. If $\lambda \geq 0$ then $\hat{x}$ is weakly efficient.

2. If $\lambda > 0$ then $\hat{x}$ is efficient.

A few important observations regarding the MOLP are:

- A single nondominated point can be identified by many different weighting vectors λ.

- A single weighting vector λ can identify many nondominated points.

- The linearity of the constraints and objectives appears to make it possible to find all nondominated points with a finite number of weighting vectors, because X and Y are polyhedra.

In conclusion, we present **Theorem 6.11** from [35]. A feasible solution $x^0 \in X$ is an efficient solution of a MOLP if and only if there exists a $\lambda \in \mathbb{R}^p_{>}$ such that $\lambda^T C x^0 \leq \lambda^T C x$ for all $x \in X$. Ehrgott proves this using the optimization problem

of Benson's method and its dual. He concludes that every efficient solution in a multiobjective linear program is properly efficient ($X_E = X_{pE} \Rightarrow Y_N = Y_{pN}$) and we can find all efficient solutions by the weighted sum method.

2.3.2   Large Scale Optimization Solution Methods.

When solving real world problems, the number of variables is sometimes too large to handle with normal optimization algorithms. This led to the discovery of methods of decomposition as well as other ideas to break large problems into manageable subproblems, and subsequently solve them to create faster solution methods.

2.3.2.1   Bender's Decomposition.

J.F. Benders first developed his decomposition method in the early 60's [7], it was generalized in the early 70's by Geoffrion [43]. Lasdon's book on Large Scale Optimization [67] provides overviews of Bender's decomposition and other column generating techniques such as Dantzig-Wolfe Decomposition. Bender's decomposition exploits a special structure present in some mixed integer linear programs and makes it possible to solve large scale problems more efficiently. Most commonly, two blocks are present, representing variables x in a linear program and variables y in an integer or binary program. It's also feasible to have a linear and non-linear partition. By partitioning an otherwise intractable problem into two separate potentially tractable problems, we can iteratively solve the two partitioned problems to converge towards optimality of the intractable master problem. When partitioning, we are either solving for x or y separately, not simultaneously.

Bender's decomposition breaks the problem into two parts, the relaxed master problem P_r and the dual subproblem D_1. The relaxed master problem is solved first giving dual variable values that feed into the dual subproblem. The dual subproblem

is then solved to optimality and the dual variables are used to make Bender's cuts in the relaxed master problem. The relaxed master problem is essentially becoming less and less relaxed until optimality is reached.

Consider a problem P_x

$$\min v = cx + f(y)$$
$$\text{s.t.} \quad Ax \geq b - g(y)$$
$$x, y \geq 0$$

The Bender's decomposition algorithm is as follows:

Step 0 : Choose some feasible solution u^0 for the dual subproblem D_1. If no feasible solution exists, P_r doesn't have a feasible solution. Else, set $r = 1$ and proceed to Step 1.

Step 1 : Solve the relaxed master problem P_r

$$\min z$$
$$\text{s.t.} \quad z \geq f(y) + u^k(b - g(y)) \quad \text{for} \quad k = 0, 1, ..., r - 1$$
$$u, y \geq 0$$

Let z^r be the optimal objective function value with y^r the optimal solution. Set $\underline{z} = z^r$, go to step 2.

Step 2 : Solve the dual LP D_1 after substituting y^r for y

$$\max w = u(b - g(y^r))$$

$$\text{s.t.} \quad uA \leq c$$

$$u \geq 0.$$

Let u^r be the optimal solution. Then $\bar{z} = f(y^r) + u^r(b - g(y^r))$. Go to Step 3.

Step 3 : If $\underline{z} = \bar{z}$ then y^r is optimal and proceed to Step 4. Else, set r=r+1 and return to Step 1.

Step 4 : Let $y^r = y^*$ and solve the original problem P_x

$$\min v = cx + f(y^*)$$

$$\text{s.t.} \quad Ax \geq b - g(y^*)$$

$$x \geq 0$$

The optimal solution is labeled x^*. The overall optimal solution is (x^*, y^*).

This boils down to the following. Set some feasible solution to the initial Dual subproblem. Solve the relaxed master problem using this initial solution and obtain an optimal integer solution. Plug this optimal integer solution into the dual subproblem and solve to obtain an updated solution to the Dual subproblem. Continue this process until the two objective function values are equal. Once this occurs, plug the optimal integer solution into the original problem and solve, this gives the optimal linear variables. The optimal integer and linear variables are the optimal solution to the original problem.

2.3.3   Sensitivity Analysis.

Sensitivity analysis is an important part of decision making. Most mathematical programming problems are an estimation of reality, therefore knowledge about the sensitivity of the parameters is needed. Different methods exist for each type of mathematical programming problem, LP, IP, MILP, and NLP. LP sensitivity analysis is the most researched and more systematic in nature than IP's and NLP's. While systematic methods do exist for IP, MILP, and NLP programming problems, many times they're are handled on a case by case basis. In many cases, it's more practical to analyze sensitivity through intuition rather than using a systematic approach. See Fiacco's book [37] for non-linear sensitivity analysis methods.

2.3.3.1   Mixed Integer Linear and Integer Programming Sensitivity Analysis.

We are interested in changes to the objective function only. Changes in value function weights will not affect the constraint matrix or right-hand side, the network structure and its variables are another problem. The constraint matrix and right-hand side therefore remain fixed while the objective function is varied and changes in the optimal basis are observed.

The literature on MILP and IP sensitivity analysis is most prevalent in the 1970's and 80's, and is composed of a relatively small number authors. Applications and variations on the techniques are also scattered across journals outside the OR literature stream. There is also a strong mathematical basis in the mathematics literature. In this section, I focus on the OR literature stream to discuss the possible techniques, but I believe a survey paper would certainly be warranted in this topic area. With a foundational base in the operations research literature, less known techniques in the mathematics literature, applications and variations scattered throughout the

engineering literature, and difficult to implement procedures, a book or survey paper on the MILP and IP sensitivity analysis could be beneficial to the literature. I haven't seen any textbooks summarizing or explaining the methods. The techniques are uncommon in IP solvers as well.

IP and MILP sensitivity analysis is not as popular as LP sensitivity analysis due to the inherent difficulties present. For instance in LP problems, necessary and sufficient conditions ar available to prove optimality. In integer programming for the most part, some algorithm must be run to test for optimality.

General discussions on parametric analysis for MILP sensitivity analysis are Geoffrion's article in Management Science [44] and Jenkins [54]. In Table 8, I come up with a rudimentary classification scheme. This is just an initial stab at the themes I noticed, there are probably several others and many other important papers. The categories defined are branch and bound, cutting plane, duality theory, and other. Other includes various techniques such as inference based, binary decision diagram, heuristics, and specific methods based on the problem being solved. Most of these methods are mathematical in nature, although [64] introduces shadow prices for integer programming problems.

Branch and bound techniques for parametric programming assume a branch and bound algorithm is being used to solve the problem. The idea is to collect information as the algorithm is running in order to assist with the sensitivity analysis, or to carry out the sensitivity analysis during the running of the algorithms. Cutting plane methods utilize the same idea, collect information during the execution of an algorithm and use it for postoptimality analysis. Cutting plane methods begin by relaxing the integrality constraints of the programming program. New cuts are then added to the resulting LP, "cutting" up the feasible region to get closer the original integer optimal solution.

Table 8. Classification of Sensitivity Methods for Combinatorial Optimization

	IP	Mixed IP/Binary Linear	Binary 0-1
Branch and Bound	[93]	[2]	
Cutting Plane	[66] [9] [111]	[66] [51] [17]	
Duality Theory	[95] [64] [110]	[24]	
Implicit Enumeration		[86]	[87] [84]
Other		[29] [33] [53] [75]	[85]

2.4 Computational Complexity

This section provides terminology used throughout the dissertation, for a thorough review of complexity theory, see [42]. Computational complexity theory arose out of the need to classify problems based on solution time and tractability. Problems are classified into two classes, P and NP. A problem is said to intractable and in NP if it cannot be solved in polynomial time using a non-deterministic or deterministic Turing machine. A problem is said to be tractable and in P if it can be solved with an algorithm bounded by a polynomial in the size of its input or in polynomial time using a deterministic Turing machine. A problem is considered NP-complete if the problem is in NP and if every other problem in NP can be reduced to it. Karps paper is the first to define a number of NP-complete problems [58]. In order to show a problem is NP-complete, all that needs to shown is a polynomial transformation to another NP-complete problem. If NP-completeness is shown, a polynomial algorithm does not exist for the problem unless P=NP.

2.5 Transportation Mode Selection

Articles modeling intermodal routing are scarce [10]. Other papers exist that optimize regional to international transportation network daily decision making, we focus on the strategic decision though. Large companies or organizations such as Dow Chemical and the U.S. Air Force rarely make decisions on a daily or product by prod-

uct schedule. Rather, they lock in yearly rates with a carrier based on the estimated tonnage to be transported. Therefore, the strategic decision chooses "contract" carriers for long-term partnerships; thus the need to model schedules is negated. Look at [10] for one detailed model of intermodal transportation.

Perhaps the most comprehensive of the articles on measures in transportation mode selection is Cullinane and Toy's content analysis of mode choice [25]. The majority of studies utilize focus groups, interviews, and researcher hypotheses, this study utilizes a content analysis methodology to scientifically justify the attributes important in transportation mode selection. The factors and their descriptions are given in Figure 8.

Category constructs and their relationship to underlying attributes, variables and terms derived iteratively from the literature

Category name	Variables/Terms covered by category
Cost/Price/Rate	Cost, Price, Rate
Service (non-specified)	Service (non-specified)
Transit time reliability	Transit time reliability
Frequency	Frequency
Distance	Distance
Speed	Speed, Transit time, Terminal time, Transhipment time
Flexibility	Flexibility, Convenient schedule, Non-specific extras, Pick-up and delivery
Infrastructure availability	Infrastructure availability, Accessibility
Capability	Capability, Service availability, Equipment available, Capacity
Inventory	Inventory
Loss/Damage	Loss/Damage, Claims
Characteristics of the goods	Type, Value, Value/weight ratio, Volume, Weight, Density, Shipment size
Sales per year	Sales per year
Controllability/tracability	Controllability, Tracability
Previous experience	Non-specific positive behaviour, Relationship, Image of mode(s) used, Stability of firm

Figure 8. Mode Selection Factors from [25]

The literature is then searched using content analysis; rankings are shown in Figure 9.

Summary results of manifest analysis using the word or term enumeration measure

Category	Number of mentions	Percentage of total mentions	Rank
Cost/Price/Rate	1152	23.6	1
Service (unspecified)	366	7.5	5
Transit time reliability	538	11.0	3
Frequency	149	3.1	11
Distance	222	4.6	10
Speed	705	14.5	2
Flexibility	226	4.6	8
Infrastructure availability	146	3.0	12
Capability	232	4.8	7
Inventory	100	2.1	14
Loss/Damage	263	5.4	6
Characteristics of the goods	377	7.7	4
Sales per year	30	0.6	15
Controllability/tracability	146	3.0	12
Previous experience	223	4.6	9
Total	4875	100	

Figure 9. Mode Selection Factors Ranking from [25]

The five most influential factors are

- Cost/price/rate

- Speed

- Transit Time Reliability

- Characteristics of the Goods

- Service(unspecified)

Dobie's article on the core shipper concept looks at important attributes of transportation from both the carrier and shippers perspective [32]. Her analysis identifies common factors discussed in other intermodal transportation articles.

A comprehensive look of the transportation mode through a survey analysis is presented in [99]. Surveys are sent to three industry groups including low perishable items (canned fruits and vegetables), toiletries(perfumes, cosmetics, others), and electronics(radio and t.v. transmitting, signaling, and detection equipment). This captured groups ranging from low density/low volume/low value to high density/high

volume/high value items. Firms were chosen in part due to their closeness to large metropolitan areas and availability of a variety of transportation mode choices.

The most influential situations causing modal preferential changes as identified through the survey are:

- Desire to Improve Customer Service

- Deterioration of Service Provided by Mode

- Desire to Reduce Overall Distribution Costs

- Poor Pickup and Delivery by an Existing Mode

- Customer Complaints

- Desire to Reduce Transit Time

- Changing Needs of Customers

- Unsatisfactory Claims and/or Loss Experience

These give insight into the importance of carrier motivations and performance. A list of multiple selection criteria is also given, a sample of which we give here:

- Consistent, On-Time Pickup and Delivery

- Freight Charges

- Time-in-Transit

- Points Served by Mode, Including Routing Authority

- Frequency of Service

- Loss and/or Damage History

- Timely Acceptance of Shipments of All Sizes

- Door-to-Door Delivery

- Shipment Tracing Capability

- Prompt Claim Service

[105] uses a theory of reasoned action (TRA) methodology to determine important selection criteria. They claim the TRA model is useful in predicting behavior in choosing alternatives for transportation. Recent terrorist attacks have heightened the importance of 3 objectives in mode selection: reducing transport cost; increasing carrier preparedness in case of unforseen events; and increasing carrier security. Figure 10 compares the traditional method with the TRA method by mean importance.

Rank	Traditional Method (Rank Criterion Importance)†	Mean	TRA Method (Intention to Purchase)††	Mean
1	Rate Charged	26.19	Delivery Reliability	4.45
2	Delivery Reliability	20.94*	Rate Charged	3.84*
3	Carrier Reputation	11.67*	Equipment Availability	3.79
4	Equipment Availability	10.69	Response	3.68
5	Response	7.55*	Carrier Reputation	3.60
6	Billing Accuracy	6.91	Driver Quality	3.41*
7	Driver Quality	5.71	Billing Accuracy	3.37
8	Complaint Follow-up	5.31	Complaint Follow-up	3.28
9	Security	5.03*	Security	3.26

† Respondents allocate 100 points across all nine criteria with more points indicating greater importance.
†† Respondents indicated the extent to which they intended to purchase from a carrier that exhibits that criterion on a 1 to 5 (Very Unlikely to Very Likely) scale.
* Indicates criterion is significantly different from the one preceding at p<.05.

Figure 10. Mode Selection Factors Ranking from [105]

Witlox and Vandaele [109] use a stated preference approach to determine utility functions of the six most important factors in transportation mode selection. They discuss how the weight of each factor changes with the type of product being shipped and give weights for several categories. This differs from other articles where product characteristics are considered a decision factor [25]. The six factors discussed are

- Cost: the price of transportation, including loading and unloading.

- Time: the duration of transport, including loading and unloading.

- Loss and damage: the percentage of commercial value lost due to damage, theft and accidents.

- Frequency: number of services per week offered by the shipping company or forwarding agent.

- Reliability: percentage of the deliveries executed in time.

- Flexibility: percentage of unplanned shipments executed without excessive delay.

[8] also present a stated preferences approach. They only consider within Europe moves and model preferences with the UTA method of preference des-aggregation of Jacquet-Lagreze and Siskos (1978,1982). 25 alternatives are presented to multiple decision makers who state preferences for the alternatives. This allows inconsistency to be modeled. They indeed build an additive utility or "value" function of the six factors from [109], and allow for sensitivity analysis through an additional constraint. Relative weights of attributes are estimated for several product types including steel, textile, electronic, chemical, cement, packing, Pharmaceutical, and building material with different shipping modes (ie waterway, multimodal, road, rail).

A similar method to the idea presented in this research was developed in 2008 from a pair in Iran [76]. The authors define an AHP model for carrier selection, stating mode selection is irrelevant in this age. The AHP model is used in conjunction with a network optimization problem, with the LP formulation assigning the most goods to the highest valued carriers. In contrast to regular AHP, ratings rather than alternatives are given values, thus eliminating the need for re-weighting when new alternatives become available. The ratings fall under four subcategories defined through

factor analysis in a previous study, not given. 28 important factors in transportation carrier selection fall under 5 categories:

- Cost Considerations

- Insurance of Service Provision

- Customer Service

- Strategic Compatibility

- Handling Service.

Each of these are given a subjective rating with the exception of cost considerations. The model then minimizes a simple inventory calculation and transportation cost and maximizes value over the last four categories. The network model also guarantees a each node's demand is satisfied, total capacity on the carriers doesn't exceed the maximum, and limits the number of carriers on each route. An inconsistency rating is still calculated for the AHP model and satisfies Saaty's rule of thumb of less than .1, yet this still implies some inconsistency. VFT eliminates all inconsistency. A case study is provided comparing transportation carriers in IRAN, the majority being truck. A few water modes are explored, but multi-modal shifts are not examined. From their perspective on the literature, Figure 11 shows all factors considered in carrier selection. There are some errors in this paper. Most notable is the use calculation of the solution; it appears they failed to consider the average value through the system. Their algorithm and solution simply take the longest path through the network. This gives the highest value, but not the highest average value. A consequence is the algorithm could choose a series of bad arcs to get the highest value through the network. Our research is aiming to eliminate this by allowing the best average through the network.

Rank	Carrier selection criteria	Murphy (1996)	Liberatore (1995)	Korpela (1994)	Whyte (1993)	Abshire (1991)	Bagchi (1989)	Coulter (1989)
1	Cost	8	1	1	2		1	1
2	Physical facilities and equipments	2		1	8	11	1	1
3	Reliability of on time pick up and delivery	1		1	15	1		1
4	Financial stability	5		1	7	5	1	1
5	Quality of drives and employees	6	4	1			1	1
6	Flexible rates	11		1		7		1
7	Total transit time for the shipments	3				4		1
8	Reputation for integrity			1	4	8		1
9	Likelihood of establishing a long-term relationship		7	1	5			
10	Flexibility to future requirements	10		1	3	17		
11	Ease of claim settlement (loss or damage)	16				13	1	1
12	Billing invoice accuracy		6					1
13	Freight loss experience with the carrier	1				13		
14	Inventory carrying costs		4					
15	Handling expedited shipments	11				6		
16	Previous experience with the carrier				9	22		1
17	Carrier's personnel knowledge of shippers' needs	17		1	13	16		1
18	Carrier response in emergency situations				3			
19	Investment cost required to produce the inventory to fill the pipeline		3					
20	Shipment tracking and tracing capabilities	14	5			22		
21	Ability to provide service that do not damage goods while in transit	7	9			18		
22	Cargo capacity limitations		8					
23	Computer link		7	1		16		
24	Geographic coverage of carrier				6	9		
25	Customs clearance capabilities		10					
26	Carrier's cooperation with shipper (small shipments, routing requests,..)			1		11		
27	Administrative support				13			
28	Carrier ability to handle special products					28		

Figure 11. Carrier Selection Factors from [76]

In their chapter on Intermodal Transportation [23], Crainic and Kim summarize the transportation problem environment. They define intermodal transportation as "the transportation of a person or a load from its origin to its destination by a sequence of at least two transportation modes, the transfer from one mode to the next being performed at an intermodal terminal."

[10] models the intermodal transportation selection problem as a multiobjective multimodal multicommodity flow problem (MMMFP) with time windows and concave costs. He notes that "It is important to include multiple objectives such as minimization of travel time and of travel cost because shippers may have different concerns." We agree but argue there should be other considerations as well. He also states "Transportation mode schedules and delivery times must be included in the modeling of routing. Otherwise some located routes might be infeasible in a real world

47

situation. The existing schedules and demanded delivery times could be treated as time window constraints." However, the time window constraints are only imposed on the air segments. Finally, transportation cost should take into account economies of scale, more weight should be cheaper to ship per pound. The model formulated is NP-hard, thus a heuristic approach is proposed. The problem is decomposed into subproblems and solved in part by Lagrangian relaxation. The results of relaxing the time window constraints and cost functions shows the results of the new methods are significant.

2.5.1 Inventory Theoretic Approach.

While the articles above break down the transportation mode selection process into some number of measures, the inventory theoretic approach combines several factors into one equation. It accounts for the tradeoffs between cost, speed, reliability, and carrying costs (freight damage) by calculating a total logistics costs. Baumol and Vinod [6] originated the inventory theoretic approach in which they state "Since no pleasure is ordinarily derived from the means chosen for freight transportation, the selection of a carrier is likely to be based on economic considerations that are amenable to formal analysis." A company isn't concerned with measures as simple as transportation cost, speed, and reliability, but rather fundamental economic considerations are at play.

The Inventory Theoretic model variables are defined as follows:

A = order processing cost per order (\$)

R = annual demand (lbs)

EOQ = Economic order quantity (lbs) = $[\sqrt{\frac{2*A*R}{V*h_1}}]$

$V = product value (\$)$

48

s = reorder point = $k * \sigma_X + \mu_X$

h_1 = holding cost percentage for cycle and safety stock (% / \$ /yr)

h_2 = holding cost percentage for in-transit stock (% / \$ /yr)

k = safety factor = $norminv(PNS)$

F = freight cost (\$/lb) = $(K1 * e^{e_1 * \sigma_L}) * (e^{e_2 * \mu_L})$

μ_D = average daily demand = $\frac{R}{365}$

Cov_D = coefficient of demand variation

σ_D = standard deviation of demand = $Cov_D * \mu_D$

μ_X = mean demand during lead time = $\mu_D * \mu_L$

σ_X = standard deviation of demand during lead time = $\sqrt{\mu_D^2 * \sigma_L^2 + \mu_L * \sigma_D^2}$

μ_L = average lead time

σ_L = standard deviation of lead time

$\hat{\mu_L}$ = maximum lead time available

$\check{\mu_L}$ = minimum lead time available

$\hat{\sigma_L}$ = maximum standard deviation of lead time available

$\check{\sigma_L}$ = minimum standard deviation of lead time available.

The optimization problem thus becomes:

$$\text{Minimize } ETLC = [\tfrac{AR}{Q} + Vh_1(\tfrac{Q}{2} + k\sigma_x) + \mu_D\mu_L Vh_2 + FR]$$

$$\text{subject to } \mu_L \leq \hat{\mu_L}$$

$$\mu_L \geq \check{\mu_L}$$

$$\sigma_L \leq \hat{\sigma_L}$$

$$\sigma_L \geq \check{\sigma_L}$$

The inventory theoretic approach is criticized because it "does not address the constraining nature and qualitative considerations of freight transportation choice

[74]." If there existed a way to address these concerns however, this would be a much better approach. The authors in [74] also discuss the trade-off model but a reference isn't given.

Inventory costs in general arise due to the following factors:

- Warehouse rent/mortgage and utilities

- Inventory taxes and insurance

- Opportunity cost of money

- Inventory damage and theft

- Warehouse labor expenses

2.6 Conclusions

Several areas of operations research were covered including decision analysis, network optimization, computational complexity, inventory theory, and transportation mode selection. The next chapters tie each of these together in a unique way. The result is new theoretical contributions and unique applications.

III. A Value Focused Thinking Tutorial for Supply Chain Applications

Jeremy D. Jordan, Capt, USAF

Jeffery D. Weir, PhD

Dr. Doral E. Sandlin, Lt Col, USAF

Abstract

The supply chain is filled with applications for multi-criteria decision making and multi-objective decision analysis techniques. Value focused thinking (VFT) is one such technique that seeks to identify important aspects of a decision and lead the decision maker to the most valuable alternatives. This tutorial reveals the need for VFT in various areas of the supply chain decision making process and clearly guides the reader through the model-building process. The result is a methodology and a powerful decision tool for easy implementation in supply chain problems such as supplier selection, location selection, and carrier selection. An example of the methodology is given on a supplier selection problem.

3.1 Introduction

Multi-objective decision analysis (MODA) and multi-criteria decision making (MCDM) are very popular decision support tools in the logistics decision environment. Rarely is a logistical decision based on a single objective, multiple objectives are always competing with one another suggesting that quantitative and qualitative methodologies should be utilized ([71]). Understanding this, many researchers have used the Analytic Hierarchy Process (AHP) to model logistic decisions such as: Supplier Selection - [101], [45], [107]; Facility Location - [5], [14], [112]; Carrier Selection - [69]; . A number of other multicriteria methods have been applied to supplier selection as well [34], [92], and [63]. One method not used in the literature is Value Focused Thinking (VFT) [61]. VFT has its roots in Decision Analysis, an Operations Research methodology, and although widely used and acclaimed as a suitable method for use in supply

chain decisions [30], VFT has yet to be applied to supplier selection, carrier selection, or the facility location problem. This could be due to lack of familiarity with the method and its many benefits among logistics practitioners. In this manuscript, we seek to introduce VFT methodology, discuss VFT advantages and disadvantages, and show its application on a supplier selection case study.

The area of supply chain management clearly presents a decision maker with multiple competing objectives and alternatives making multiple objective decision analysis an ideal tool for decision making. The literature however reveals a missing link between value modeling and supply chain management. While several papers utilize techniques of MCDM, including Analytic Hierarchy Process, risk models, etc, papers using VFT are absent from the logistics literature stream. Of particular interest to supply chain decision makers should be the values of their decisions, and VFT provides this. The majority of analysis in decision making is done through a comparison of alternatives. [61] calls this alternative-focused thinking. Decisions are made based solely on the alternatives perceived to be available. Value-focused thinking is different in that it seeks to solicit important aspects of a decision and then quantify the alternatives based on this information. This allows the decision maker to obtain the actual value of his/her decision as well as generate new alternatives that may yield higher values.

Several differences exist between AHP and VFT although the two methodologies are sometimes thought to be similar. In terms of time, AHP can be more time consuming than VFT, in that pairwise comparisons for each of the measures and alternatives are required, for n options this means $\frac{n(n-1)}{2}$ comparisons. In a small case where 5 alternatives and 10 measures exist, this amounts to 10 and 45, respectively. Additionally, each time new alternatives are introduced, new pairwise comparisons are required and a new ranking occurs. Conversely in VFT, once a value hierarchy has

been defined, ANY alternative can be valued. A one-time commitment by the decision maker results in a reusable product for multiple future decisions. A single value model allows the decision maker to value an essentially infinite number of possible alternatives and understand the overall value each of those alternatives provides. This is especially important in supplier selection where a company is selling multiple products, a tool is needed to rapidly value multiple alternatives for multiple products without the need to return to the decision maker each time new alternatives are available.

VFT measures the true value of the decision while AHP makes the best decision with the available alternatives. In certain applications, the best available decision may not be a good choice. VFT reveals the value of each alterative, and shows the inferiority of an alternative if it is a bad choice. Although VFT has been used in a number of strategic decision application areas, it has yet to be used in logistics applications. A Google scholar search of "supplier selection" within Saaty's [90] ground breaking article in Management Science produces 435 hits, in [89] and [91], 53 and 88 respectively. AHP is used extensively in suppler selection. VFT however, although comparable and even a stronger methodology in some areas, yields 11 hits when "supplier selection" is searched for within [61], none of which actually use the methodology to model supplier selection. Similar results occurred with searches for "facility location" and other logistics problems. Why the large discrepancy? They are similar methodologies, both suited to handle the complexities of a multicriteria decision problem, each with strengths and weaknesses based on the application. This could be because VFT is being taught only at select schools, whereas AHP has a much broader audience.

Although determined as a suitable OR-method for supplier selection in [30], VFT has yet to be used in any part of the supplier selection decision process. It could be

that logisticians haven't been introduced to the method; this is the purpose of this tutorial.

3.2 Decision Analysis and Value Focused Thinking in Logistics

Here we review foundational Value Focused Thinking literature and show its immediate applicability in the supply chain. The majority of the VFT methodology throughout the paper is gathered from the most referenced textbook on the subject, [61], as well as [65].

Decision analysis [16] is a powerful and widely used technique in Operations Research ([52],[19]), but recently has been defined further [59] as "a set of quantitative methods for analyzing decisions based on the axioms of consistent choice. This excludes techniques such as AHP, fuzzy sets, MCDM, traditional math programming, and other useful decision making techniques. Value-Focused Thinking and Decision analysis seek instead to aid in human decision making, not model the human decision making process. The justification for this purpose is decision makers should desire to make rational choices given any situation. Sometimes it may be a poor choice to model inconsistent or irrational behavior. A better approach if this is true is to build models of rational choice and let the decision maker utilize the models in their decision making process. Thus, removing the techniques above and their applications in logistics leaves few true decision analysis studies in the logistics literature. Yet, most would agree that all logistics decisions hold to the following axioms of consistent choice from [65] where $\succ$ means some consequence c is preferred:

1. (Transitivity) If $c_i \succ c_j$ and $c_j \succ c_k$, then $c_i \succ c_k$

2. (Reduction) If the rules of probability can be used to show two alternatives have the same probability for each c_i, then the two alternatives are equally preferred.

3. (Continuity) If $c_i \succ c_j \succ c_k$, then a p exists such that an alternative with a probability p of yielding c_i and a probability of 1-p of yielding c_k is equally preferred to c_j.

4. (Substitution) If two consequences are equally preferred, one can be substituted for the other in any decision without changing the preference ordering of alternatives.

5. (Monotonicity) For two alternatives that each yield either c_i or c_j, where $c_i \succ c_j$, the first alternative is preferred to the second if it has a higher probability of yielding c_i.

These axioms clearly hold for logistics decisions, it can be argued they hold for any decision. If its agreed these the axioms of consistent choice hold for logistics decisions, why is VFT not being used? AHP allows for inconsistency in its decisions, we argue this should not be allowed in supplier selection or for any other strategic logistics decision. Companies should care about making rational decisions.

Alternative focused thinking techniques, such as AHP, encourage a "best" choice among the available alternatives; value focused thinking begins with the fundamental inputs in a decision and reveals what is truly valued. Rather than starting with alternatives, VFT starts with objectives and measures. Alternatives can then be generated from these measures and assigned values based on their fulfillment of the objectives. In this way, "value gaps" between a best available alternative and an ideal alternative can be identified, providing a decision maker with a more complete analysis of the problem at hand.

Other important benefits of VFT exist as well and are shown in Figure **??** from [61], three of which are uncovering hidden objectives, creating alternatives, and improving communication; these don't commonly come to mind in most decision support

studies. The pervasiveness of multiple alternatives lures decision makers away from thinking about fundamental objectives, and traps them in paradigmatic thought processes. By improving communication through the VFT modeling process, new ideas emerge, objectives are uncovered, and alternatives are generated.
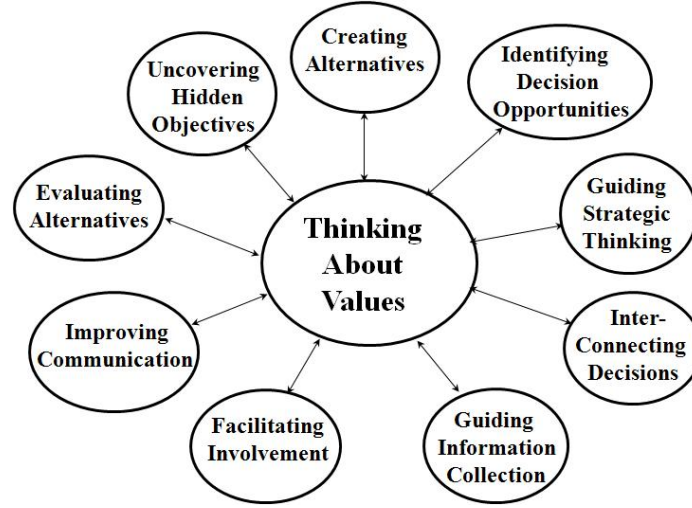
Figure 12. FIGURE: VFT BENEFITS

As with any methodology, a list of suggested implementation procedures is useful. [65] and [61] describe the steps in the VFT process rather generally, a more specific declaration of events is given by [96] in Figure 13. This detailed process shows 10 essential steps in a value model study. Problem identification (Step 1) is self explanatory, ie. supplier selection, transportation mode selection, facility location, etc. Creating the value hierarchy (Step 2), developing measures (Step 3), creating value function (Step 4), and weighting the hierarchy (Step 5) will be discussed in Sections 3.3 and 3.4. Following model construction is the generation (Step 6) and scoring (Step 7) of alternatives through deterministic analysis (Step 8). Because of the subjectiveness in defining values and weights, a proper sensitivity analysis (Step 9) is essential for a good analysis. Finally, conclusions and recommendations (Step 10) are to help communicate the results to decision makers.
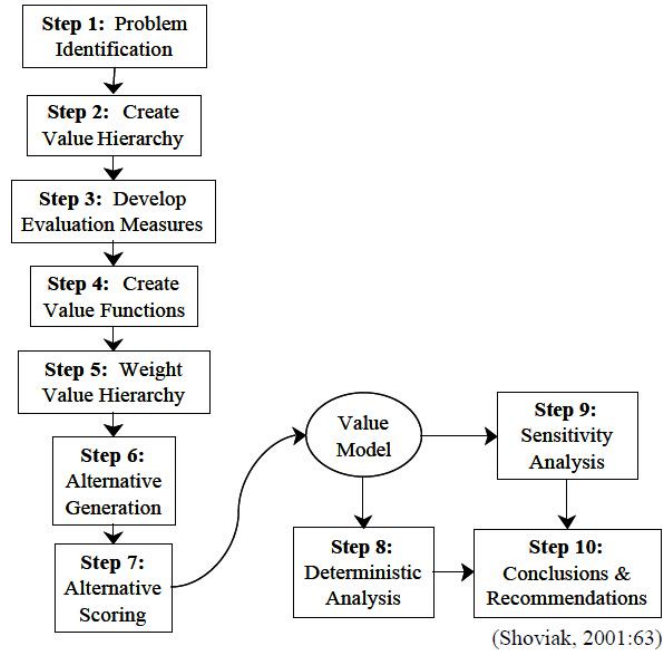
Figure 13. Steps in VFT Study

3.3 Building the Model Framework

This section describes proper identification and structure of a decision's objectives and measures, this being the value hierarchy. Two approaches are possible when building a hierarchy, bottom-up and top-down. When alternatives to a decision problem are previously known, a bottom-up method is used to establish ways they are different. The important ways in which alternatives differ end up being the measures used to compare them. This "bottom up" approach is called such because the value hierarchy is built from the measures up to the objectives. Conversely, a top-down approach begins with the fundamental objectives and works down to the measures. Finally, properties and uses of a good value hierarchy are discussed.

3.3.1 Deciding on Measures and Objectives.

Determining good objectives is critical to the accuracy of the model. Initially, the fundamental objectives of the problem must be identified. For instance, in supplier selection, the overall objective of one organization may be to minimize costs while another seeks to maximize customer satisfaction. These differing views on the purpose of supplier selection may affect the weighting and value function discussed in the next section. Always ask the decision maker why he feels that an objective is important and what is trying to be accomplished through that objective. Doing this ensures the decision maker thinks through the problem completely, and drills down to the actual overall objective. Attainment of the fundamental objective is achieved through further objectives called means objectives, while achievement of the means objectives is gauged through measures. Measures themselves are generally quite easy to generate; the difficulty comes in deciding which measures should be included in the value hierarchy. Including every possible measure ensures completeness but increases difficulty in weighting. Less measures are preferred given they adequately represent the decision problem. When deciding what measures to include, use the "test of importance" [60], evaluations should only be included if for two given alternatives, a change in the measure could change the preference between the two.

Types of measures are natural or constructed, and direct or proxy. Natural scales need not be produced, that is they are "naturally" occurring. Examples include cost in dollars, container loads shipped, and time after due date. A constructed scale on the other hand does not exist and must be developed for a measure if a natural scale isn't available or practical. Natural and constructed scales can be either direct or proxy. Direct scales measure the direct attainment of an objective whereas proxy scales measure the degree of attainment of an objective. Natural scales are clearly objective in that a clairvoyant, able to see the future, will score an alternative

identically, now and in the future, unless the levels of that measure change with time. Constructed scales are different. There needs to be a test for clairvoyance when setting levels on these scales, allowing subjectivity into the model introduces noise. For instance, when rating a supplier, it's best to clearly define the difference between a high and medium score for a measure such as responsiveness. Simply allowing an individual scorer to assign a value of high or medium based on personal preferences creates a poor model, taking away from the advantages of using VFT.

Metrics or measures used for many logistical decisions can easily be gathered from the logistics literature. The true objectives however, are not as easily defined. These objectives can be obtained through specifying the reasons the measures are used to compare alternatives. Why is each measure important? Is this measuring some level of achievement of the objective? Once these questions are answered, the quality of the hierarchy can be weighed against the proven properties of a good hierarchy.

A general hierarchy is provided in Figure 14 showing the breakdown of objectives, measures, tiers and branches of the hierarchy, and the global and local weights discussed in Section 3.4.2.
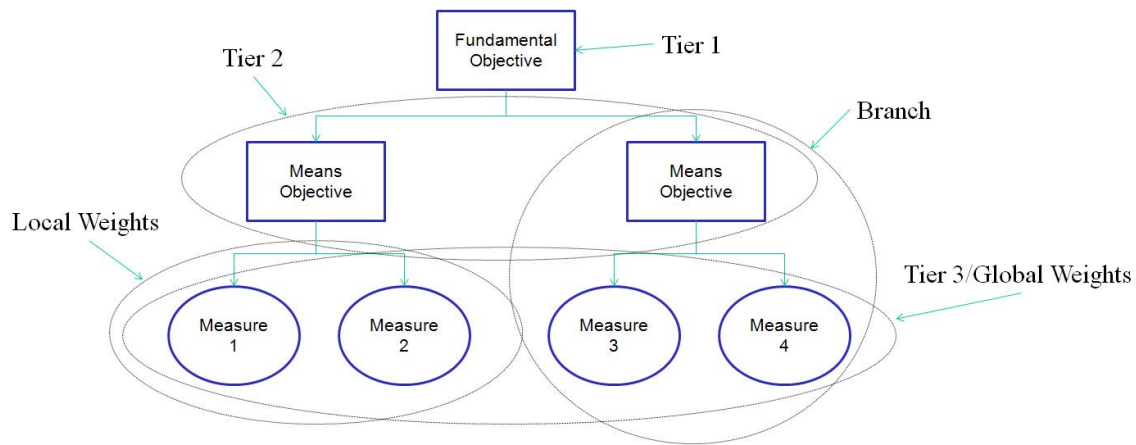
Figure 14. General Value Hierarchy

3.3.2 Properties of a Good Value Hierarchy.

Keeney [61] defines the properties of a good value hierarchy as completeness, non-redundancy, independence, operability, and small size. Completeness ensures that every important objective and measure valued by the decision maker is accounted for. A good hierarchy should have mutually exclusive or non-redundant measures; large overlaps between measures are not preferred. This is inevitable in some cases however. If measures must overlap, its important the decision maker is aware of the implications on the model, this being the combined weights of the measures in the overall decision. In addition to non-redundancy and completeness, a value hierarchy should be operable or understandable by all interested parties and small in design. Another property is small size. These may seem non-intuitive as people normally enjoy building complex models, even though model sparsity usually results in a better solution. Smaller hierarchies are easier to communicate and usually have enough information to make good decisions. The art of value modeling lies in the ability to balance the defensibility and practicality of a model. The final property, independence, is covered in Section 3.4.5.

3.3.3 Using the Value Hierarchy.

The primary use of a value hierarchy is to evaluate alternatives, assigning each alternative some value based on the objectives and measures being defined, and taking into consideration the weight each objective contributes to the overall decision. In addition to assigning values to alternatives, value hierarchies can actually help generate alternatives. This makes it easy to identify value gaps, or differences between the best available alternative and the best possible alternative in a perfect world. Finally, the value hierarchy helps to facilitate communications between decision makers and initiate data collection. Gathering each of the interested parties into a room for a

discussion of objectives is unquestionably beneficial and certainly data will result. To carry out the former two uses, evaluating and generating alternatives, values and levels need to be assigned to the measures and weights to the measures/objectives, this being the topic of the next section.

3.4   The Additive Value Model

Development of a good value hierarchy is followed by assigning values to the measure scale and weighting the objectives and measures. Several assumptions are necessary to use an additive value model, these are discussed in this section. If these assumptions are not realistic, other models such as multiplicative models are viable as well [60].

3.4.1   Assigning Values to Measures.

Eliciting values from a decision maker can be complicated. Single dimension value functions (SDVF) are normally used to capture preferences for varying levels of the measures, assigning each level of the function a value between 0 to 1 . These functions can be monotonically increasing or decreasing. For instance, performance is a monotonically increasing function because higher values are more desirable, whereas cost is monotonically decreasing, because higher levels are less desirable. This section covers two popular SDVF's [65], piecewise linear and exponential. For an in depth look at values and preference functions see Keeney and Raiffa [60]. Initially, the modeler needs to find the high and low values of a measure, high being the most preferred level and low the least preferred. For instance, in supplier selection, say a defective product rate of say 1% is desired. Anything above 1% is unacceptable, and thus will have no value, but anything less than say .5% is superfluous and achieves all the desired value possible for that measure. The high level that achieves all value

is assigned a 1 and the low level a 0. All other values are derived from these high and low levels. If a defective product rate of .75% achieves half the value for a measure, a linear monotonically decreasing function is fit from 0 to 1 on the y axis with an x axis of 1% and .5%, respectively. However, this is not always the case. If .75% achieved a quarter of the value for a measure, a non-linear monotonically decreasing function must be used. Continuous measures are captured by either the piecewise linear or exponential functions, preferable the exponential, depending on the preferences of the decision makers. For cases where measures are not continuous, or preferences do not match the exponential function, [65] suggests using the piecewise linear SDVF.

3.4.1.1   Piecewise Linear Single Dimension Value Functions.

When measure levels are discrete, a piecewise linear SDVF is best. Values are assigned to different levels of the measure. In a supplier selection, it may be that a cost of $20 per widget has a value of .75, while $25 has a value of .25 and $22 is valued at .7. This is easily captured through a piecewise linear SDVF. As another example, take a qualitative rating of supplier responsiveness between 1 and 5. Each of these 5 levels are assigned a value, 1 receiving a 0 and 5 a 1. See Figure 16 for an example with of 5 levels and values. Setting the scales up is simple as well using three steps:

1. Place each value increment in order - smallest to largest value increment

2. Scale each value increment as a multiple of the smallest value increment (eg. 2:1 or 10:1 or 4:1)

3. Sum the value increments to one and solve for the smallest value increment

For an example application, see the supplier selection example in Section 3.5.

Figure 15. Exponential Value Function
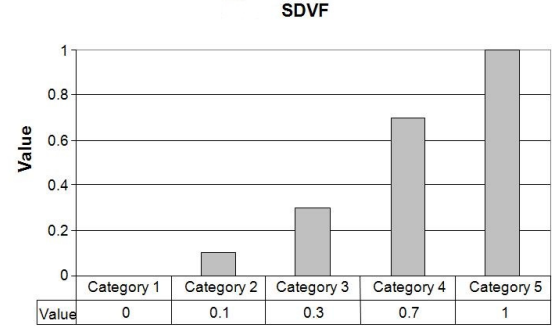
Figure 16. Piecewise Linear Value Function

3.4.1.2 Exponential Single Dimension Value Functions.

Many of the measures we encounter are continuous in nature. The exponential SDVF is equipped to handle preferences on a continuous scale and is easy to explain to a decision maker. Initially, as with the piecewise linear SDVF, high and low levels need definition. For monotonically increasing measures, the continuous exponential function is

$$v_i(x_i) = \begin{cases} \frac{1-e^{-(x_i-x_L)/\rho}}{1-e^{-(x_H-x_L)/\rho}} & \rho \neq \text{Infinity}; \\ \frac{x_i-x_L}{x_H-x_L}, & \text{otherwise.} \end{cases} \tag{11}$$

where x_H is the most preferred level (assigned a value of 1), x_L is the least preferred level (assigned a value of 0), x_i is the level of the ith measure, ρ is an unknown parameter, and $v_i(x_i)$ is the value of the ith measure at level x_i. With this equation, any continuous measure is valued, however the equation must be solved for the unknown parameter ρ. Unfortunately, no closed form solution exists. In practice, ρ can be estimated if the decision maker can assign a value to a mid-level between x_H and x_L. Take a measure such as defects per million parts and let x_L=500 and x_H=1500, having values 0 and 1 respectively. A mid level of 1000 may have a value of .7, or .3 say. Using .7, one method is to plug all numbers into Equation 11,

$$.7 = \frac{1 - e^{-(1000-500)/\rho}}{1 - e^{-(1500-500)/\rho}}$$

and solve using Excel Solver for ρ, which turns out to be 590 (note:seems a little high). With a solution for ρ, any level of measure x_i can be valued. Figure 15 shows the shape of the exponential value function for differing levels of ρ. If the user is not familiar with Excel, an alternative method to find ρ exists, see p. 68 in [65]. Next, weights must be assigned to each of the measures.

3.4.2 Determining Weights.

There are several means to determining weights, AHP, swing weights, direct assessment, or group weights to name a few. Note, AHP can be used to determine weights for a value hierarchy but this differs from using it to evaluate alternatives. An actual AHP model requires the decision maker to make pairwise comparisons across all combinations of the possible alternatives, whereas VFT provides a value for each of any large number of alternatives. The AHP technique in this case is used only for weights. Measures can weighted globally or locally; the former compares all measures simultaneously, the latter looks at each measure in the context of its means objectives. Local weighting is preferred as it tends to be easier for decision makers to weight measures within objectives rather than across all objectives. In turn, this provides more accurate weights, and global weights are easily calculated from local weights. Either way, the global weights are needed to calculate the overall value function. w_i is the notation used for the weight of measure i.

An easy method to implement for determining weights is direct assessment. This is accomplished through examining the measures from one means objective and weighing tradeoffs between them. see hierarchy in (generic hierarchy). The least important measure is assigned a 1, the remainder of the measures are assigned numbers based

65

on how much more important they are than the least important measure. A measure that is twice as important as the least important measure is given a value of 2, call it r_i for measure i, and so on. Since the weights must sum to one, that is $\sum w_i = 1$, these values must be scaled to a decimal between 0 and 1. To determine a weight w_i for measure i, take its successive r_i and divide by the sum of all the ranks, that is

$$w_i = \frac{r_i}{\sum r_i}.$$ (12)

This works well for most cases, if consensus isn't reached, a more robust method should be used, either swing weights or AHP.

3.4.2.1 Swing Weights Method.

Here we show the standard swing weights method for determining decision maker weights. Advanced swing weight methods have been developed as well but are not covered here, see for instance Parnell's method [83] which uses a matrix like that in Figure 18. The standard method begins by building a table like that in Figure 17, where each of the measures 1 through n are set at high to create a list of n hypothetical alternatives. Where one measure is set to high, the rest of the measures are set to their lowest level. Next, each of these alternatives are ranked from 1 to n, with 1 representing the best alternative and n the worst case.

Following the ranking assignments, each measure is assigned a rate. The baseline, or worst alternative, is assigned a rate of 0 and the highest ranking measure is assigned a weight of 100. The decision maker is then asking the following questions: how much less satisfaction do you get from swinging a lower ranked measure versus swinging the highest ranked measure? If swinging the top ranked measure from low to high gives 100% satisfaction, what percentage satisfaction do you get from swinging the lower ranked measure?. After assignment of all rates, the weight is calculated as a ratio of

$$\frac{Rate_i}{\sum_{i=1}^{n} Rate_i}. \tag{13}$$

Attribute Swung from Worst to Best	Consequence to compare	Rank	*Rate*	Weight
(Benchmark)	All attributes at X$_L$	n	0	
Measure 1	(x^1_H, x^2_L, ...,x^n_L)			
...	...			
Measure n	(x^1_L, x^2_L, ...,x^n_H)			

		Level of Importance of Value Measure			
		Extremely Important	Very Important	Important	Less Important
Variation in Measure Range	Very High				
	High				
	Medium				
	Low				

Figure 17. Swing Weights Standard Method

Figure 18. Swing Weights Matrix Method

The AHP method of determining weights is not covered here, as applications are plentiful for logistics. Application is similar to the methods used in a popular AHP supplier selection application [45], however the AHP is NOT used to rank alternatives. For an extensive list of application papers see [50]. With weights and values for each of the measures defined, we can calculate the overall value of each of the alternatives.

3.4.3 Mathematical Formulation of the Model and Scoring.

The value of an alternative is given by inserting levels of the weights w_i and the values of the levels of each measure x_i, or $v_i(x_i)$. Doing this for each alternative creates the additive value model:

$$v_j(x) = \sum w_i v_i(x_{ij}) \tag{14}$$

for $i = 1, 2, ..., n$ measures and $j = 1, 2, ..., m$ alternatives, where $v_i(x_{ij})$ is the single dimension value function of measure i for alternative j, w_i is the weight of

67

measure i, and $v_j(x)$ is the multiobjective value for alternative j.

After inserting all inputs into the equation above, or performing Deterministic Analysis as referred to in Figure 13, an overall score is obtained for each alternative being compared. The meaning of the final score is straightforward, it's the amount of value the alternative provides as a solution to the problem. If the final value is say .674, the alternative provides 67.4% of the total possible value that could be achieved when making this decision, implying a "value gap" of 32.6% exists between the perfect alternative and the current alternative. Since most of the process to create this value is subjective, the analysis is not complete.

3.4.4 Sensitivity Analysis.

Because of the subjectivity of the VFT modeling process, a requisite sensitivity analysis is needed to reveal the effects of changes in value functions and weighting schemes, primarily weighting schemes as these produce greater changes. The idea is to vary the weights w_i of Equation 14 for each evaluation measure and determine at what point changes in the most valued alternative occur. Several techniques are currently used for sensitivity analysis including math programming [79], algorithms [11], and simulation [4], [55], [73]. Historically, sensitivity analysis has been limited to one-way analysis, changing only one weight at a time [15]. However, new techniques are emerging (TRIAGE Method, COSA method, VBR Method) that advance the ability to conduct sensitivity analysis.

Sensitivity analysis is generally performed through manipulation of the weights one at a time (COSA). Results of such manipulations are best viewed on a "break even" chart. The purpose of the chart is to visualize the effects of changes in the weight of a measure on the preferred alternative. For instance, changing a weight from .3 to .35 could affect which alternative is valued highest, and thus have an effect on the

overall solution. Some view this as a weakness of VFT, however a befitting sensitivity analysis alleviates such concerns. An illustrative example is given in Section 3.5.

3.4.5   Assumptions in using an Additive Value Model.

Use of an additive value model requires the satisfaction of several mathematical assumptions. If the assumptions fail, other models such as multiplicative or multilinear models may be used. We only discuss 3 of the important mathematical requirements of the additive value model in this section, more info can be found in [65].

Definition 4 *A function $v(x)$ is a value function if $v(x') > v(x'')$ if and only if $x' \succ x''$, where x' and x'' are specified but arbitrary levels of x.*

In order to use this additive value model, measures must be preferentially independent. We first discuss the corresponding tradeoffs condition that must hold when dealing with two measures. Two measures X and Y hold to the corresponding tradeoffs condition if: for any levels x_1, x_2, y_1, and y_2 of the measures, if $(x_1, y_1) \sim (x_1 - a, y_1 + b)$ and $(x_2, y_1) \sim (x_2 - d, y_1 + b)$, then for c such that $(x_1, y_2) \sim (x_1 - a, y_2 + c)$ it is true that $(x_2, y_2) \sim (x_2 - d, y_2 + c)$. This condition must hold for any x_1, x_2, y_1, and y_2. Thankfully, for 3 or more measures, this need not be shown, only mutual preferential independence is needed.

Definition 5 Preferential independence: *Suppose that Y and Z are a partition of $X_1, X_2, ..., X_n$, each X_i being in exactly one of Y or Z. Then Y is preferentially independent of Z is the rank ordering of alternatives that have common levels for all attributes in Z does not depend on these common levels. (The common levels do not have to be the same for different attributes, but the level of each X_i in Z is the same for all alternatives.)*

Given this, we can now define mutual preferential independence.

Definition 6 Mutual Preferential independence: *A set of attributes $X_1, X_2, ..., X_n$ displays mutual preferential independence if Y is preferentially independent of Z for every partition Y, Z of $X_1, X_2, ..., X_n$.*

For proofs, see [65].

## 3.5 Supply Chain Application Example

Here we apply the methodology presented in Sections 3.3 and 3.4 to a common logistics problem, supplier selection. A decision maker can rapidly apply value focused thinking to any logistics problem, this section gives a practical example. The following problem is adapted from a case study [80]. A bottom-up approach to building the hierarchy is taken as the measures have already been defined.

3.5.1 Measures and Objectives.

Measures for the majority of well studied supply chain decision problems are likely defined in the literature already. There may be some variation due to a company/decision maker's fundamental objectives for operating the business, but for the most part these problems are well studied. Measures from the case study [80] are given in Figure 19, these are consistent with the literature [45], [101], [92]. Using a bottom up approach, we assume the fundamental objective of the decision maker, shown in the 1st and 2nd tier, is to maximize profit and customer satisfaction by choosing the best supplier. Certain decision makers may be prone to valuing profit more than customer satisfaction and vice versa. In this case, the hierarchy can be restructured to reflect these personal objectives, which likely affects the weights of the measures. Another option is to build every possible objective and measure into the hierarchy and weight the measures and objectives with small or no effects a zero. Alternatively, if a top down approach is used, the decision maker determines fundamental objectives, the

70

means objectives to achieve these, and finally the measures important to measuring achievement of the objectives. If a general framework for a problem is desired, its best to include all measures a decision maker may consider.

Means objectives, given in the 2nd and 3rd tier, achieve the fundamental objectives (1st tier) and are needed to establish the measures in Figure 19, the 4th tier. These measures are used to assess the realization of the means objectives. Each measure from the figure is calculated using data from the case study [80]; for data and formulas see Appendix B. Also, let the measures be numbered from left to right, so $i = 1 = $ Cost, $i = 2 = $ Quality, and so on.
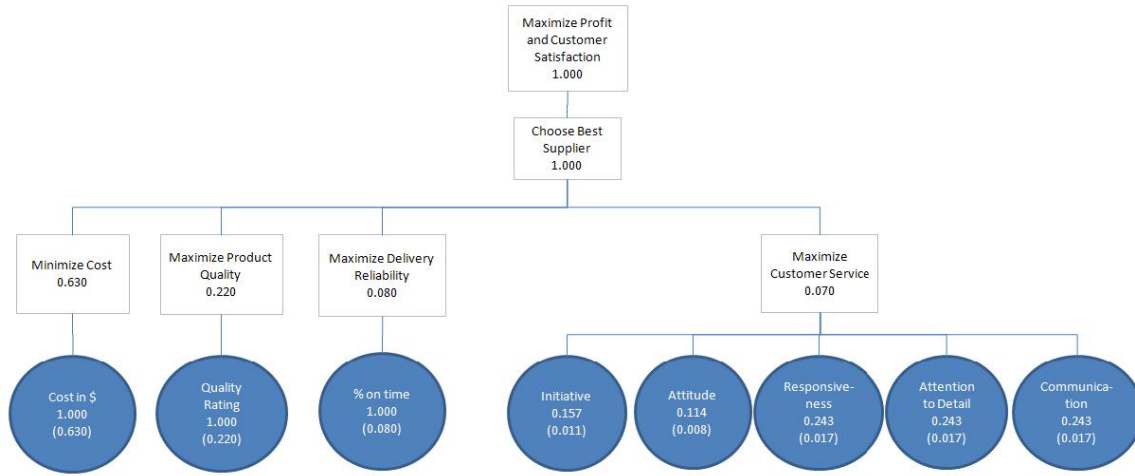
Figure 19. Supplier Selection Hierarchy

Single dimension value functions must next be assigned to each of the measures in Figure 19.

3.5.2 Assigning Values to Measures.

Assigning values to the measures, as shown in Section 3.4.1, is relatively straightforward. Here we show an application of the piecewise linear single dimensional value function on quality rating and the exponential single dimensional value function on

delivery rating. Applying the piecewise linear SDVF to the quality rating from Figure 19 is simple, the x axis is already defined. Simple elicitation from the decision maker can be used to assign values, the notional values for different levels of quality rating are shown in Figure 20. Alternative E from Appendix B has a quality rating of 1423 meaning $v_2(x_{25}) = .62$. For calculations of quality rating, delivery rating and the other measures, see Appendix B.

Figure 20. Quality Rating Piecewise Linear Single Dimensional Value Function
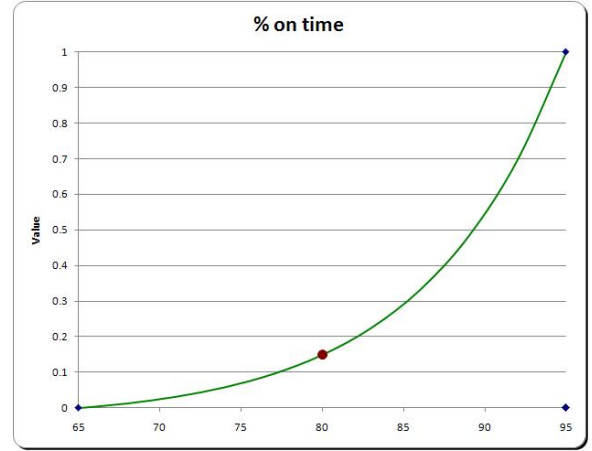
Figure 21. Delivery Rating (% on time) Exponential Single Dimensional Value Function

Values for delivery rating can be captured on a continuous scale so the exponential function is used.

3.5.3 Determining Weights.

Local weights for each of the measures are given in Figure 19, global weights are in parenthesis. The notional weights for the means objectives in Tier 3 are taken from a popular AHP article [45] and adapted slightly to fit the case study hierarchy. From here, local weighting using direct assessment is used to weight the measures in Tier 4. Since AHP has been used on several logistics problems, the weights in most

cases can be determined through direct assessment. If for some reason weights are not available, or the decision maker believes the weights gathered by direct assessment aren't representative, weights could be determined using AHP or swing weights.

What if weights for the means objectives in Tier 3 are not readily available from the literature or the corporation feels differently than the general consensus? Using direct assessment, let's assume the least important objective is maximizing customer service, and the objectives are numbered left to right, making $r_4 = 1$. If maximizing delivery reliability is 10% more important than maximizing customer service, $r_3 = 1.1$. Similarly, if maximizing product quality and minimizing cost are 2 and 4 times more important than maximizing customer service, $r_2 = 2$ and $r_1 = 4$. Thus from Equation 12,

$$w_1 = \frac{r_1}{\sum r_i} = \frac{4}{8.1} = .49,$$

$$w_2 = \frac{r_2}{\sum r_i} = \frac{2}{8.1} = .25,$$

$$w_3 = \frac{r_3}{\sum r_i} = \frac{1.1}{8.1} = .14,$$

and

$$w_4 = \frac{r_4}{\sum r_i} = \frac{1}{8.1} = .12.$$

Note: these weights were only used for purposes of demonstrating direct assessment and for the remainder of the paper, w_i refers to the ith measure, not means objective.

Alternatively, the methods in Section 3.4.2.1 may be used if direct assessment in unsuccessful.

3.5.4 Mathematical Formulation of the Model.

Here we show an example calculation of the value of one measure, quality rating, for one alternative, E. We know $v_2(x_{25})=.62$ and $w_2=.22$ for measure 2 and alternative

5(E). The value added to alternative 5 by quality rating is thus

$$w_i v_i(x_{ij}) = w_2 v_2(x_{25}) = .22 * .62 = .1364.$$

Doing this for each measure for alternative 5 gives

$$v_5(x) = \sum w_i v_i(x_{i5}) = .669$$

for $i = 1, ..., 8$, see Figure 22 for overall values.

3.5.5 Alternative Comparison and Sensitivity Analysis.

After calculating $v_j(x)$ for each of the $j = 1, ..., 5$ alternatives in Appendix B, we compare differences in values in Figure 22. The amount of value each measure contributes to the overall value is color coded. For example, alternative D, the lowest valued alternative, receives the majority of its value from cost but no value from quality. For alternative B, on the other hand, quality contributes more to the overall value than does cost, even though quality is weighted $1/3$ the importance of cost. Alternative E is the top valued alternative for the specified weights w_i and values $v_i(x_i)$. Since these values and weights are again subjective, the effects of varying them should be explored through sensitivity analysis.
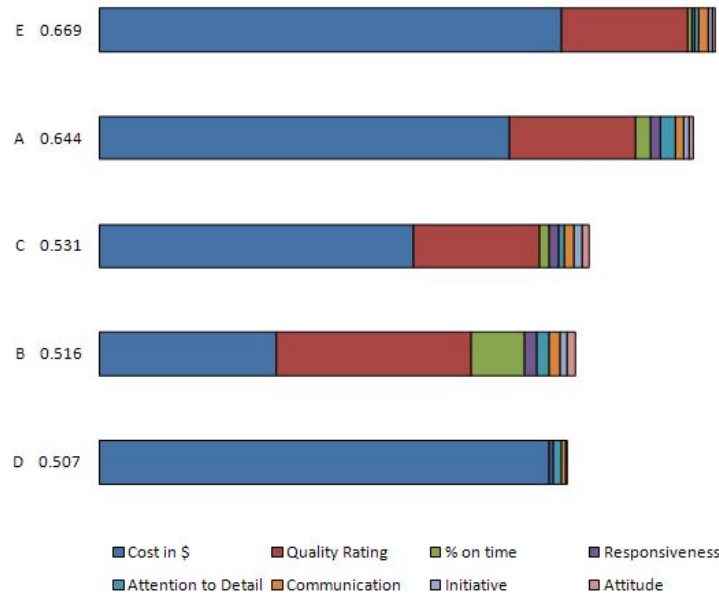
Figure 22. Analysis of Alternatives

Sensitivity analysis can be accomplished in many ways, the global proportional method is used here to view the impacts of varying the global weights of cost and quality rating on the preferred alternative. This method varies the global weights between 0 and 1. From Figure 23, a change in the weight of cost from .63 to anything $\leq$.45 changes the preferred alternative from E to B. To put this in practical terms, if a decision maker is somewhat uncertain of the weight of cost, he should consider both alternatives E and B when making a final decision. For quality rating, if the decision maker agrees the weight is .20$\leq w_i \leq$1, both alternatives E and B should be examined.
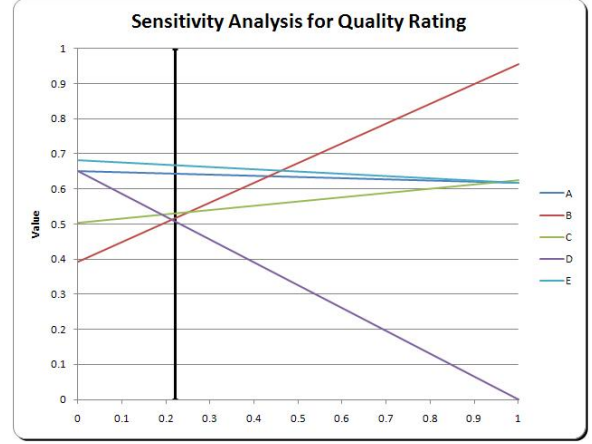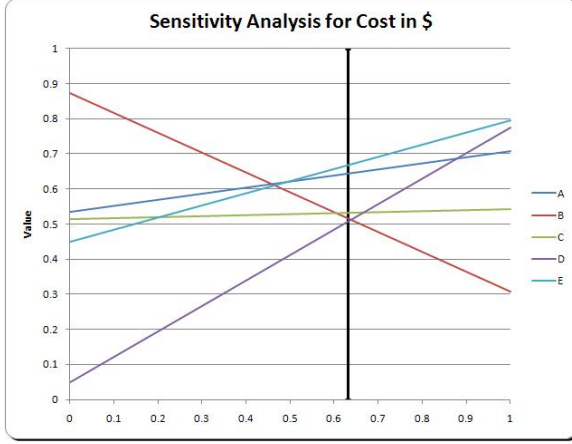
Figure 23. Sensitivity of Cost Weight on Preferred Alternative

Figure 24. Sensitivity of Quality Weight on Preferred Alternative

In addition to the weights, sensitivity analysis can also be performed on the value functions. A change in the mid-value of the cost value function changes the ρ parameter for the Exponential SDVF. The effects of manipulating ρ between 1 and 50 for the cost SDVF are given in Figures 25 and 26. In general, if cost is valued higher, more value is achieved shown by the values on the left of each graph. Changes in the value of cost also results in more value coming from quality, resulting in those alternatives with better quality to rank higher. A decision maker can continue to easily weigh tradeoffs between competing measures through the use of these methods and view the consequences of each of the decisions made for weights and values of the hierarchy. This isn't possible or convenient with other methods of modeling such as AHP.

76
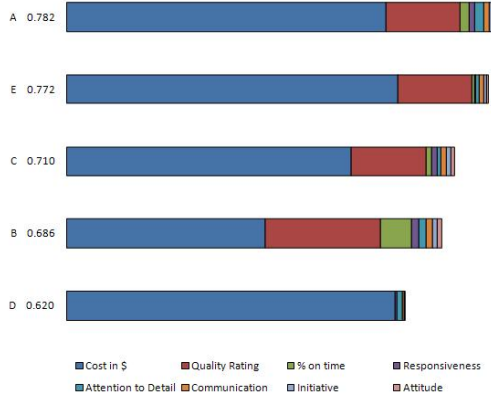
Figure 25. **Preferred Alternatives using Cost Value Function with** $\rho = 1$
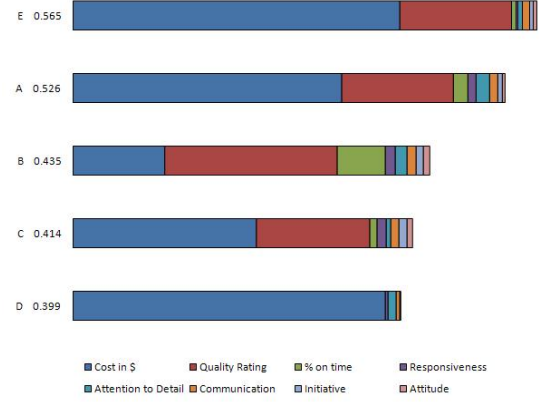
Figure 26. **Preferred Alternatives using Cost Value Function with** $\rho = 50$

Although prescriptive solutions are provided by the additive value model, we recommend using the analysis as a guide to make a better decision. If after applying sensitivity analysis, certain alternatives consistently rank near the bottom, dismiss them and refocus energy to making a decision among the remaining strong alternatives. This also can result in a portfolio of carriers, where the highest ranking alternatives receive some part of the overall business.

3.6 Managerial Implications and Discussion of the Consequences of applying the proposed ideas

Applying the proposed methodology is simple and profitable for those making decisions about logistics. Although similar methods have been proposed in the past, value-focused thinking adds several perks not available with other methods. Decision makers can quickly rank order potential suppliers, or alternatives for any problem, without the need to re-score each time a new supplier/alternative becomes available. This strength alone leads us to conclude that VFT may be a more practical

methodology than AHP for logistics problems. An initial commitment to build a VFT hierarchy saves the time and energy to formulate AHP models. Secondly, the proposed methodology is much easier to implement than AHP. Complicated eigenvalues and eigenvectors are unnecessary with VFT, the only requirement in a knowledge of basic algebra and a decision maker. Third, VFT allows post-analysis techniques such as sensitivity analysis without the need to re-score and re-weight alternatives. Lastly, approaching the problem from a value-focused thinking mentality forces the decision maker to think about what is truly valued in the decision. When making a decision among a list of provided alternatives (perhaps from a consulting firm), if no alternatives provide adequate value to the decision, the decision maker can request more alternatives and quickly value them again.

The methodology is easily applied using a spreadsheet and the techniques in this paper, and can easily be coded into Microsoft Excel. Keep in mind that as with any methodology, ignoring parts of the recommended procedure results in poor models, common mistakes in value modeling are found in [62]. For a copy of a Visual Basic coded hierarchy builder that makes the analysis easy, contact the corresponding author.

3.7 Conclusions

The purpose of the VFT process is to help decision makers make better decisions. Through completion of the process, a decision maker will truly understand what he values in his decision, and consequently will make a better decision even if he chooses not to use the model prescriptively. The VFT process helps the decision maker understand the strategic problem he is facing and assists in viewing the effects of making tradeoffs among competing objectives. Rather than simply comparing available alternatives, VFT unveils what value an alternative truly has to the decision maker. This

allows the decision maker to seek out better alternatives if the available alternatives are not achieving what is valued in the decision. This is especially important in the supply chain where boundless alternatives exist. In supplier selection, numerous suppliers can be easily valued, providing the decision maker with a defendable solution for a list of preferred suppliers through the use of this powerful tool, and quickly valuing any alternative without the need to re-weight each time more are available. Value-focused thinking is applied successfully in countless areas in the literature, the time for logistics is now.

Appendix A

Table 9. Calculating the Exponential Constant

$z_{0.5}$	R	$z_{0.5}$	R	$z_{0.5}$	R	$z_{0.5}$	R
0	-	0.25	0.41	0.5	Inf	0.75	-0.41
0.01	0.01	0.26	0.44	0.51	-12.5	0.76	-0.39
0.02	0.03	0.27	0.46	0.52	-6.24	0.77	-0.36
0.03	0.04	0.28	0.49	0.53	-4.16	0.78	-0.34
0.04	0.06	0.29	0.52	0.54	-3.11	0.79	-0.32
0.05	0.07	0.3	0.56	0.55	-2.48	0.8	-0.3
0.06	0.09	0.31	0.59	0.56	-2.06	0.81	-0.29
0.07	0.1	0.32	0.63	0.57	-1.76	0.82	-0.27
0.08	0.12	0.33	0.68	0.58	-1.54	0.83	-0.25
0.09	0.13	0.34	0.73	0.59	-1.36	0.84	-0.24
0.1	0.14	0.35	0.78	0.6	-1.22	0.85	-0.22
0.11	0.16	0.36	0.85	0.61	-1.1	0.86	-0.2
0.12	0.17	0.37	0.92	0.62	-1	0.87	-0.19
0.13	0.19	0.38	1	0.63	-0.92	0.88	-0.17
0.14	0.2	0.39	1.1	0.64	-0.85	0.89	-0.16
0.15	0.22	0.4	1.22	0.65	-0.78	0.9	-0.14
0.16	0.24	0.41	1.36	0.66	-0.73	0.91	-0.13
0.17	0.25	0.42	1.54	0.67	-0.68	0.92	-0.12
0.18	0.27	0.43	1.76	0.68	-0.63	0.93	-0.1
0.19	0.29	0.44	2.06	0.69	-0.59	0.94	-0.09
0.2	0.3	0.45	2.48	0.7	-0.56	0.95	-0.07
0.21	0.32	0.46	3.11	0.71	-0.52	0.96	-0.06
0.22	0.34	0.47	4.16	0.72	-0.49	0.97	-0.04
0.23	0.36	0.48	6.24	0.73	-0.46	0.98	-0.03
0.24	0.39	0.49	12.5	0.74	-0.44	0.99	-0.01

Appendix B

Table 10. Supplier Selection Data

Measures	Supplier				
	A	B	C	D	E
Cost	$247.46	$249.15	$248.28	$247.04	$246.90
Rejects	33	19	40	83	23
Supplied Products	23,169	26,492	28,284	28,832	16,156
Early Deliveries	1	1	0	1	1
Late Deliveries	1	0	3	7	2
Over/Short Deliveries	2	1	3	6	1
Total Deliveries	23	26	28	28	16
Initiative	3	4	5	2	3
Attitude	3	5	4	1	3
Responsiveness	3	4	4	2	1
Attention to detail	5	4	2	3	2
Communication performance	3	4	3	2	3

Defective PPM $= \frac{Rejects}{SuppliedProduct}$ x 1,000,000

Total Delivery Defects = Early Deliveries + Late Deliveries + Over Deliveries

% on time $= \frac{TotalDeliveries - TotalDeliveryDefects}{TotalDeliveries}$ x 100

Note: The defective PPM is used for the quality rating in Figure 19.

IV. The Multiobjective Average Longest Path Problem

Jeremy D. Jordan, Capt, USAF

Jeffery D. Weir, PhD

Doral E. Sandlin, Lt Col, USAF

Steven P. Dillenburger, Capt, USAF

Abstract

We integrate multiobjective decision methods, such as the analytic hierarchy process or value focused thinking, with the shortest/longest path problem for directed graphs. Since a decision maker desires to maximize value with these techniques, this creates the Multiobjective Average Longest Path (MALP) problem. The MALP allows multiple quantitative and qualitative factors to be captured in a network environment without the use of multicriteria methods, which typically only capture 2-3 factors before becoming intractable. The MALP (equivalent to average longest path) and the average shortest path problem for general graphs are NP-hard, proofs are provided. The MALP for *directed acyclic graphs* can be solved quickly using an existing algorithm or a dynamic programming approach. The existing algorithm is reviewed and a new algorithm using DP is presented. We also create a faster heuristic allowing solutions in $O(m)$ as opposed to the $O(nm)$ and $O(n^3)$ solution times of the optimal methods. This scaling heuristic is empirically investigated under a variety of conditions and is easily modified to approximate the longest or shortest average path problem for directed acyclic graphs. Finally, the steps used by the existing algorithm and dynamic programming approach automatically generate an efficient frontier for a special case of the bicriteria average shortest path problem involving arcs and value. The efficient frontier allows a decision maker to make tradeoffs between increasing value in the network and decreasing the number of arcs used in the chosen path. We provide the problem formulation and solution. The methods are discussed in the context of a transportation mode selection decision.

Keywords

network flows, decision analysis, heuristics, multiple objective programming

4.1 Introduction

Multiobjective techniques such as the analytic hierarchy process [89] and value focused thinking [61],[60] are widely used decision making techniques [38], [104]. Integrating these with other mathematical methods is also important [50], [103]. In this paper, we combine these multiobjective methods with network optimization, specifically the shortest path problem for directed acyclic graphs. The shortest path problem is thus transformed into a highest or longest average path problem in order to maximize value within the network setting, allowing multiple qualitative and quantitative factors to be captured without the use of multicriteria optimization. Normal shortest path algorithms cannot solve the average shortest or longest path problem. Since we are dealing with a non-linear cost function, using any normal shortest path algorithm will only produce a longest or shortest path. One way to solve this problem is through the use of non-additive shortest path algorithms. The idea of non-additive paths is a relatively recent development in the network optimization literature. Research motivation comes from the fact that not all network paths are additive in nature, that is a path cost may be some function other than simply the addition of all the arcs costs. The methods were sparked by the traffic equilibrium problem and are discussed in [40] and [41]. These algorithms give a solution for the general non-additive shortest path problem but are complicated slow heuristics. The average path is actually just a special case of the non-additive shortest path where the function is simply the sum of arc costs of the path divided by the number of arcs. Because of its simple structure, faster solutions are attainable. This paper describes the methodology in terms of VFT, however AHP could be substituted without complication; the value function is

just replaced with the AHP function. The general additive value model is given by

$$v_j(x) = \sum w_i v_i(x_{ij}) \tag{15}$$

for $i = 1, 2, ..., n$ measures and $j = 1, 2, ..., m$ alternatives, where $v_i(x_{ij})$ is the single dimension value function of measure i for alternative j, w_i is the weight of measure i, and $v_j(x)$ is the multiobjective value for alternative j. Value-focused thinking seeks to solicit the important factors in a decision and quantify the alternatives based on this information. This allows the decision maker to obtain an actual value of their decision as well as generate new alternatives.

The MALP replaces the normal cost function in a shortest path problem with an AHP or VFT function as in Equation 15. The result is a shortest path problem with a nonlinear cost function. Recall the shortest path problem. Assume a graph $G = (V, E)$ to be a directed acyclic network with $n \in V$ vertices and $m \in E$ arcs. Each arc $(i, j) \in E$ has some cost $c_{i,j}$ and flow $x_{i,j}$ associated with its use. The source and sink vertices are designated s and t, respectively. The linear programming formulation of the general shortest path problem is:

$$\min z = \sum_{(i,j) \in E} c_{i,j} x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = \begin{cases} 1, & \text{if i=s;} \\ -1, & \text{if i=t;} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{i,j} \geq 0 \ \forall \ (i, j) \in E$$

$$\tag{16}$$

85

where $x_{i,j}$ is the flow from vertex i to j and $c_{i,j}$ is the cost of a unit of flow from i to j. This is easily solved using the simplex method or any other LP algorithm. However, since it maintains a unique structure, faster algorithms are available. Dijkstra's algorithm [3] is the most common shortest path algorithm and solves any shortest path problem in $O(n^2)$. However, since we are concerned with directed acyclic graphs only, even faster algorithms exist. A dynamic programming algorithm that solves the DAG in linear time $O(m)$ is called the Reaching Method. It begins with a topological sort of the nodes and labels each vertex successively.

Step 1: Topologically order the DAG G

Step 2: For $i = 1, ..., n$, set dist(i)=0

Step 3: For $i = 1, ..., n - 1$, for each edge $V(i), u$ outgoing from $V(i)$, if dist(V_i) + $G(V_i, u) >$ dist(u), then set dist$(u) =$ dist$(V_i) + G(V_i, u)$

Step 4: dist(n) is the longest path to n

The algorithm above is easily modified to gather the shortest path to every node as well. This is the underlying dynamic programming approach behind the proposed scaling heuristic.

This problem is important for many reasons and useful in a variety of applications. Some examples include transportation mode selection, driving directions, and finance. In each of these, a decision maker may desire to incorporate more than 1 or 2 factors into his decision. For instance, in transportation mode selection, a DM may desire to consider cost, speed, and reliability in choosing the best carrier to ship goods. Perhaps a driver is not interested simply in time, but also weather, road conditions, scenery, and traffic conditions. Each investor may esteem different financial variables when allocating his resources. Analytic Hierarchy applications abound as well [103,

104]. Many of these applications could be modeled using shortest paths with multiple factors.

4.2 Formulations and Complexity

Here we formulate the multiobjective average shortest path problem and present a multicriteria optimization problem of the multiobjective average shortest path problem.

4.2.1 Multiobjective Average Longest Path Problem.

Combining the shortest path problem with the additive value model results in a unique formulation. Simply substituting the $c_{i,j}$ in Formulation 16 with the $v_j(x)$ in Equation 15 ignores the fact that we are seeking to find the best average value through the network. Such a substitution simply results in a longest path rather than a highest average. Substituting the AHP or VFT function and dividing by the number of arcs in the solution gives a non-linear formulation of the multiobjective average longest path (MALP):

$$\max z = \frac{\sum\limits_{(i,j)\in E} \left(\sum\limits_{k=1}^{l} w_k v_{i,j,k} \right) x_{i,j}}{\sum x_{i,j}}$$

$$s.t. \sum_{(i,j)\in E} x_{i,j} - \sum_{(j,i)\in E} x_{j,i} = \begin{cases} 1, & \text{if i=s;} \\ -1, & \text{if i=t;} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{i,j} \geq 0 \ \ \forall \ \ i,j \in E$$

$$(17)$$

where w_k is the weight of measure k, $v_{i,j,k}$ is the value function at measure k between

nodes i and j, $x_{i,j}$ is the flow from vertex i to j, and $\sum_{k=1}^{l} w_k = 1$. The $c_{i,j}$ from the shortest path problem in Formulation 16 is replaced with the the value function in Equation 15. This sum is divided by the sum of $x_{i,j}$'s; this remains integer because the constraint matrix is unimodular.

4.2.1.1 NP-hardness.

Theorem 1 *The average longest path problem is strongly NP-Complete.*

Proof 1 *Reduce from the longest path problem which is strongly NP-complete (Garey and Johnson 1979). The problem is:*

INSTANCE: Given a graph $G = (V, E)$, edges $e \in E$ with some cost $c_{i,j}$, positive integer K, and specified vertices $s, t \in V$.

QUESTION: Is there a simple path in G from s to t of length K or higher.

Assume we are given an instance of the longest path problem with n nodes such that every node i possesses a directed arc to the next node $i+1$ for $i = 1, 2, ..., n-1$ and all these edges $e \in E$ have cost $c_{i,j} = u$. Also assume no cycles are present and all other edges $e^* \in E$ have cost $c_{i,j} < u$. Let $K = u(n-1)$ and $u \in Z^+$. To reduce from the longest path problem, construct the same instance but solve using the average longest path. A constructed example instance for $n = 4$, $K = 6$, and $c_{i,j} = 2$ is given in Figure 27.

Clearly, the answer to the longest path problem is yes if and only if the objective function value of the average longest path problem is $\frac{u(n-1)}{n-1} = u$. This can be easily seen from the figure; the longest path through the network is 6 while the average longest path is 2. Solving the longest path problem in the constructed instance will

88

always yield a value of $u(n-1)$ and solving the average longest path problem will always give u. Moreover, if we let a cycle Φ be positive such that the costs of $e \in \Phi$ are greater than the costs of e not in Φ, the longest path and the longest average path include Φ and the solution is therefore unbounded. The average longest path problem is therefore strongly NP-complete.
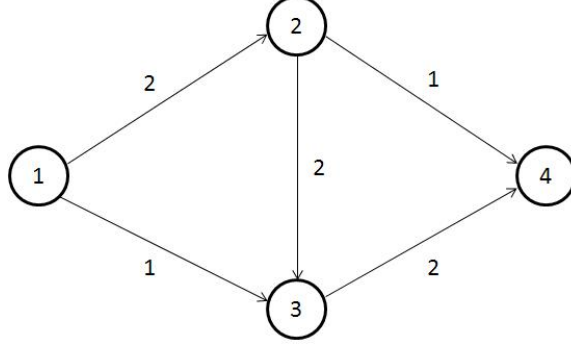
Figure 27. Constructed Instance of the Average Longest Path Problem

If G is a directed acyclic graph, the longest path problem can be solved in polynomial time [68]. In the next sections, we show the same is true for the average longest and shortest path problem for directed acyclic graphs.

Theorem 2 *The average shortest path problem is strongly NP-hard.*

Proof 2 *The proof is similar to Theorem 1 except we convert to the shortest path problem, another known NP-hard problem for general graphs [42]. The only difference in Figure 27 is the graph is negated. The answer to the longest path problem is yes if and only if the average longest path is $-u$. If the value of the average longest path problem is $-u$, the longest path problem must have a value of $-u(n-1)$. Hence, the average shortest path problem for is strongly NP-hard.*

For shortest average paths in directed acyclic graphs, a polynomial algorithm exists [108]. The longest average path for dag's is similarly solved. In Section 4.3,

we introduce a faster polynomial algorithm and linear heuristic that can be used for longest or shortest average paths in directed acyclic graphs.

4.2.2 A Special Case of the Bicriteria Average Shortest Path Problem.

The multicriteria optimization version of the multiobjective average shortest path problem in this section allows a decision maker to make tradeoffs between increasing value in the network and decreasing arcs utilized in the chosen path. Average shortest path algorithms and heuristics may take lengthy paths (in terms of arcs) in order to increase the overall average. However, in some applications such as transportation mode selection, taking long paths may cause unseen problems. Increased handling normally results in increased costs, increased damage/loss, and increased chance of mistakes. The multicriteria optimization for the MALP allows the decision maker to tradeoff value with additional arcs in the network path by generating all pareto optimal solutions. Because multiple objectives are handled by the additive value model and an average calculation is needed, the problem does not remain a bi-objective multicriteria optimization problem. Rather, a biobjective *average* shortest path problem is created, the first criteria being the average value model and the second being the number of arcs in the solution path. This is important because including more than three criteria typically makes the problem intractable. The average in the original MALP is still needed and results in a special structure. The special case of the bicriteria average shortest path problem for the MALP problem is:

$$\min z = \lambda_1 \left(\frac{\sum\limits_{(i,j) \in E} (1 - v_{i,j}) x_{i,j}}{\sum x_{i,j}} \right) + \lambda_2 \sum x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = \begin{cases} 1, & \text{if i=s;} \\ -1, & \text{if i=t;} \\ 0, & \text{otherwise.} \end{cases}$$

$$\sum \lambda_k = 1$$

$$x_{i,j} \geq 0 \ \ \forall \ (i,j) \in E$$

$$(18)$$

where $x_{i,j}$ is the flow between node i and j (1 if an arc is used), $v_{i,j}(x)$ is the multiobjective value of using node i and j, and λ_k is the multicriteria weight of criteria k.

This problem is similar but not quite the same as the more general fixed charge problem, initially formulated by Hirsch and Dantzig in 1968 [49], [97]. A variety of approaches exist to solve the problem [78], [18], [31], [106] and could be used as heuristics. Formulation 18 could also be *estimated* as a special case of the bicriteria shortest path problem [35], for which many algorithms exist. Since it's also a special case of the bicriteria average shortest path problem, faster algorithms are available. In fact, a separate algorithm is not necessary. We show the information attained from the solution approaches to Problem 17 is sufficient.

4.3   Solution Approaches

Three solution methods to Problem 17 are given in this section. The first minimizes the average path and solves to optimality in $O(n^3)$. The second uses a dy-

namic programming approach to gather longest paths to every node in the network in $O(nm)$. The third is a heuristic based on dynamic programming that retrieves the longest path from source to sink in $O(m)$. Each of the algorithms are easily modified to find either the shortest or longest path in a directed acyclic graph.

4.3.1   Wimer's Average Path Length Minimization Algorithm.

[108] lays out the initial theoretic foundation for determining the optimal average path length through the path length minimization algorithm for directed acyclic graphs. This algorithm pursues the optimal path by determining the best average path of cardinality j at each node. First, each vertex is assigned a rank according to its maximum cardinality (number of arcs) of a path from s to the vertex. The source, s, is obviously assigned a rank of 0 and the sink, t has the highest rank. Vertices are numbered according to their rank, starting with s, and numbering vertices with equal ranks arbitrarily. So, s is numbered 1, t is numbered $|U|$ and for every arc $e(u, v)$ the vertex u is assigned a smaller number than the vertex v.

Define $G = (U, E)$. Let u be a vertex on some path from source s to sink t. Only the shortest path with cardinality j can be part of the shortest average arc length path from s to t. Each vertex $u \in U$ is next assigned a vector $L(u)$ of length $|U|$. The jth element of $L(u)$, $L_j(u)$ with $(0 \le j \le |U| - 1)$, is the minimum length of any path from s to u with cardinality j. $\Pi_j(u)$ denotes the minimum length path or paths. Since $G(U, E)$ is acyclic, the cardinality of a path cannot be greater than $|U| - 1$. If no path exists for a cardinality, the path is assigned ∞. Another vector P_u of length $|U|$ is associated with u, whose jth element $P_j(u)$ is the last vertex preceding u on $\Pi_j(u)$. This is the vertex v for which $L_{j-1}(v) + l(e(v, u)) = L_j(u)$. If $L_j(u) = \infty$, then $P_j(u) = 0$

Starting at s, a new vertex is marked at each iteration until t is reached. Once

a vertex u is labeled, the length of the shortest path from s to u is known for every cardinality between 0 and $|U| - 1$. The sets of arcs entering and leaving $u \in U$ are denoted $\Gamma^{in}(u)$ and $\Gamma^{out}(u)$. The algorithm is as follows:

Step 0: Initialization. Set $L_0(s) = 0$ and $L_j(s) = \infty$, $1 \leq j \leq |U| - 1$. Mark s and set $T = U - \{s\}$. For every $u \in T$ set $L_j(u) = \infty$, $0 \leq j \leq |U| - 1$. For every $u \in U$ define $P_j(u) = \emptyset$, $0 \leq j \leq |U| - 1$.

Step 1: New Vertex Selection. Find a vertex $u \in T$ for which all the tail vertices of the arcs in $\Gamma^{in}(u)$ are already marked. Such a vertex must exist since $G(E, U)$ is an acyclic digraph with a single source and a single sink whose vertices are numbered as described above.

Step 2: Updating the minimum path lengths. Determine the shortest path length vector L_u by considering every vertex v for which $e(v, u) \in \Gamma^{in}(u)$ as follows.

$$L_j(u) = min\{L_{j-1}(v) + l(e(v, u))|e(v, u) \in \Gamma^{in}(u)\}, \quad 1 \leq j \leq |U| - 1. \qquad (19)$$

Let v^* be the vertex obtained when solving 19 for given u and j. Then, set $P_j(u) = v^*$.

Step 3: Updating the set of marked vertices. Mark u and set $T = T - \{u\}$.

Step 4: Termination Test. If $u = t$ then go to Step 5, else go to Step 1.

Step 5: Retrieving the minimum average arc length path.

Upon termination, every $L_j(t) \leq \infty$ is the length of the shortest path from s to t among all the paths of cardinality j. For every j satisfying $L_j(t) = \infty$ there exists no path of cardinality j from s to t. Evidently, $min\{L_j(t)/j|1 \leq j \leq |U| - 1$ yields the minimum average arc length for any path from s to t. Let j^* be the cardinality of the path for which the minimum average arc length was obtained. Then, the desired path is retrieved by traversing backwards from t to s as follows. We start from t and go backwards to the vertex stored in $p_{j^*}(t)$. We then go backwards to the vertex stored

in $P_{j^*-1}[P_{j^*}(t)]$ and continue in the same manner until s is reached.

Wimer [108] also presents a heuristic in addition to their optimal algorithm. The vertex balancing algorithm retrieves the minimum average length path using a user input accuracy setting. While their first algorithm is cubic in nature, this algorithm runs in exponential time $O(2^U(U + log(\frac{1}{\epsilon})))$. The algorithm is claimed to converge very quickly in practice, however it was only tested on networks with a maximum of 10 nodes.

4.3.2 Orlin's Algorithm.

The average shortest path problem for a directed acyclic graph can be solved using dynamic programming [82]. "Assuming you are finding a shortest path from node s to node t, you can let d(j, k) be the shortest path from node s to node j with exactly k arcs. (This does not work in networks with cycles because it actually computes the shortest walk from s to j with k arcs.) The values of d(,) can be computed using dynamic programming. The shortest average length of a path from s to t is min d(t, k)/k for k = 2 to n-1. The same technique computes a shortest average path from node s to each other node." We now develop this idea.

Assume a graph G with node-arc cost matrix C and cost coefficients $c_{i,j}$ denoting a cost from arc i to j. Where no arc exists, use 0 when minimizing and ∞ when maximizing. The optimal matrix containing the best path in $k = 1$ to $n - 1$ arcs to each of the $r = 2$ to n nodes is denoted Θ. Therefore, Θ an $(n - 1) \times (n - 1)$ matrix of optimal paths. Each column of Θ is divided by its representative k to give the best average to each node in k steps. The algorithm stated formally as a maximization is:

Step 1: Assign 1st column of Θ. Let 1st column of Θ equal 1st row of C,

$\Theta(k - 1, 1) = C(1, l)$ for $k = (2, \ldots, n)$ and $l = (2, \ldots, n)$.

Step 2: Assign remainder of 1st row of Θ. Let the first row of Θ be equal to 0,

$\Theta(1, k) = 0$ for $k = (2, \dots, n - 1)$.

Step 3: Create remainder of Θ *matrix columns.* While $\Theta(r, k-1)$ and $C(r+1, l+1)$ are greater than 0, for $k = (2, \dots, n - 1)$,

$\Theta(l, k) = \mathrm{MAX}(\Theta(r, k - 1) + C(r + 1, l + 1))$ for $l = (2, \dots, n - 1)$ and $r = (1, \dots, l - 1)$. Else, $\Theta(l, k) = 0$.

Step 4: Use Θ *matrix to compute average distances.* Divide each element of columns 2 through $n - 1$ of Θ by $k = (2, \dots, n - 1)$, giving the average value from the source to each node.

Step 5: Retrieve Optimal Solution. Optimal average value from s to t is

$\mathrm{MAX}(\Theta(n - 1, :))$.

If minimizing, change the maximums to minimums and let the first row of Θ be equal to ∞ instead of 0 in Step 2. Longest average paths to every node are available in Step 4. This algorithm solves in $O(nm)$.

4.3.3 Scaling Heuristic.

We begin with a topological sort of the nodes. Following the methodology of [108], we first assign each vertex according to its maximum cardinality (number of arcs) of a path from s to the vertex. The source, s, is obviously assigned a rank of 0 and the sink, t has the highest rank. Vertices are numbered according to their rank, starting with s, and numbering vertices with equal ranks arbitrarily. So, s is numbered 1, t is numbered $|U|$ and for every arc $e(u, v)$ the vertex u is assigned a smaller number than the vertex v. Let the rank of a particular node be denoted $R(j)$. Hence $c_{i,j}$, the cost of a unit of flow from i to j, need only be scaled by

$$\frac{R(j) - R(i)}{R(t)}. \tag{20}$$

Let

$$c_{i,j}^* = c_{i,j} \left(\frac{R(j) - R(i)}{R(t)} \right),$$

(21)

and the average shortest path problem becomes

$$\min z = \sum_{(i,j) \in E} c_{i,j}^* x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = \begin{cases} 1, & \text{if i=s;} \\ -1, & \text{if i=t;} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{i,j} \geq 0 \;\; \forall \;\; (i,j) \in E$$

(22)

Formulation 22 can now be solved using the Reaching Algorithm from Dynamic Programming in linear time $O(m)$. The heuristic is summarized as such:

Step 1: Use Node-Arc Incidence Matrix. Temporarily replace each $c_{i,j} > 0$ with 1.

Step 2: Use Reaching Algorithm. Use the Reaching Algorithm to solve the longest path to each vertex giving its maximum cardinality (number of arcs) from s to the vertex.

Step 3: Assign Ranks to Vertices. Vertices are numbered according to their rank, starting with s, and numbering vertices with equal ranks arbitrarily. Let the rank of a particular node be denoted $R(k)$, for $k = 1, 2, ..., n$ vertices.

Step 4: Scaling Arc Costs. Scale the original $c_{i,j}$'s by $\frac{R(j)-R(i)}{R(t)}$. Let $c_{i,j}^* = c_{i,j} \left(\frac{R(j)-R(i)}{R(t)} \right)$.

Step 5: Substitute. Replace $c_{i,j}$ with $c_{i,j}^*$ in original formulation.

Step 6: Use Reaching Algorithm again. Solve new formulation with the Reaching Algorithm.

Summarizing, the reaching algorithm is used to obtain the maximum number of arcs to each node; this is used to scale the original cost coefficients. The reaching algorithm is then used a second time to solve the scaled network to optimality. Small errors are possible when using the heuristic due to the scaling effect, however theoretically no bound exists on the worst case error.

Theorem 3 *The worst case error of the Scaling Heuristic is infinitely large.*
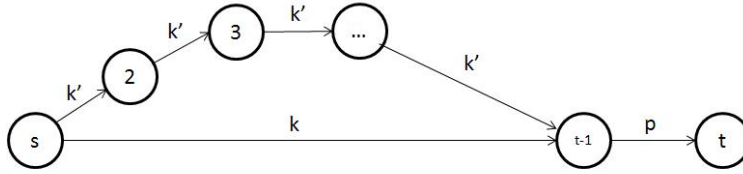
Figure 28. Constructed Graph

Proof 3 *Assume a network of two paths p_1 and p_2, with cardinalities of $t - 1$ and 2, respectively. Let p denote the arc cost from node $t - 1$ to t, k the arc cost from s to $t - 1$, and k' the arc cost of all other nodes. Furthermore, assume $k' > k$, $t > 3$, and p_2 has a higher average value than p_1. The path p_2's average value is then $\frac{k+p}{2}$, while p_1's average value is $\frac{(t-2)k'+p}{t-1}$. Standard error is given by:*
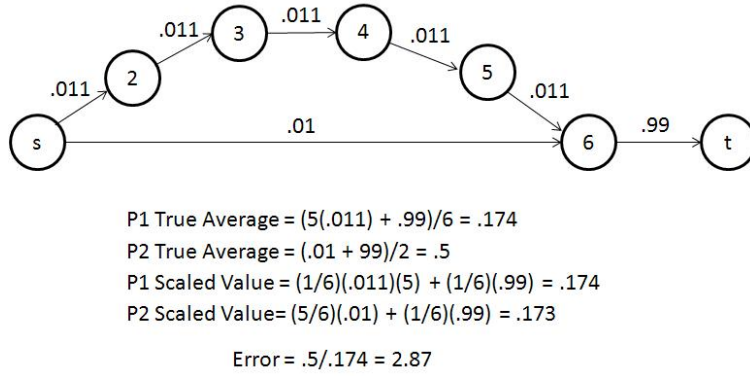
$$x = \frac{\frac{k+p}{2}}{\frac{(t-2)k'+p}{t-1}}$$

(23)

97

Rearranging terms yields:

$$x = \frac{k(t-1) + p(t-1)}{2k'(t-2) + 2p}$$

(24)

As $t \to \infty$, $k \to 0$, $k' \to 0$, *and* $p \to 1$, *clearly* $x \to \infty$.

An example of this phenomenon is given in Figure 29. Clearly, Path 2 has a better average path length than Path 1. After scaling however, Path 1 appears better than Path 2.

P1 True Average = (5(.011) + .99)/6 = .174
P2 True Average = (.01 + 99)/2 = .5
P1 Scaled Value = (1/6)(.011)(5) + (1/6)(.99) = .174
P2 Scaled Value= (5/6)(.01) + (1/6)(.99) = .173

Error = .5/.174 = 2.87

Figure 29. Example Constructed Graph

The error for any given network will vary; even in this example the error is dependent on k, k', and p. For instance, letting $t \to \infty$, $k \to 1$, and $p \to 0$, the heuristic finds the optimal solution, $x \to 1$. Additionally, the probability of generating the network above approaches 0 as $t \to \infty$ [36]. Quite unlikely is the case that any network be generated that yields an error greater than 2. Nevertheless, the worst case error can always be bound by bounding the arc costs.

Theorem 4 *Bounding an arc cost gives bounds on the worst case error of the Scaling Heuristic.*

Proof 4 *Assume the same network as in Theorem 3 and Figure 28. If a bound is set on k, a corresponding bound on x can be achieved. Arbitrarily letting $k \geq .1$ gives a bound on x of:*

$$x = \frac{k(t-1) + p(t-1)}{2k'(t-2) + 2p}$$

$$x \to 5.5$$

(25)

Clearly, as k approaches p, the error approaches 0. Therefore, bounding arc costs gives bounds on the worst case error.

Again, this example is a worst case network, the probability of encountering this network in practice is extremely small. Even if the network did arise, bounding with reasonable values negates the effects. Furthermore, while Theorem 4 gives some comfort, testing shows that the error in most networks is minuscule. In the next section, we examine the performance of the heuristic for 10,000 randomly generated matrices with bounded and unbounded arc costs. While the worst case error approaches ∞ in theory, in practice the heuristic performs very well.

4.4 Scaling Heuristic Performance

Three experimental factors were used in testing: number of nodes in the network, density, and range in the arc costs. Range is the range of values allowed in the multiobjective value function at each arc and represents the bounding discussed above. Since values between 0 and 1 are the only achievable levels, the allowable range cannot be above 1.

To test the heuristic's performance, networks of varying arc cost ranges, nodes sizes, and densities were generated using a random number generator. Arc cost range

varies between 0 and 1 (representing any desired number), node sizes between 50 and 350, and density between .02 and .49 (maximum density in a DAG). Let
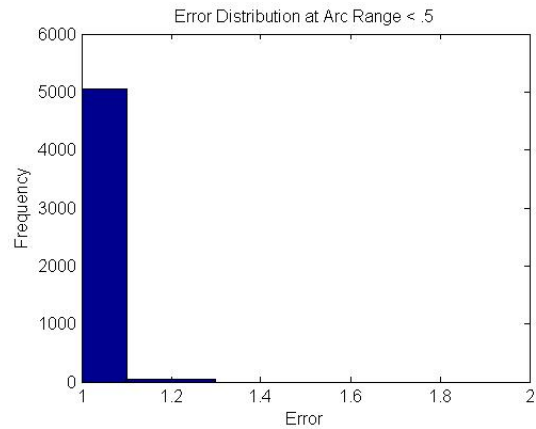
X_1 - Range in Arc Value

X_2 - # of Nodes

X_3 - Density

The heuristic was applied to 10,000 randomly generated networks with varying levels of X. Orlin's algorithm solved the networks to optimality, the 2 solutions are compared to obtain the error. The overall average error of the scaling heuristic is 3% with a 95% confidence the error will be below 13%. To further analyze the origin of the error, we compared distributions for two groups for each variable. Range appears to be the most influential factor. Of course, as the range of possible values increases, one expects error to increase as well. In the case of the MALP, all values will be between 0 and 1 and therefore a range of 1 is the worst case. Even at this range, the heuristic performs within bounds of the optimal solution divided by 1.05 for the most part, which is very desirable. Regardless of the setting at each X, the heuristic performs very well.
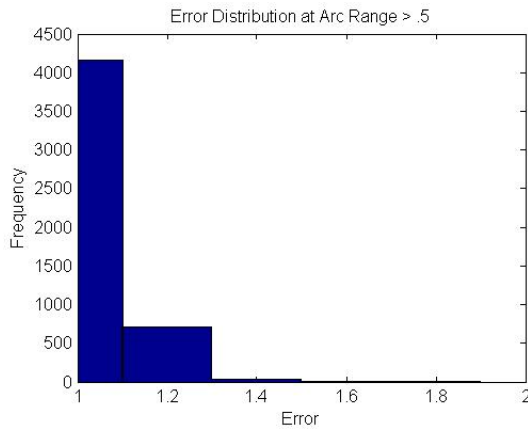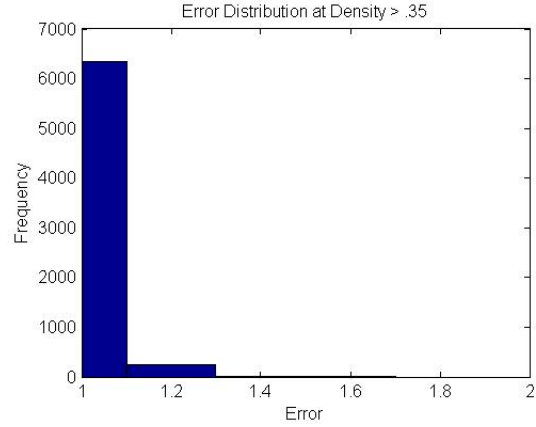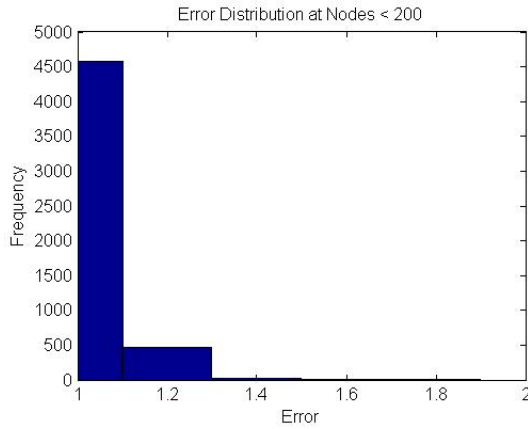

Figure 30. Error Distribution   **Figure 31. Error Distribution**
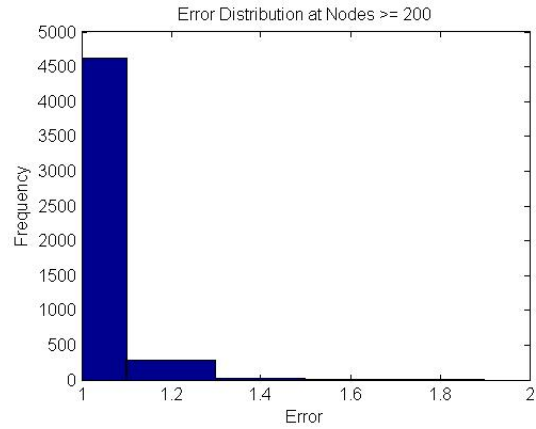
100
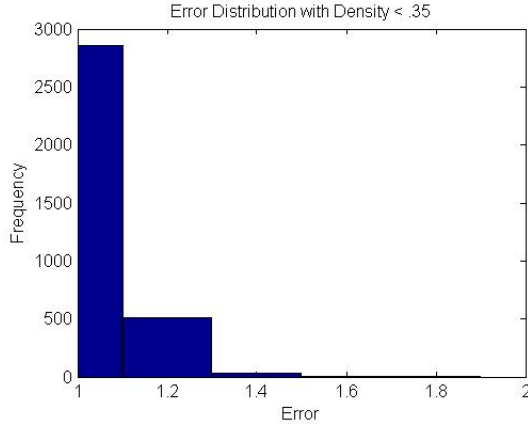
Figure 32. Error Distribution

Figure 33. Error Distribution

Figure 34. Error Distribution

Figure 35. Error Distribution

Simple linear equations were fit to the error to show performance for each of the variable settings. The results are intuitive and confirm the results above in Figures 37, 36, 38. Error increases as the range in arc value increases. A certain decrease is realized when increasing the number of nodes in the network or increasing the density. In general, more nodes equal more possible routes. As possible routes increase and the heuristic misses the optimal solution, it's more likely the next best route is closer in value to the optimal route. Similarly, increasing the density of the network yields more possible routes and therefore less chance for error. As density decreases, fewer routes are available, and the probability of the next best route being close to the optimal

101

decreases. These results reinforce the benefits of using the heuristic for large scale networks. Since the optimal algorithms are slightly less efficient on dense networks, it makes sense to use the heuristic for large scale problems with high density.
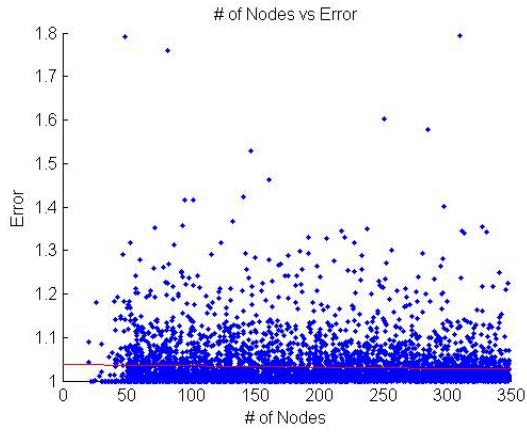
Figure 36. Error vs Nodes
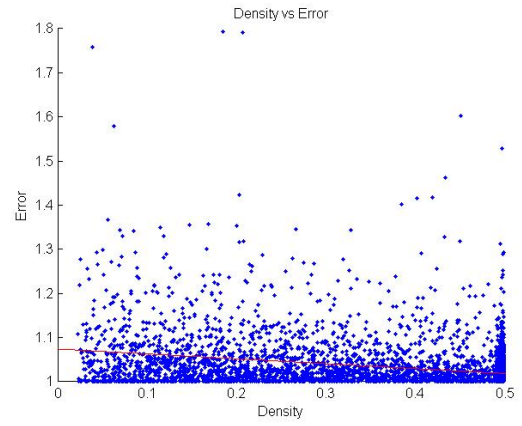
Figure 37. Error vs Density
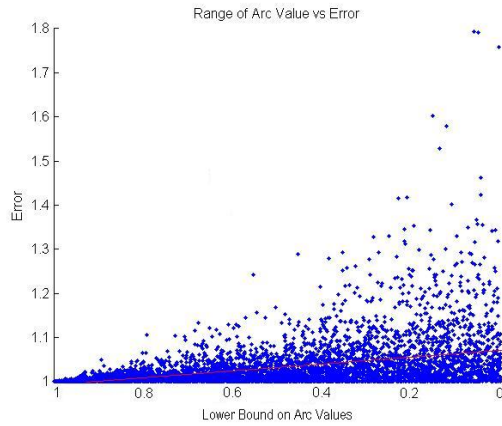

Figure 38. Error vs Range of Arc Values

Fitting a least squares regression model to the error and including terms with a p-value less than .01 yields the following equation:

$$Y = 1.03151 + .06705X_1 - .00004X_2 - .07311X_3 - .00014X_1X_2 - .14259X_1X_3. \quad (26)$$

where Y is the error. The regression confirms our conclusions from the observed results. Each of the variables and two of the interactions significantly contribute to the error. The model is a poor predictor of error however, with an R^2 below .2. Next, the usefulness of the methods is demonstrated on an example application problem.

4.5 Application

This methodology is directly applicable to the transportation mode selection problem. It allows selection of the best mode of transportation given important factors to a decision maker. For instance, in freight transportation, important factors include cost, shipping speed, reliability, loss and damage, and flexibility [25]. These qualitative and quantitative variables are best captured with the VFT or AHP function in Equation 15. Each arc represents the various transportation options and transfer points are nodes. Clearly, the combination of transportation options that maximizes the average value through the network is the desired route. This section creates notional networks of 100 nodes, varying densities, and arc values ranging between 0 and 1.

The scaling heuristic solves a network of any size in linear time, but the focus now is on the value in using Wimer's or Orlin's algorithms. These algorithms are used to create optimal and non-optimal paths and values from which a decision maker can make tradeoffs. The optimization problem from Formulation 18 in Section 4.2

underlie the graphs in this section. As a weighted sum problem, we are seeking to make tradeoffs between increasing value in the network and decreasing arcs utilized. Each of the graphs are generated wi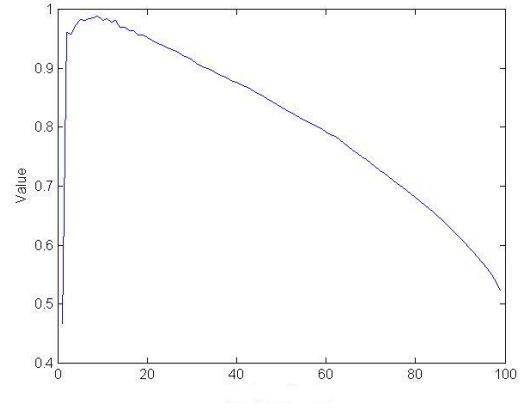th the distance matrix from Orlin's or Wimer's algorithm. Because the shortest path to each node is known, we can easily get the shortest average path to every node, along with the number of arcs to each node. Formulation 18 is a special case of the bicriteria average shortest path problem which allows tradeoff between total value and number of arcs.

Figures 39,40,41,42 show surfaces created by random graphs. Each surface is unique. In Figure 39 for instance, the optimal solution is somewhere around 7 arcs. However, the decision maker could choose to sacrifice roughly .07 in value to go from source to sink in one arc. In contrast, going from the source to sink in one step in Figure 40 creates a significant loss of value, around .5. Figure 42 shows a discontinuous efficient frontier. For arc lengths around 4, the solution is strictly dominated by the arc length of 1, but also dominated by some arc lengths greater than 5.
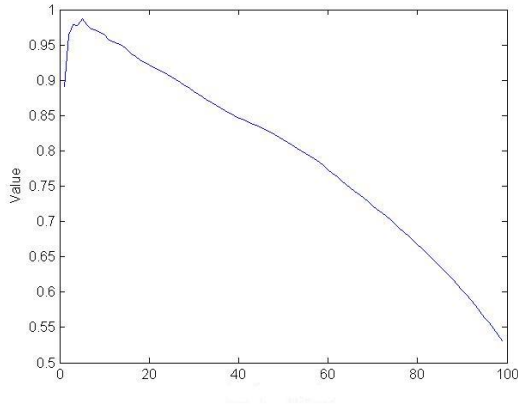
Figure 39. Values vs Arcs   Figure 40. Values vs Arcs

The optimal solution to the graph in Figure 43 is a 10 arc path (1 10 18 28 39 47 48 84 91 94 100) with an optimal average value of 0.9765. Table 11 shows all paths shorter than the optimal path. All paths longer than the optimal solution in this case

104
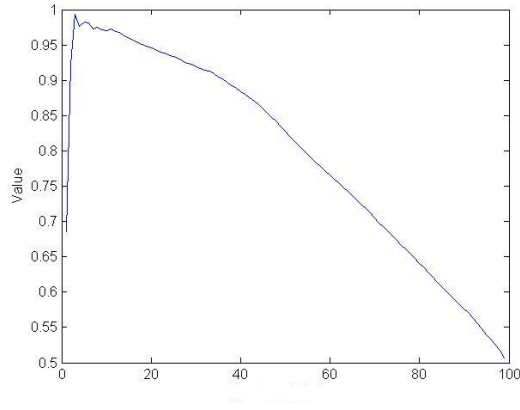
Figure 41. Values vs Arcs
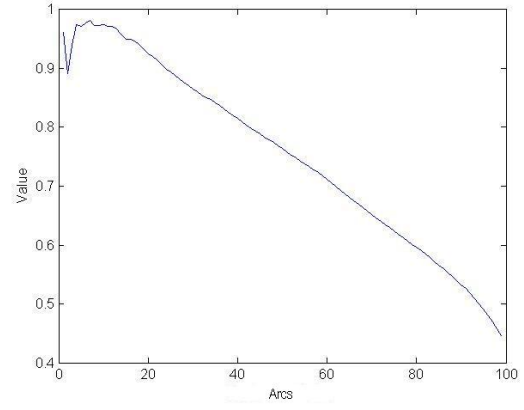
Figure 42. Values vs Arcs

give a worse solution. The optimal path and all paths in Table 11 strictly dominate

the longer paths to the right. In this case, the user may choose the path with 2 arcs

and a value of .9223 rather than choosing the optimal value of .9765 with 9 arcs. In

many cases, an added value of .05 or .1 may not be worth adding multiple arcs. In

other cases however, transfers may require minimal effort and therefore the user may

choose to optimize value and ignore arc length.

Table 11. Arcs vs Value

Number of arcs	1	2	3	4	5	6	7	8	9
Value	0.6905	0.9223	0.9549	0.9604	0.9631	0.9665	0.9685	0.9733	0.9765
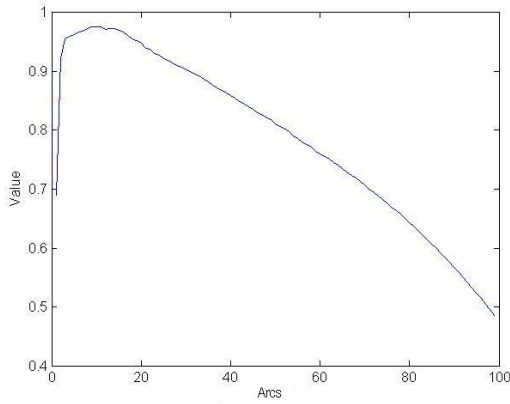
Figure 43. Values vs Arcs
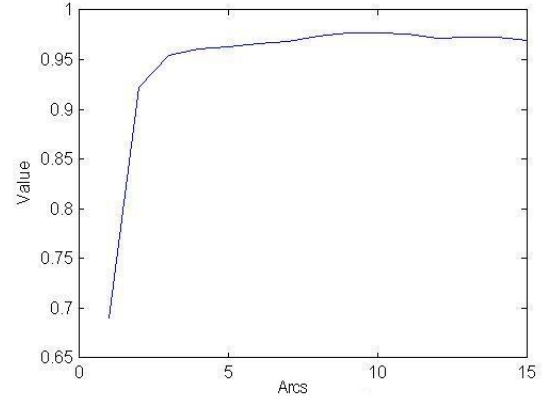
Figure 44. Closeup of Efficient Frontier in Figure 43
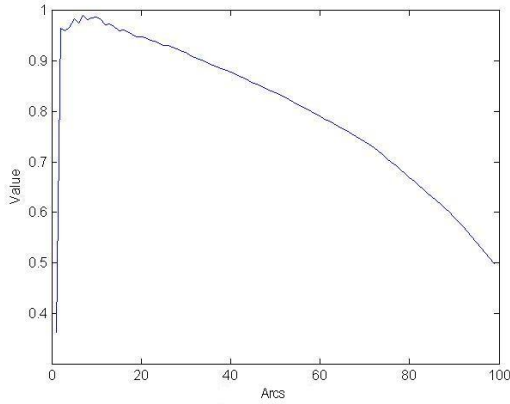
Figure 45. Values vs Arcs
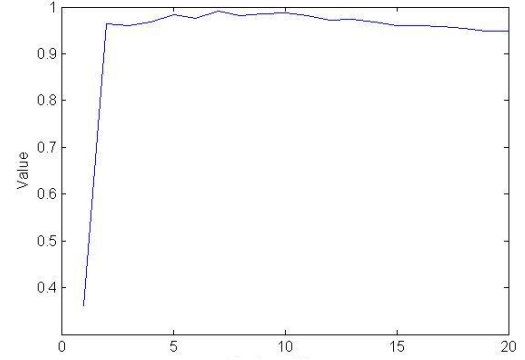
Figure 46. Closeup of Efficient Frontier in Figure 45

4.6 Conclusion

Integrating value focused thinking or the analytic hierarchy process with the shortest path problem allows a decision maker to capture multiple factors in a network environment. The resulting multiobjective average longest path problem has clear applications to a variety of problems. We presented an algorithm and heuristic to solve the MALP for directed acyclic graphs more efficiently than current methods and revealed the usefulness of both our methods and those in the literature. The heuristic

performed satisfactorily when applied to a number of randomly generated matrices. As number of nodes and density increase, the heuristic obtains better solutions and should therefore be used in large scale networks. Additionally, since tradeoffs between value and number of arcs exist, a multicriteria optimization of the MALP is useful to a decision maker. The path chosen along the efficient frontier created by Orlin's and Wimer's algorithms is dependent on the application. When transfers are a concern, shorter paths with less value may be more appealing. Transportation mode selection is one intuitive application for this methodology yet it could be applied to a multitude of existing problems.

Acknowledgements

V. The Multiobjective Average Minimum Cost Flow Problem

Jeremy D. Jordan, Capt, USAF

Jeffery D. Weir, PhD

AFIT/DS/ENS/12-09

Abstract

Multicriteria (or Multiobjective) decision analysis (MCDA/MODA) is implemented into the minimum cost flow problem; this creates a multiobjective average minimum cost flow (MAMCF) problem, a problem equivalent to the average minimum cost flow problem. We show the problem is NP-complete for general graphs. However, for directed acyclic graphs efficient pseudo-polynomial time heuristics are possible. An average shortest path algorithm is implemented in a successive shortest path fashion to create an efficient average minimum cost flow heuristic. Furthermore, an average shortest path heuristic is used successively as an even faster average minimum cost flow heuristic. This scaling method is developed to estimate the average shortest path problem and extended to approximate the average minimum cost flow problem. Both heuristics are then proven to have an infinitely large error bound. However, in random networks the heuristics generate solutions within a small percentage of the optimal solution. Finally, a general bicriteria average minimum cost flow (BAMCF) problem is given. In the case of the MAMCF, decision makers may choose to minimize arcs in a path along with maximizing multiobjective value. Therefore, a special case of the BAMCF is introduced allowing tradeoffs between arcs and value. This problem is clearly NP-hard, however good solutions and equivalent solution times are attainable using the pseudo-polynomial time heuristics for solving the average minimum cost flow problem.

5.1   Introduction

Multiobjective decision analysis (MODA) or multicriteria decision analysis (MCDA) techniques such as the analytic hierarchy process [89] and value focused thinking [61],[60] are used widely in decision making applications [38], [104]. Putting these in the context of the minimum cost flow problem creates a unique problem we call the multiobjective average minimum cost flow (MAMCF) problem. Current minimum cost flow algorithms cannot solve the MAMCF or the average minimum cost flow problem (AMCF) as the problems are non-additive in nature. Non-additive minimum cost flow research is an up and coming area, however algorithms are not yet available. The average minimum cost flow problem is a special case of the non-additive minimum cost flow problem; specialized heuristics can therefore exploit the problems structure for efficiency. We begin by introducing concepts from decision analysis and network optimization.

The general additive value model is

$$v_j(x) = \sum w_i v_i(x_{ij}) \tag{27}$$

for $i = 1, 2, ..., n$ measures and $j = 1, 2, ..., m$ alternatives, where $v_i(x_{ij})$ is the single dimension value function of measure i for alternative j, w_i is the weight of measure i, and $v_j(x)$ is the multiobjective value for alternative j. The additive value model is used throughout as an example, however other multiobjective methods are easily interchangeable with Equation 27 in the proposed methodology.

Network optimization is a thriving area of research and multiple problems exist [3]. Given a graph $G = (V, E)$, the general minimum cost flow (MCF) problem is:

$$\min z = \sum_{(i,j)\in E} c_{i,j} x_{i,j}$$

$$s.t. \sum_{(i,j)\in E} x_{i,j} - \sum_{(j,i)\in E} x_{i,j} = s(i) \quad \forall \ i \in N$$

$$l_{i,j} \le x_{i,j} \le u_{i,j} \quad \forall \ (i,j) \in E$$

$$(28)$$

where $x_{i,j}$ is the flow from vertex i to j, $c_{i,j}$ is the cost of a unit of flow from i to j, $l_{i,j}$ is the lower bound on the flow from i to j, $u_{i,j}$ is the upper bound on the flow from i to j, and $s(i)$ is the supply or demand at node i.

The shortest path problem is a special case of the minimum cost flow problem where $s(i)$ is 1,-1, or 0. A dynamic programming algorithm that solves the shortest path problem for directed acyclic graphs in linear time $O(m)$ is called the Reaching Method [28]. The same method can be used in a successive shortest path fashion to solve the minimum cost flow in directed acyclic graphs. Choose the shortest path and reduce flow by the minimum arc capacity on that path, then repeat until flow is satisfied. The Reaching Method begins with a topological sort of the nodes and labels each vertex successively.

Step 1: Topologically order the DAG G

Step 2: For $i = 1,...,n$, set dist(i)=0

Step 3: For $i = 1,...,n-1$, for each edge $V(i), u$ outgoing from $V(i)$, if dist(V_i) + $G(V_i, u) >$ dist(u), then set dist(u) = dist(V_i) + $G(V_i, u)$

Step 4: dist(n) is the longest path to n

The algorithm above is easily modified to gather the shortest or longest path to every node. This dynamic programming approach is used in the proposed successive *average* shortest path scaling heuristic.

Many times, a decision maker may need to make tradeoffs between differing objectives in a minimum cost flow environment. This problem is called the multiple objective minimum cost flow problem (MMCF) [47]. The multiobjective minimum cost flow problem is a special case of multiobjective linear programming (MOLP) which is a special case of multiple objective programming (MOP). For objective functions of 3 or greater, the MMCF is intractable. In fact, even in the special case of the bicriteria minimum cost flow (BMCF) with 2 objective functions, the number of non-dominated points on the efficient frontier is exponential in size [88]. The BMCF objective functions are

$$min(f_1(x), f_2(x)) \tag{29}$$

where the constraint set remains as that in Formulation 28. With this formulation, decision makers are able to make trade-offs between competing objectives and see the outcomes of these tradeoffs. [35] defines the types of efficiency for a feasible solution $\hat{x} \in X$:

- weak efficiency - no other $x \in X$ such that $f(x) < f(\hat{x})$

- strict efficiency - no other $x \in X$, $x \neq \hat{x}$ such that $f(x) \leq f(\hat{x})$

- proper efficiency - there is a real number $M > 0$ such that $\forall$ i and x $\in$ X satisfying $f(x) < f(\hat{x})$, there exists an index j such that $f(\hat{x}) < f(x)$ and $\frac{f(x)-f(\hat{x})}{f(\hat{x})-f(x)} \leq$ M.

The most popular method of generating the efficient set or set of pareto optimal solutions are though scalarization techniques such as the weighted sum method.

Letting $\lambda \in \mathbb{R}_{\geq}^p$, the weighted sum linear program LP(λ) is

$$\min \lambda^T C x$$
$$\text{subject to } Ax = b$$
$$x \geq 0.$$

The idea with this is to generate all efficient solutions. In the case of the MMCF however, this can be an exponential number of solutions. Therefore, more specialized algorithms exist to generate these solutions or some subset of the efficient solutions [47]. Generating all solutions is thought to be too confusing for the decision maker; generating some subset of the efficient solutions is more practical in many cases. The bicriteria average minimum cost flow introduced in the next section is similar to this problem but more difficult because of the inherent non-linearity.

5.2   Problem Definitions and Formulations

Variants of the minimum cost flow problem are plentiful [3]. The minimal average cost flow variant was first formulated to minimize a total cost, consisting of a fixed cost of using a network plus a variable cost per unit of flow, divided by the total flow [12]. Our variant of the problem seeks to minimize the average cost per unit of flow for *every* arc. The maximization version of the multiobjective formulation (using a value function as cost) as a non-linear optimization problem is:

$$\max z = \frac{\sum_{(i,j)\in E} \left(\sum_{k=1}^{l} w_k v_{i,j,k} \right) x_{i,j}}{\sum x_{i,j}}$$

$$s.t. \sum_{(i,j)\in E} x_{i,j} - \sum_{(i,j)\in E} x_{j,i} = s(i) \ \ \forall \ i \in N,$$

$$x_{i,j} \geq 0 \ \ \forall \ i,j \in E$$

$$\sum_{k=1}^{l} w_k = 1$$

$$(30)$$

where w_k is the weight of measure k, $v_{i,j,k}$ is the value function at measure k between nodes i and j, $s(i)$ is supply or demand entering or leaving the node i, and $x_{i,j}$ is the flow from vertex i to j. To convert this to a general average minimum cost flow problem, the $\sum_{k=1}^{l} w_k v_{i,j,k}$ is replaced by a cost coefficient $c_{i,j}$ and change from maximum to minimum. The constraint set remains unimodular similar to the normal minimum cost flow problem if each $s(i)$ and the lower and upper bounds on $x_{i,j}$ are integer. If minimization is required, simply negate the objective function or replace $\sum_{k=1}^{l} w_k v_{i,j,k}$ with $\sum_{k=1}^{l} 1 - w_k v_{i,j,k}$.

There is a negative to optimizing the average value through the network; the heuristics may choose a very long path (long in # of arcs) of optimal average value rather choosing a much shorter path with a slightly lower average value. One way to offset this problem is through a special case of the bicriteria average minimum cost flow problem trading off number of arcs with average value. The more general bicriteria average minimum cost flow is given first, where one objective function is required to be an average.

$$\min f_1(x) = \frac{\sum\limits_{(i,j) \in E} (c_{i,j}) x_{i,j}}{\sum x_{i,j}}$$

$$\min f_2(x)$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = s(i) \quad \forall \ i \in N,$$

$$x_{i,j} \geq 0 \quad \forall \ i, j \in E$$

$$(31)$$

where $x_{i,j}$ is the flow between node i and j, $c_{i,j}$ is the cost of using node i and j, $s(i)$ is supply or demand entering or leaving node i, and $f_2(x)$ is some function linear or otherwise. A special case of this problem trading off arcs and average value is

$$\min z = \lambda_1 \left(\frac{\sum\limits_{(i,j) \in E} (1 - v_{i,j}) x_{i,j}}{\sum x_{i,j}} \right) + \lambda_2 \sum x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = s(i) \quad \forall \ i \in N,$$

$$x_{i,j} \geq 0 \quad \forall \ i, j \in E$$

$$(32)$$

where $x_{i,j}$ is the flow between node i and j, $v_{i,j}$ is the multiobjective value of using node i and j, $s(i)$ is supply or demand entering or leaving node i, λ_k is the multicriteria weight of criteria k, and $\sum \lambda_k = 1$.

Bicriteria minimum cost flows are a special case of the multiobjective minimum cost flow problem, which is a special case of the more general multiobjective linear program. In the same way, this problem is special case of the BAMCF, which is a

special case of the multiobjective average minimum cost flow problem that is a special case of the multiobjective program. Generating efficient solutions for bicriteria minimum cost flow is intractable in general [88]. This obviously carries over to the more difficult bicriteria average minimum cost flow problem. In the upcoming sections, we show how to estimate a subset of the efficient frontier using a pseudo-polynomial time heuristic. This gives a workable number of efficient solutions from which to choose in a reasonable amount of time.

5.3   NP-Hardness

Garey and Johnson [42] describe the theory of NP-Completeness. Many problems are shown to be NP-complete by reduction to known NP-complete problems. Several problems are transformed into the exact cover by 3-sets (X3C) [42], which is transformed from the 3-dimensional matching, a known NP-complete problem [58]. In this section we show the *average* minimum cost flow problem reduces to the exact cover by 3-sets problem, thus proving the MAMCF or AMCF problem to be NP-complete. The following theorems hold for general graphs.

Theorem 5 *The average minimum cost flow problem is strongly NP-Complete.*

Proof 5 *Create an instance of the exact cover by 3-sets (X3C), a known NP-Complete problem. The X3C decision problem from [42] is:*

INSTANCE: A set X of $3q$ elements, $X = 1, 2, 3, ..., 3q$, and a collection $C = s_1, s_2, ..., s_m$, of sets of cardinality 3.

QUESTION: Does a subcollection $C' \subseteq C$ exist such that every element of X is contained in exactly one set of C'?

We transform the X3C problem into the average minimum cost flow problem by constructing an instance as such:

1. Create Nodes:

    (a) Define a source node s, with supply $3m$.

    (b) Define m transhipment nodes, each with an outdegree of 3, corresponding to the set C.

    (c) Define $3q$ sink nodes, each with demand 1, corresponding to the set X

2. Create Arcs:

    (a) Define arcs (s, C_i) with capacity 3 and cost 0 for $i = 1, 2, ..., m$.

    (b) Define arcs (C_i, x_j) when x_j is contained in the set C_i, for $i = 1, 2, ..., m$ and $j = 1, 2, ..., 3q$. Each of these arcs has capacity 1 and a cost of -1.

If the answer to the exact cover by 3-sets question above is yes, then clearly there exists an average minimum cost flow with an average of $-\frac{1}{2}$ (as calculated in Formulation 30) and a flow of $3q$ where q is obviously equal to the number of 3-sets chosen from C. Because the demand of the nodes in X is such that the sum is exactly $3q$ (or supply), any feasible flow through the network corresponds to an exact cover by 3-sets. Conversely, an average minimum cost flow through the network must satisfy demand in X or be the trivial solution 0. Thus, since the arcs (C_i, x_j) are less than 0, any average minimum cost flow solution will always equal $-\frac{1}{2}$ and have a flow of
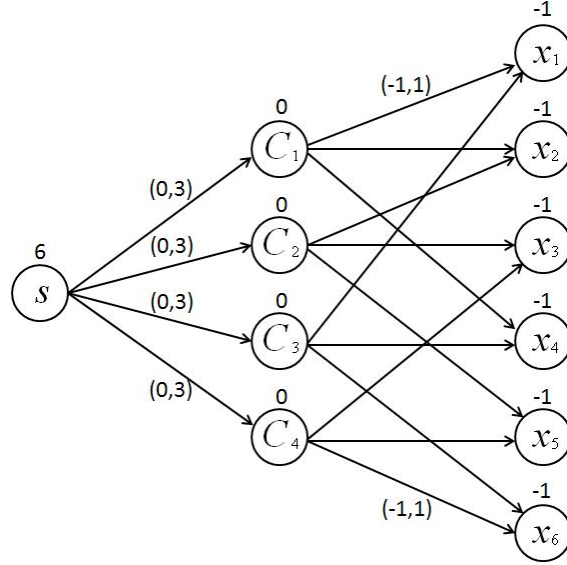
Figure 47. Constructed Instance of the Average Minimum Cost Flow Problem

$3q$. Therefore, any average minimum cost flow solution will always correspond to an exact cover by 3-sets.

Corollary 1 *Any average minimum cost flow problem is strongly NP-Complete even if the arc capacities are all equal to 1.*
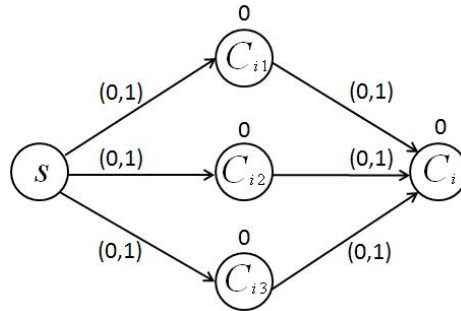
Figure 48. A Constructed Instance of the Average Minimum Cost Flow Problem with Arc Capacities of 1

Proof 6 *In Theorem 5.3, replace the arcs (s, C_i) with 3 arcs of capacity 1 for $i = 1, 2, ..., m$ and cost 0 as before. Also add $3m$ nodes between each s and C_i as shown*

118

in Figure 48. Each arc in Figure 47 now has a capacity of 1, the result follows.

The average maximum cost flow proof is similar and left to the reader. Given the average minimum cost flow problem is proven NP-hard, it follows that the biobjective average minimum cost flow problem is NP-hard. Even in the special bicriteria case of Formulation 32, the problem remains NP-hard. Directed acyclic graphs make the problem easier yet an exponential number of efficient solutions are still possible [88].

5.4 Pseudo-Polynomial Time Heuristics for Directed Acyclic Graphs

This section introduces two successive average shortest path heuristics to estimate the average minimum cost flow problem for directed acyclic graphs. Successive shortest path algorithms for the minimum cost flow problem are optimal, but this is not the case for the average minimum cost flow problem. Successive average shortest path heuristics obtain a feasible solution with each iteration but violate the mass balance constraints until near optimality is achieved. The algorithm terminates when the mass balance constraints are satisfied. Optimality is maintained throughout while the algorithm works to achieve feasibility. Although very efficient, we show the potential error of the successive average shortest path algorithms is infinitely large. However, with realistic bounds on our arc coefficients, the error is quite low.

5.4.1 Successive Average Shortest Path with Dynamic Programming.

The average shortest path problem for a directed acyclic graph can be solved using dynamic programming [81], [57]. We take this concept a step further and define the successive average shortest path heuristic. This works in the same fashion as successive shortest path algorithms for the minimum cost flow problem: a shortest average path is obtained and called a pseudoflow, the maximum flow possible is sent through this path and the arc capacities are updated, the algorithm continues until

119

all flow requirements are met.

Assume a graph G with node-arc cost matrix C^m and cost coefficients $c_{i,j}$ denoting a cost from arc i to j. Recall the supply to node i is $s(i)$; $s(s)$ and $s(t)$ for the source and sink nodes, respectively. Where no arc exists, use 0 when minimizing and ∞ when maximizing. The optimal matrix containing the best path in $k = 1$ to $n - 1$ arcs to each of the $r = 2$ to n nodes is denoted Θ. Therefore, Θ an $(n - 1) \times (n - 1)$ matrix of optimal paths. Each column of Θ is divided by its representative k to give the best average to each node in k steps. Furthermore, an arc capacity matrix is defined $\pi_{i,j}$, P_i is the ith path, and z_i corresponds to the objective function of the ith path. The algorithm stated formally as a maximization is:

Step 1: Assign 1st column of Θ. Let 1st column of Θ equal 1st row of C,

$\Theta(k - 1, 1) = C(1, l)$ for $k = (2, \ldots, n)$ and $l = (2, \ldots, n)$.

Step 2: Assign remainder of 1st row of Θ. Let the first row of Θ be equal to 0,

$\Theta(1, k) = 0$ for $k = (2, \ldots, n - 1)$.

Step 3: Create remainder of Θ matrix columns. While $\Theta(r, k-1)$ and $C(r+1, l+1)$ are greater than 0, for $k = (2, \ldots, n - 1)$,

$\Theta(l, k) = \text{MAX}(\Theta(r, k - 1) + C(r + 1, l + 1))$ for $l = (2, \ldots, n - 1)$ and $r = (1, \ldots, l - 1)$. Else, $\Theta(l, k) = 0$.

Step 4: Use Θ matrix to compute average distances. Divide each element of columns 2 through $n - 1$ of Θ by $k = (2, \ldots, n - 1)$, giving the average value from the source to each node.

Step 5: Determine Average Shortest Path. Optimal average value from s to t is

$\text{MAX}(\Theta(n - 1, :))$.

Step 6: Update Arc Capacities. Decrease arc capacities by $a_i = \min(\pi_{i,j})$ on path P_i corresponding to z_i.

Step 7: Update Supply, Flow, and Cost Matrix. Update supplies $s(i)$ and flow.

Cost matrix C^m updates by changing $c_{i,j}$ to 0 if $\pi_{i,j} = \min(\pi_{i,j})$. Reduce the remainder of $\pi_{i,j}$'s on path P_i by $\min(\pi_{i,j})$.

Step 8: Test for Near Optimality. If $\sum a_i = -s(t)$, you have satisfied all constraints and have a near optimal solution, else return to Step 1.

If minimizing, change the maximums to minimums and let the first row of Θ be equal to ∞ instead of 0 in Step 2. Longest average paths to each node are then known in Step 4. The running time of the algorithm is $O(nmC)$, where C is max capacity of arcs. Since C is not necessarily bound by a polynomial, the worst case complexity is pseudo-polynomial.

5.4.2 Successive Average Shortest Path with a Scaling Heuristic.

The running time of the successive average shortest path with a scaling heuristic is O(mC) linear time, where C is max capacity of arcs. This is an order of magnitude faster than the previous heuristic.

Following the methodology of [108], we first assign each vertex according to its maximum cardinality (number of arcs) of a path from s to the vertex. The source, s, is obviously assigned a rank of 0 and the sink, t has the highest rank. Vertices are numbered according to their rank, starting with s, and numbering vertices with equal ranks arbitrarily. So, s is numbered 1, t is numbered $|U|$ and for every arc $e(u, v)$ the vertex u is assigned a smaller number than the vertex v. Let the rank of a particular node be denoted $R(j)$. In the same as above, an arc capacity matrix is defined $\pi_{i,j}$, P_i is the ith path, and z_i corresponds to the objective function of the ith path. The heuristic is summarized as such follows:

Step 1: Use Node-Arc Incidence Matrix. Temporarily replace each $c_{i,j} > 0$ with 1.

Step 2: Use Reaching Algorithm. Use the Reaching Algorithm to solve the longest

path to each vertex giving its maximum cardinality.

Step 3: Assign Max Cardinality. Assign each vertex a rank according to its maximum cardinality (number of arcs) of a path from s to the vertex.

Step 4: Assign Ranks to Vertices. Vertices are numbered according to their rank, starting with s, and numbering vertices with equal ranks arbitrarily. Let the rank of a particular node be denoted $R(k)$, for $k = 1, 2, ..., n$ vertices.

Step 5: Scaling Arc Costs. Scale the original $c_{i,j}$'s by $\frac{R(j)-R(i)}{R(t)}$. Let $c_{i,j}^* = c_{i,j}\left(\frac{R(j)-R(i)}{R(t)}\right)$.

Step 6: Substitute. Replace $c_{i,j}$ with $c_{i,j}^*$ in original formulation.

Step 7: Use Reaching Algorithm again. Solve new formulation with the Reaching Algorithm to determine the shortest average path.

Step 8: Update Arc Capacities. Decrease arc capacities by $a_i = \min(\pi_{i,j})$ on path P_i corresponding to z_i.

Step 9: Update Supply, Flow, and Cost Matrix. Update supplies $s(i)$ and flow. Cost matrix C^m updates by changing $c_{i,j}$ to 0 if $\pi_{i,j} = \min(\pi_{i,j})$. Reduce the remainder of $\pi_{i,j}$'s on path P_i by $\min(\pi_{i,j})$.

Step 10: Test for Near Optimality. If $\sum a_i = -s(t)$, you have satisfied all constraints and have a near optimal solution, else return to Step 1.

Summarizing, the reaching algorithm is used to obtain the maximum number of arcs to each node. Using this, the original cost coefficients are scaled. The reaching algorithm is then used a second time to solve the shortest path in the scaled network to optimality. The arc capacities on this shortest path are decreased by the smallest capacity and the algorithm then repeats itself. Small errors are possible when using the heuristic due to the scaling effect, however no bound exists on the worst case error. In fact, it's been proven that the worst case error of the average Scaling Heuristic is infinitely large [57] even for the average shortest path problem.

5.4.3 Error Bounds.

Theorem 6 *The worst case error of the Scaling Heuristic for average shortest paths is infinitely large.*
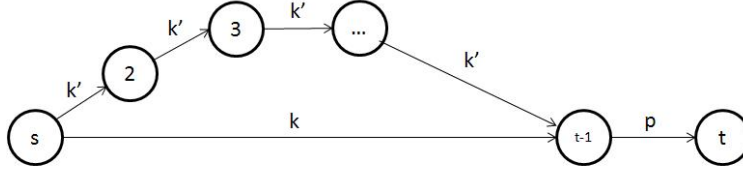
Figure 49. Constructed Graph

Proof 7 *Assume a network of two paths p_1 and p_2, with cardinalities of $t - 1$ and 2, respectively. Let p denote the arc cost from node $t - 1$ to t, k the arc cost from s to $t - 1$, and k' the arc cost of all other nodes. Furthermore, assume $k' > k$, $t > 3$, and p_2 has a higher average value than p_1. The path p_2's average value is then $\frac{k+p}{2}$, while p_1's average value is $\frac{(t-2)k'+p}{t-1}$. Standard error is given by:*

$$x = \frac{\frac{k+p}{2}}{\frac{(t-2)k'+p}{t-1}}$$

(33)

Rearranging terms yields:

$$x = \frac{k(t-1) + p(t-1)}{2k'(t-2) + 2p}$$

(34)

As $t \to \infty$, $k \to 0$, $k' \to 0$, and $p \to 1$, clearly $x \to \infty$.

The error for any given network will vary; even in this example the error is depen-

123

dent on k, k', and p. As $t \to \infty$, $k \to 1$, and $p \to 0$, the heuristic finds the optimal solution, $x \to 1$. Additionally, the probability of generating the network above approaches 0 as $t \to \infty$ [36]. Quite unlikely is the case that any network be generated that yields an error greater than 2.

This becomes a mute point since the worst case error for either of the above successive average shortest path heuristics approaches ∞ as well.

Theorem 7 *The worst case error of* any *successive average shortest path heuristic for the average minimum cost flow problem is infinitely large.*

Proof 8 *Assume a network with arc capacities of 1 as in Figure 50, we've already shown this problem to NP-Hard in Corollary 5.3. Denote P_i as path i for $i = 1$ to 4, with average values of z_i.*
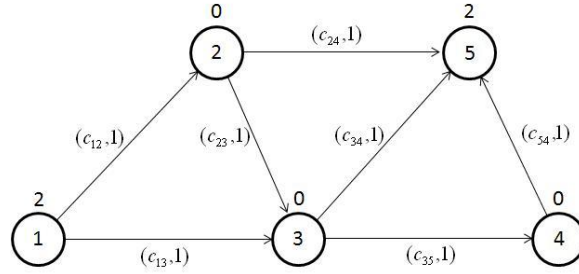
Figure 50. Constructed Instance of the Average Minimum Cost Flow Problem

P_1: *$1 \to 2 \to 4$*

P_2: *$1 \to 2 \to 3 \to 4$*

P_3: *$1 \to 3 \to 4$*

P_4: $1 \rightarrow 3 \rightarrow 5 \rightarrow 4$.

Assume $z_1 + z_3 < z_2 + z_4$ and $z_2 < z_1, z_3, z_4$. Let $z_4 \rightarrow \infty$. Any successive average shortest path algorithm will initially choose the shortest path through the network. In this case, when P_2 is chosen, the only path available to choose is P_4 since arc capacities are 1, meaning $z_2 + z_4 \rightarrow \infty$. The standard error is defined as $x = \frac{z}{z^*}$, where z^* is the optimal solution. Therefore $x = \frac{z_2 + z_4}{z_1 + z_3}$. Clearly, as $z_4 \rightarrow \infty$, $x \rightarrow \infty$.

While in theory, the worst case error of both heuristics approaches ∞, in practice the heuristics perform very well.

5.5 Heuristics Performance and Application

This section compares the performance of the heuristics and describes possible applications. For all the graphs in this section, assume we are trying to maximize value $v_{i,j}$ rather than minimize $1 - v_{i,j}$. As discussed above, the heuristics are interchangeable with respect to maximization and minimization; maximizing value in practice is more intuitive.

5.5.1 Performance Comparison of Heuristic Methods.

This section compares the performance of the successive shortest path heuristics for the average minimum cost flow problem. The best average minimum cost flow through the network can be no better than the best average shortest path through the network. Therefore, we compare the performance of the heuristics against each other and the average shortest path, giving a worst case error. In reality, the error will be lower. The matrices used in the analysis were randomly generated using random arc costs between 0 and 1, number of nodes between 50 and 350, and density between .01 and .49. Arc capacities were randomly assigned between 10 and 50 and

supply between 50 and 200. Obviously, increased supply and decreased arc capacities will result in greater estimated error. More repetitions completed by a successive shortest path heuristic will bring the overall average further from the original shortest path. The random components typically required 20 to 40 repetitions of the shortest path algorithm. Figure 51 shows the cumulative distribution of errors for the successive shortest path heuristic using dynamic programming. This heuristic performs very well in practice for random graphs, errors are almost entirely below 1.2, or 20% from the optimal solution, with 95% below 1.1 and none over 35%. The 95% confidence interval for the successive shortest path heuristic using dynamic programming is $1.024 < x < 1.029$.
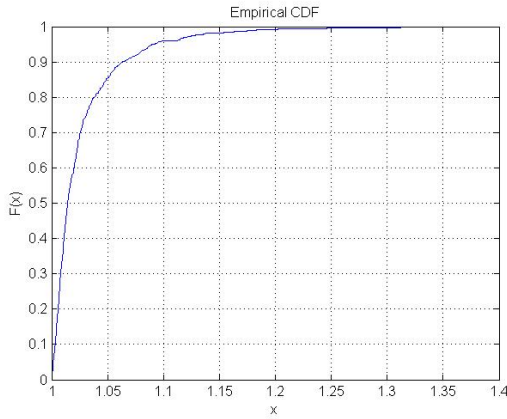
Figure 51. Cumulative Distribution Function for Successive SP with DP
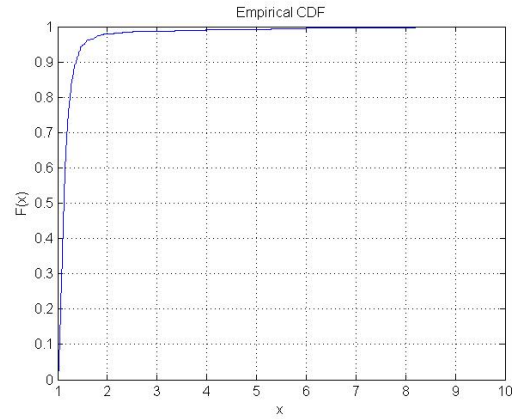
Figure 52. Cumulative Distribution Function for Successive SP with Scaling Method

The successive average shortest path heuristic using the Scaling method performed adequately as well. The majority of errors were below 1.4 with well over half below 1.2. 17 anomalies occurred with errors above 2 times the optimal, a worse case of 9.61. This is expected given the compounding error discussed in the proofs from the previous section. The 95% confidence interval for the successive shortest path heuristic using the scaling heuristic is $1.19 < x < 1.27$, and cumulative distribution

126

function is shown in Figure 52. Since the running speed of the heuristic is much faster, a decision maker will still benefit from using the heuristic on large scale problems.

5.5.2 Application.

In some areas of application, a decision maker may wish to not only maximize value, but also to minimize the number of arcs used in the solution. For instance, consider transportation mode selection. Nodes, or points of transfer, may have some damage percentage associated with them. Increased handling may not be desirable. The bicriteria problem introduced in Section 3.2 is the solution to this problem.

The bicriteria shortest path problem is already proven NP-complete, except in the case of a directed acyclic graph where it's P-complete [94], [35]. The bicriteria average shortest path problem is at least as hard and must have similar complexity. Since we've proven the average minimum cost flow problem is NP-Complete, it follows that the bicriteria average minimum cost flow problem is NP-Complete. Recall the parts of the objective function in Formulation 32. The first part of the objective function is simply the average

$$\frac{\sum\limits_{(i,j)\in E} (1 - v_{i,j})x_{i,j}}{\sum x_{i,j}}.\tag{35}$$

The second half is the total capacity of each arc used in the solution of the average minimum cost flow

$$\sum x_{i,j}.\tag{36}$$

The bicriteria problem works in the same way as the successive shortest path algorithm with one slight change. Rather than minimizing the average shortest path, the algorithm minimizes the objective function from Problem 32 in Step 5 of the algorithm

127

from Section 5.4.1. It's also intuitive to treat the problem as an integer program. The $n-1$ row of the Θ matrix is the shortest average path to the sink node in some $y_{i,j}$. Let $a_1 = 0$, $b_1 = 1$, $A_1 = \min\left(\sum_{(i,j)\in E}(1-v_{i,j})x_{i,j}\right)$, $B_1 = \max\left(\sum_{(i,j)\in E}(1-v_{i,j})x_{i,j}\right)$, $a_2 = 0$, $b_2 = 1$, $A_2 = \min\left(\sum y_{i,j}\right)$, and $B_2 = \max\left(\sum y_{i,j}\right)$ where $y_{i,j} = 1$ when arc i,j is used for some number of arcs $y_{i,j}$. Normalizing the objective function gives

$$\min z = \lambda_1\left(a_1 + \left(\left(\sum_{(i,j)\in E}((1-v_{i,j})x_{i,j} - A_1)\right) * \frac{b_1 - a_1}{B_1 - A_1}\right)\right) + \lambda_2\left(a_2 + \left(\left(\sum(y_{i,j} - A_2)\right) * \frac{b_2 - a_2}{B_2 - A_2}\right)\right).$$

(37)

The algorithm chooses the number of arcs and value that minimizes Equation 37 in accordance with the decision makers preferences as captured in the λ values.
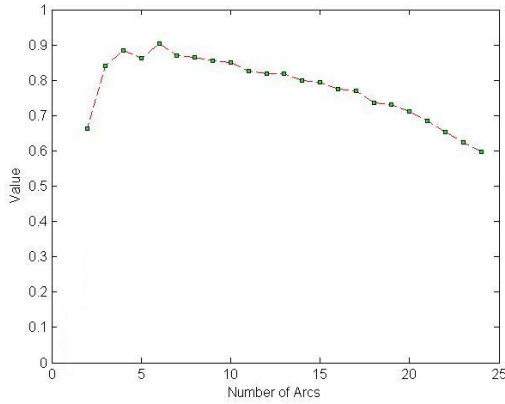
Figure 53. Efficient Frontier
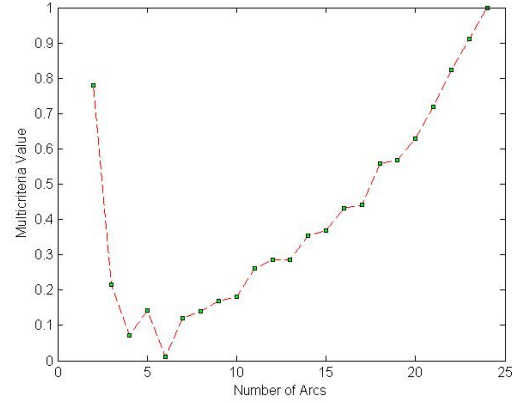
Figure 54. Multicriteria Value for $\lambda_1 = 1$, $\lambda_2 = 0$

The algorithm begins with generation of the shortest path efficient frontier as in Figure 53. These values are transformed in accordance with the decision makers preference as shown in Figures 54, 55, and 56. The minimum value relating to some path is chosen; arc capacities along this path decreased and the algorithm generates a new shortest path efficient frontier. Notice the minimum points in Figures 54, 55, and 56 correspond to one of the solutions on the efficient frontier in Figure 53.

128
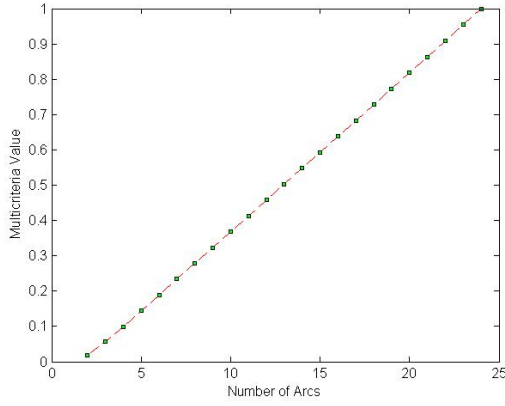
Figure 55. Multicriteria Value for $\lambda_1 = 0$, $\lambda_2 = 1$
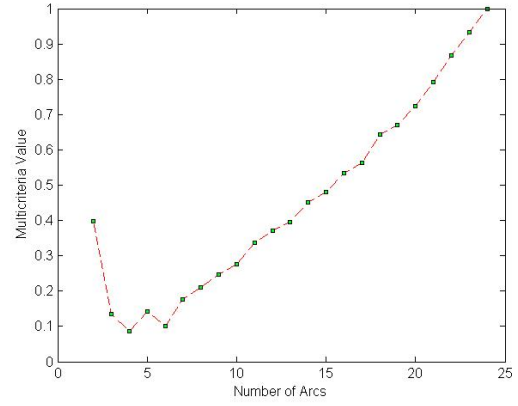
Figure 56. Multicriteria Value for $\lambda_1 = .5$, $\lambda_2 = .5$

Figures 57,58,59,60,61,62 show the efficient frontier for the shortest path of another graph G as it evolves over time using the successive shortest path algorithm. Slight changes occur with each iteration of the algorithm. The shortest path efficient frontiers will change based on the λ value specified.
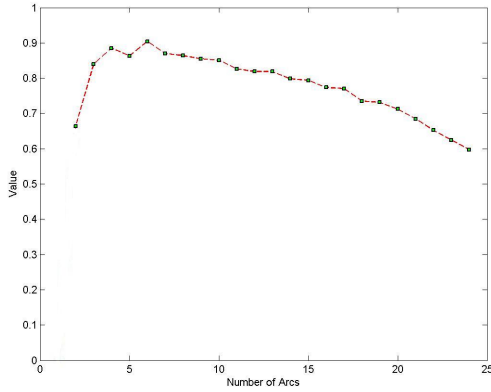
Figure 57. 1st Efficient Frontier
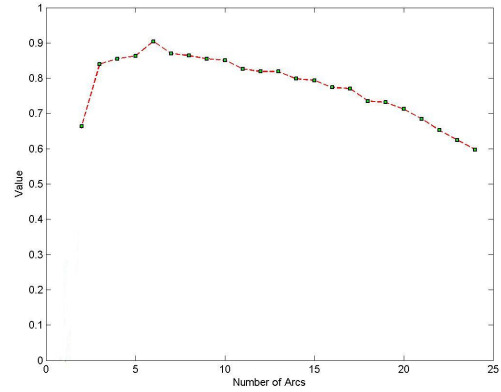
Figure 58. 2nd Efficient Frontier

Although we are using an algorithm to solve each of these average shortest path problems, the efficient frontier of the average minimum cost flow problem is still an estimate, be it a very good estimate as shown in the previous section. The pseudo-polynomial time heuristic from Section 5.4.1 is used to generate the estimated efficient

129
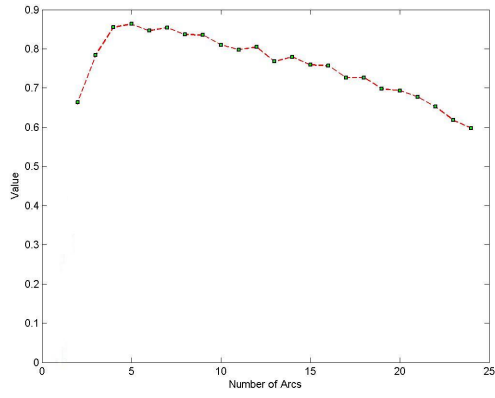
Figure 59. 3rd Efficient Frontier
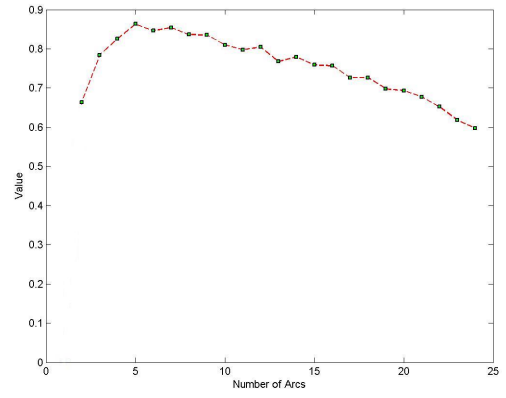
Figure 60. 4th Efficient Frontier
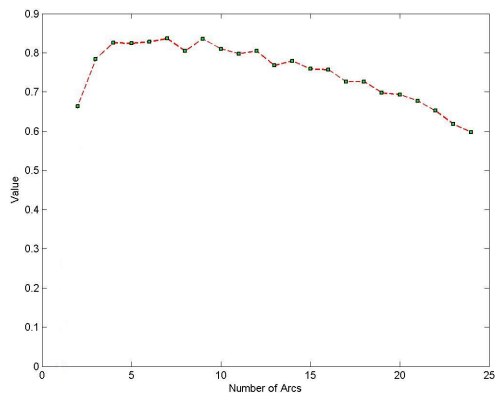
Figure 61. 5th Efficient Frontier
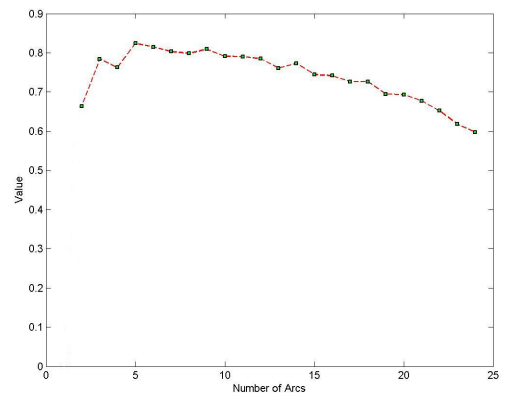
Figure 62. 6th Efficient Frontier

130

solutions. It remains pseudo-polynomial because the number of value for λ must be bounded by some constant. Therefore, the complexity reduces to $O(\lambda nmC) \approx O(nmC)$.

The estimated efficient solutions for example average minimum cost flow problems are given in Figures 63, 64, 65, and 66. For each of these, graphs of around 100 nodes were generated. Arc ranges varied between 0 and 1, total supply between 50 and 150, and density between .08 and .45. Higher supply resulted in more repetitions of the average shortest path algorithm, which resulted in more possible efficient solutions. Another interesting point to note, generating the efficient frontier sometimes produced a solution better than the optimal average in both value and arcs.
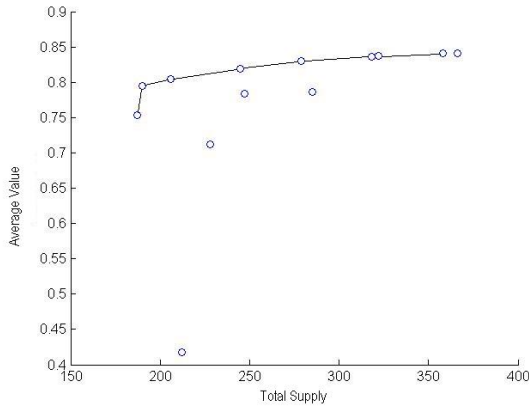
Figure 63. MCO Efficient Frontier   **Figure 64. MCO Efficient Frontier**
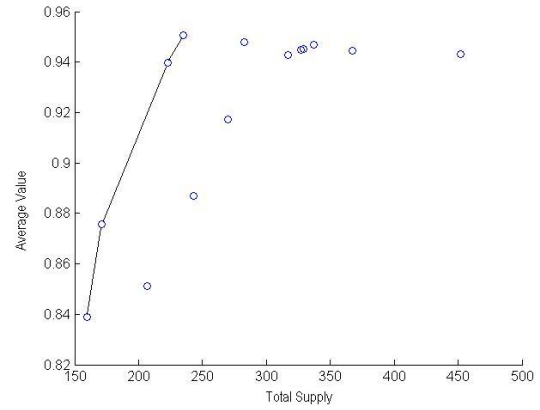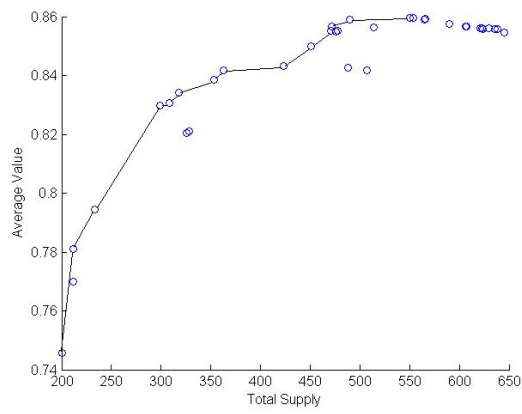
Figure 65. MCO Efficient Frontier
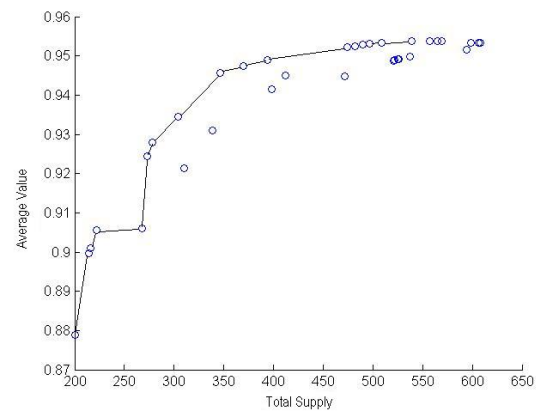
Figure 66. MCO Efficient Frontier

5.6 Conclusions

The importance of the multiobjective average minimum cost flow problem is now clear; it allows any quantitative or qualitative factors to be incorporated into a minimum cost flow structure. We've shown the problem is NP-Complete for general graphs, including the single source single sink problem. For directed acyclic graphs however, pseudo-polynomial heuristics are possible that estimate within 3% of the optimal solution. The question of whether the principles used in solving graphs with cycles for minimum cost flow problems are relevant for average minimum cost flows is still open. Even though the worst case error bounds for any successive average shortest path heuristic is infinitely large in theory, in random graphs both heuristics presented perform satisfactorily. In practice, it may not make sense to simply optimize average value. For instance, in transportation mode selection, more transfer points may result in unexpected damage. The bicriteria average minimum cost flow problem allows the tradeoff of number of arcs and average value. Efficient solutions are generated using the pseudo-polynomial successive average shortest path heuristics for the average minimum cost flow problem. This estimated efficient frontier is a small percentage from the true efficient frontier and gives a decision maker good solutions from which to choose.

Acknowledgements

VI. Further Contributions

The 3 previous chapters contained articles for publication in various journals. This chapter is additional research completed but not yet in publication format. The inventory theory and value focused thinking section contains work suitable for publication in a supply chain journal. The targeted journal for the section on variants of the network flow problem is Operations Research Letters.

6.1 Transportation Mode Selection Hierarchy and Objectives

Obviously, the fundamental objective of any company or organization is to run a successful operation. This entails doing transportation mode selection well. Many articles attempt to do transportation mode selection well by minimizing some cost, maximizing reliability, or choosing a quality shipping company. Another goal is to ensure a product doesn't stock out due to low reliability, but the reason a stock out is dangerous is because demand may be unmet. Why does demand need to be met? If demand is not met, customers will go elsewhere which decreases revenue. So the real reason for maximizing reliability is to maximize revenue. The real objectives of a company are to minimize costs (this is not simply transportation costs), maximize revenue (by meeting demand), and maximize employee satisfaction. In light of this, we now propose a value focused thinking hierarchy for transportation mode selection. The idea behind the use of value focused thinking is to provide logistics decision makers with a defendable analysis for choosing the best transportation mode/carrier to ship products between locations. By accounting for differing preferences, the model is adaptable to both civilian and military transportation. The ideas are kept at the strategic level, the interest is not in modeling day to day operations. Since VFT is being combined with network optimization, certain measures aren't needed in the

value hierarchy. Capacity limitations, infrastructure, and equipment availability are captured with the network optimization model. Characteristics of goods is the driving force behind the value model, each measure changes with the type of good being shipped. Distance is inherent in the network model as well.

Figure 67 includes a number of these measures in an example value hierarchy. Any others simply fall under another means objective.
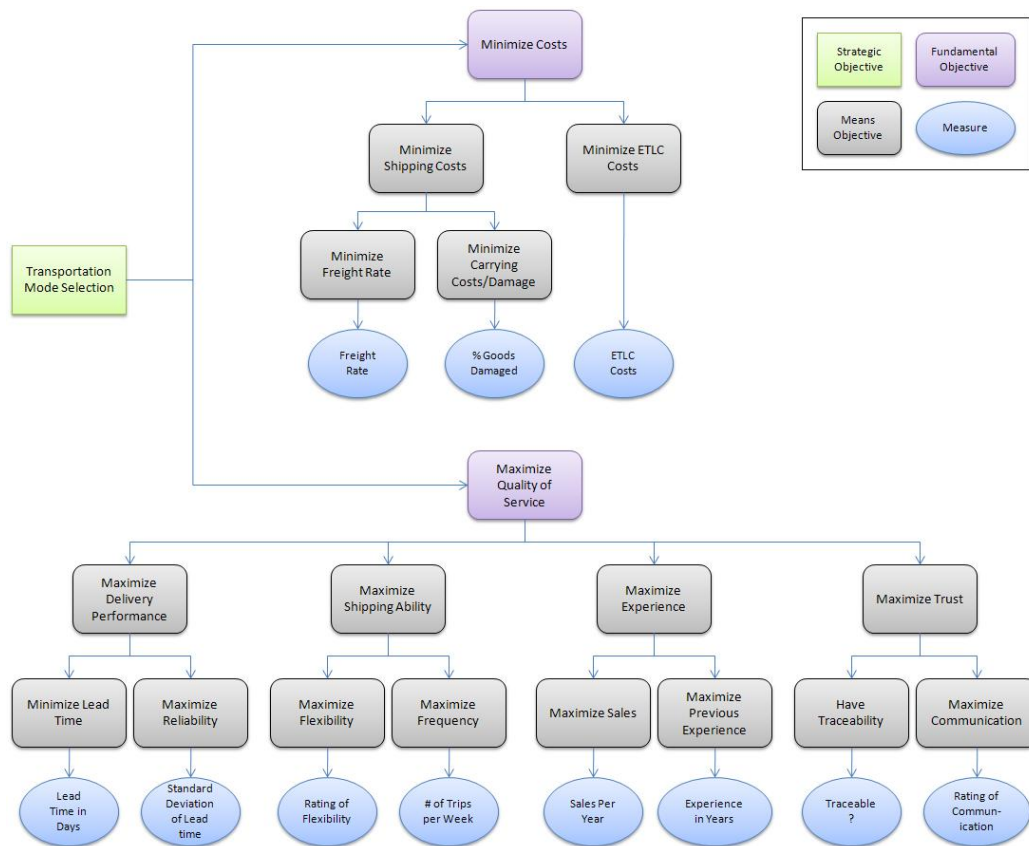
Figure 67. Transportation Mode Selection Value Model

The values from Figure 67 are the arc costs in the shortest path and minimum cost flow problem.

6.2   Variants of the Average Shortest Path and Average Minimum Cost Flow Problem

Several variants naturally flow out of the problems in the previous chapters.

Theorem 8 *The single source single sink average minimum cost flow problem is NP-Complete.*
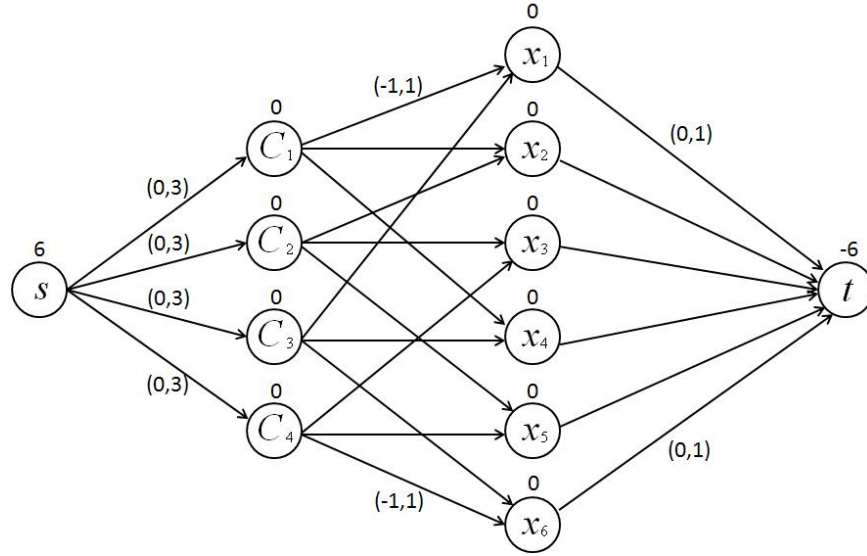
Figure 68. Constructed Instance of the Single Source Single Sink Average Minimum Cost Flow Problem

Proof 9 *This is similar to the proof of Theorem ??, except we define a single sink as follows. Let a single arc emanate from each node in the set X, such that each x_j is connected with node t for $j = 1, 2, ..., n$, and each has a cost of 0 and capacity of 1, as in Figure 68. Obviously, the average minimum cost flow corresponds to an exact cover by three sets just as in Theorem ??.*

Corollary 2 *Any average minimum cost flow problem is strongly NP-Complete even if the arc capacities are all equal to 1.*
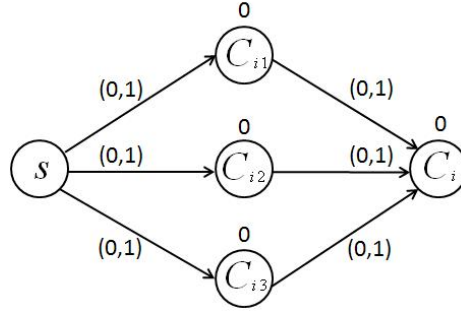
Figure 69. Constructed Instance of the Average Minimum Cost Flow Problem

Proof 10 *In Theorem's* **??** *and 6.2, replace the arcs* (s, C_i) *with 3 arcs of capacity 1 for* $i = 1, 2, ..., m$ *and cost 0 as before. Also add 3m nodes between each* s *and* C_i *as shown in Figure 69. Each arc in Figures* **??** *or 68 now have a capacity of 1, the result follows.*

Theorem 9 *The average minimum cost flow problem with homogeneous positive arc costs is solvable in polynomial time P.*

Proof 11 *It is easy to see this reduces to the maximum flow problem. Any path in the network will have the same average value as any other path. Therefore, the problem reduces to finding any feasible flow through the network satisfying node demands, a special case of the maximum flow problem, which is known to solve in polynomial time [3].*

We may also wish to reduce this to the minimum cost flow problem because no negative cycles exist when arc costs are positive. This ensures we are not taking the longest path through the network even though it will have the same average as the shortest path through the network, which could be desirable even if we are only concerned with the average.

Theorem 10 *The average minimum cost flow problem with homogeneous arc costs for directed acyclic graphs is solvable in polynomial time P.*

137

Proof 12 *Similar to the proof of Theorem 9, each path has the same average value, and therefore the problem reduces to finding a feasible flow satisfying node demands.*

Table 12 summarizes complexity of the average network flow variants.

Table 12. Average Network Path Length Problems

Problem	Complexity	Optimal Algorithm	Heuristic
ASP (DAG)	Polynomial	Orlin's DP Approach(O(nm))	Scaling (O(m))
ASP (Homogenous Arc Costs)	Polynomial	Dyskstra's(O(n^2))	?
ASP (Positive Arc Costs)	NP-Hard	?	?
ASP (General)	NP-Hard	?	?
AMCF (DAG)	NP-Hard	?	Successive Scaling (O(Cm))
			Successive DP (O(Cnm))
AMCF (Positive Arc Costs)	NP-Hard	?	?
AMCF (General)	NP-Hard	?	?
AMCF (Homogenous Arc Costs, DAG)	Polynomial	Successive DP (O(Cnm))	Successive Scaling (O(Cm))
AMCF (Homogenous Arc Costs)	Polynomial	Successive Dykstras (O(Cn^2))	?

VII. Conclusions and Open Questions

7.1 Summary

We've motivated the need for an average network flow problem. Incorporating value focused thinking as a cost function in the shortest path problem results in a mulitobjective average shortest path problem; the same is true for the minimum cost flow problem. Each of these problems were reduced from known NP-Complete problems proving NP-Completeness for general graphs. Furthermore, polynomial time algorithms and heuristics are possible for the average shortest and longest path problem for directed acyclic graphs. The average minimum cost flow problem for directed acyclic graphs is NP-Complete, but accurate efficient solutions are possible using concepts from the algorithms for the average shortest path problem. In both of these problems, a decision maker may need to make tradeoffs between arcs and value. This leads to the bicriteria average shortest path and bicriteria average minimum cost flow problem. These problems are clearly NP-complete as well. However, as is the case for the average shortest path problem, the solutions for the bicriteria average shortest path problem for directed acyclic graphs can be generated efficiently. For the bicriteria average minimum cost flow problem, very accurate estimated efficient solutions are possible using the heuristics for the average minimum cost flow problem.

7.2 Open Questions

Several areas of research could emanate from this study. An obvious follow-on is the development of an algorithm to solve the average shortest path or average minimum cost flow problem for graphs with cycles. Some of the principles from minimum cost flows for cyclic graphs will be useful. Finding an algorithm to solve the average minimum cost flow problem for directed acyclic graphs is possible, however

obviously not a polynomial time algorithm.

Advanced sensitivity analysis of the MALP and the MAMCF to include measure weights of the value function, arc capacities, and number of arcs utilized would be interesting. Each change in the weights of the value function will produce a new efficient frontier. The idea is then to observe where each of the efficient frontiers overlap across the range of weights specified by the decision maker.

The idea of non-additive paths is a relatively recent development in the network optimization literature. Research motivation comes from the fact that not all network paths are additive in nature, that is a path cost may be some function other than simply the addition of all the arcs costs. The methods were sparked by the traffic equilibrium problem and are discussed in [40], [1], and [41]. There aren't many citations to these papers and extensions are possible. One idea is to compare these non-additive algorithms with the algorithms and heuristics from this dissertation. The average path is actually just a special case of the non-additive shortest path discussed in [40] and [1], where the function is simply the sum of arc costs of the path divided by the number of arcs. Because of its simple structure, we've shown in this dissertation that faster solutions are attainable. A study to compare the performance of these non-additive algorithms with our algorithms and heuristics would be beneficial.

Many other extensions potentially exist. Research on several of these is underway and therefore are not exposed here.

Appendix A. 2011 Western Decision Sciences Conference Proceedings

A Value Focused Network Optimization Solution for Transportation Mode Selection

Jeremy D. Jordan, Jeffery D. Weir, and Doral E. Sandlin

Air Force Institute of Technology

2950 Hobson Way, Wright-Patterson AFB, Ohio, 45433,

USA, +1 937-255-3636, jeremy.jordan@afit.edu

jeffery.weir@afit.edu

doral.sandlin@afit.edu

1.1   Abstract

United States Transportation Command's (USTRANSCOM) current approach for transportation mode selection involves countless complicated decisions, few of which are mathematically driven. The proposed methodology combines Value Focused Thinking and network optimization/linear programming to provide a novel strategic multi-objective multi-commodity flow model suitable for use by USTRANSCOM. A general formulation of the network and accompanying linear program along with an example application showing the simplicity of the approach are provided. Because of the structure of linear program, solutions can be obtained efficiently.

1.2 Introduction

With a $120 billion annual budget, logistics in the Department of Defense is overwhelmingly complicated [12]. Transportation costs alone total $10 billion–opportunities for improvement with a budget of this size are countless. The joint command in charge of this budget and responsible for all military transportation during peace and war times is USTRANSCOM. With a direct link to the President and Secretary of Defense, they are an integral part of our nations warfighting capability, and are in need of advanced modeling methods in order to accomplish their mission efficiently and effectively.

Supply chain and logistics in the U.S. military is vitally important to the security of our citizens. It ensures our military force is ready to protect and defend the United States. The following quote sums up the military supply chain, "The DoD supply chain is a global network of DoD and commercial supply, maintenance, and distribution activities that acquires and delivers materiel and logistic services to the joint force globally. Its fundamental goal is to maximize force readiness while optimizing the allocation of limited resources" ([9], pg. I-11). Two important words from this quote are network and optimization. USTRANSCOM's purpose in managing the DOD's supply chain is to build a global network and optimize the allocation of limited resources to ensure force readiness. Yet, how many of USTRANSCOM's decisions are really based on mathematical optimization? Very few we suspect, even though techniques such as network optimization could provide a quintessential modeling tool for this transportation problem [1]. However, many may question the use of optimization with such a large complicated network. For this reason, we introduce value focused thinking (VFT) in addition to network optimization. VFT is capable of aiding complex decisions involving many stakeholders, conflicting objectives, and

uncertainty ([5],[10]).

We limit the discussion to sustainment, however the framework could be easily adapted to capture troop movements and units and equipment as well. Each sustainment item, shipped from supply warehouses and bases in the U.S. to overseas bases, are assigned different priorities from several DoD ranking systems; this a major factor in determining its mode of transportation. These systems include the Force/Activity Designator, Urgency of Need Designator, and the Priority Designator. For more information on these and other priority systems used see ([9], app. B). USTRANSCOM is responsible for setting all rules, regulations and policies governing the DoD supply chain. These rules then dictate how the military executes transportation mode selection. They are broken into four sections: Surface Deployment and Distribution Command (SDDC) is responsible for ground transportation rules and regulations; Military Sealift Command (MSC) governs all ocean transportation policies; AMC's Tanker Airlift Control Center (TACC) schedules and dictates military airlift policies; and the commercial industry comprised of the Civilian Reserve Air Fleet (CRAF) program with such shippers as UPS supplement military capabilities.

Using the priority rules and policies of USTRANSCOM, the four services setup a transportation scheme through a physical network consisting of roads, railroads, storage facilities, warehouses, ports, waterways, and pipelines. This includes capabilities of military organizations and those of commercial partners as well as those of multinational and interagency participants ([9], pp. II-1). This network setup and priority scheme are the driving force behind the transportation mode selected for each product. A model that incorporates these priorities, existing infrastructure, and existing modes of transportation could be very useful. The proposed approach

accomplishes this by looking at the value of shipping a product on each type of transportation mode and assists in choosing the most valuable method of transporting a product from one location to another. It uses optimization to maximize value to the military while ensuring capacities and demand are met.

A review of the literature shows that articles on transportation mode selection in the military are scarce. One thesis exists that looks at a small part of the transportation process, from port in the occupied country to foxhole for army sustainment goods [7]. In our review of the supply chain and operations management literature, the use of value focused thinking for transportation mode selection in both civilian and military situations has not been documented. Additionally, the multi-objective multi-commodity flow problem developed in general methodology section is innovative, mathematically consistent, and practically useful.

1.3 Value Focused Thinking and Transportation Mode Selection

Numerous approaches have been used for transportation mode selection and therefore measures of effectiveness are well defined in the logistics community. We know from the literature [3] the 15 most important factors used for civilian transportation mode selection, the top 4 being cost, speed, reliability, and product characteristics. Cost, speed, and reliability are self explanatory, product characteristics in the military problem turns out to be the priorities discussed above. [8] lays out the measures used in civilian transportation mode selection, we argue these are the same factors used in military transportation mode selection however with differences in importance. Rather than choosing a carrier and mode based solely on speed and reliability, decision makers regularly consider cost, reputation, and other factors also noted in [2]. How can all these factors be captured? Value-focused thinking provides a model

to capture both tangible and intangible factors.

Value focused thinking (VFT) allows tradeoffs to be made between competing objectives. It allows, for example, a decision maker to trade off increases in cost for increases in speed and reliability. For instance, a decision maker may decrease the importance of cost while increasing the importance of speed and reliability for a high priority product. A low priority product with a specified deadline may see a decrease in the importance of speed while the importance of reliability and cost increase.

Value-Focused Thinking and Decision analysis seek to aid in human decision making, not model the human decision making process. The justification for this purpose is decision makers should desire to make rational choices given any situation. What is the purpose of modeling inconsistent or irrational behavior? A better approach is to build models of rational choice, and let the decision maker utilize the models in their decision making process. Alternative focused thinking techniques, such as the Analytic Hierarchy Process (AHP), encourage a "best" choice among the available alternatives; value focused thinking begins with the fundamental inputs in a decision and reveals what is truly valued. Rather than starting with alternatives, VFT starts with objectives and measures. Alternatives can then be generated from these measures and assigned values based on their fulfillment of the objectives. In this way, "value gaps" between a best available alternative and an ideal alternative can be identified, providing a decision maker with a more complete analysis of the problem at hand.

As with any process, a list of suggested implementation procedures is useful as well. [6] and [5] describe the steps in the VFT process rather generally, a more spe-

cific declaration of events is given by [11]. He describes the value modeling process in 10 essential steps. Problem identification (Step 1) is self explanatory, transportation mode selection. Creating the value hierarchy (Step 2) and developing measures (Step 3) are covered in the transportation mode selection literature. Creating the value function (Step 4) and weighting the hierarchy (Step 5) require time from subject matter experts. Following model construction is the generation (Step 6) and scoring (Step 7) of alternatives through deterministic analysis (Step 8). Because of the subjectiveness in defining values and weights, a proper sensitivity analysis (Step 9) is essential for a good analysis. Finally, conclusions and recommendations (Step 10) relate the findings back to the decision maker. See the references above for more detailed information.

1.4   Problem Formulation

With measures and objectives for transportation mode selection already defined, a value function can be solicited from the decision maker. Single dimension value functions (SDVF) are normally used to capture preferences for varying levels of the measures, assigning each level of the function a value between 0 to 1 . For an in depth look at values and preference functions see Keeney and Raiffa [4]. For the purposes of this paper, we simply need to know that a value function $v_k x_k$ is easily captured from a decision maker. Weights can be determined using either AHP, swing weights, direct assessment, or group weights. Direct assessment is the most convenient; it's accomplished through examining the measures from an objective and weighing trade-offs between them. The least important measure is assigned a 1, the remainder of the measures are assigned numbers based on how much more important they are than the least important measure. A measure that is twice as important as the least important measure is given a value of 2, call it r_k for measure k, and so on. Since the weights

146

must sum to one, that is $\sum w_k = 1$, these values must be scaled to a decimal between 0 and 1. To determine a weight w_k for measure k, take its successive r_k and divide by the sum of all the ranks, that is

$$w_k = \frac{r_k}{\sum r_k}.$$ (38)

The form of the multi-objective value function then becomes

$$v(x) = \sum w_k v_k(x_k)$$ (39)

for $k = 1, 2, ..., n$ measures, where $v_k(x_k)$ is the single dimension value function of measure k, w_k is the weight of measure k, and $v(x)$ is the multiobjective value. Figure 70 shows an example network formulation of a transportation problem.
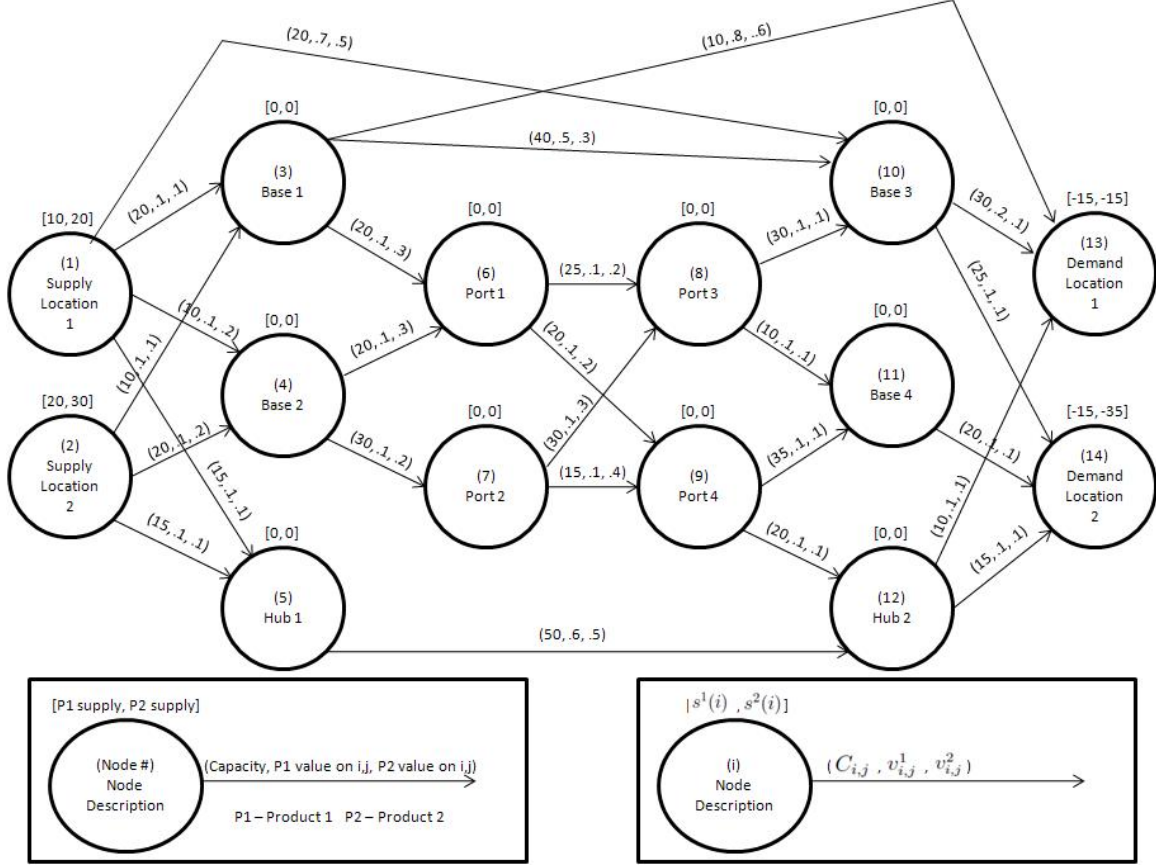
Figure 70. Example Network Formulation

We set the problem up in the context of a transportation network for USTRANSCOM. Consider a graph $G(N, E)$ as a directed network. The nodes $i \in N$ could represent various locations available to USTRANSCOM for shipment through, to, or from, including bases, ports, airports, and rail stations. The edges $i, j \in E$ are the connections between these nodes. These may be connected in either direction. The value function $v_{i,j}^p = w_k^p v_{i,j,k}^p$ defines the value $(v_{i,j}^p)$ of shipping a product p on the transportation mode from node i to j for measure k. $x_{i,j}^p$ defines the number of pounds of product p shipped between node i and j per year. Given this graph $G(N, E)$, the general minimum cost network flow problem is

148

$$\min z = \sum_{(i,j)\in E} c_{i,j}, x_{i,j}$$

$$s.t. \sum_{(i,j)\in E} x_{i,j} - \sum_{(j,i)\in E} x_{i,j} = s(i) \ \ \forall \ \ i \in N$$

$$l_{i,j} \le x_{i,j} \le u_{i,j} \ \ \forall \ \ (i,j) \in E$$

$$(40)$$

where $x_{i,j}$ is the flow from vertex i to j, $c_{i,j}$ is cost of a unit of flow from i to j, $l_{i,j}$ is the lower bound on the flow from i to j, $u_{i,j}$ is the upper bound on the flow from i to j, and $s(i)$ is the supply or demand at node i.

The value function weight w_k^p of measure k is dependent on product p for each of the legs i, j. Higher priority products will put a lower weight on cost and higher weight on speed and reliability where a lower priority product weights cost much higher, and speed and reliability lower. The cost value functions $v_{i,j,k}^p$ will also be variable for each product p, with higher priority items valuing lower costs less, and speed and reliability higher. The capacity of each node is $C_{i,j}$ while the supply or demand in and out of node is denoted $s_p(i)$. These are discussed further in the next section where a linear program to solve the network in Figure 70 is given.

1.5 General Methodology

This section develops a general methodology synthesizing value focused thinking and the minimum cost network optimization problem. The minimum cost problem now transforms into a maximum value network optimization problem, where the decision maker is maximizing value while meeting demand and staying within capacity.

This formulation can be used to solve the network in Figure 70. Combining the value function with the general network formulation and accounting for multicommodities between nodes gives the following linear program:

$$\max z = \sum_{p=1}^{m} \sum_{(i,j) \in E} \left(\sum_{k=1}^{l} w_k^p v_{i,j,k}^p \right) x_{i,j}^p$$

$$s.t. \sum_{(i,j) \in E} x_{i,j}^p - \sum_{(j,i) \in E} x_{i,j}^p = s^p(i) \quad \forall \ i \in N, \ p = 1, ..., m$$

$$\sum_{p=1}^{m} \sum_{(i,j) \in E} x_{i,j}^p \leq C_{i,j} \quad \forall \ i, j \in E$$

$$x_{i,j}^p \geq 0 \quad \forall \ i, j \in E$$

$$(41)$$

where w_k^p is the weight of measure k for product p, $v_{i,j,k}^p$ is the value function defined for product p at measure k between nodes i and j, $s^p(i)$ is the supply or demand of product p entering or leaving the node i, and $C_{i,j}$ denotes the capacity of the transportation mode between node i and j. Note, when viewing the problem from a strategic lens, the capacities and supply/demand represent the yearly volume. Solving the network in Figure 70 with the linear program in Equation 52 is computationally efficient as well since the network flow problem is known to be totally unimodular when formulated as in Equation 40 and 52.

1.6 Conclusions

USTRANSCOM's current process for transportation mode selection can be improved by using this approach. The proposed methodology is both robust and novel, creating a mathematically defendable model for transporting military sustainment

goods. Although demonstrated at the strategic level in this paper, the methodology could easily adapt to assist with decisions at the tactical level, as well as capture any other constraints that may be needed (ie. shipping time, costs, etc). Additionally, the value function can capture any other variables deemed important to the decision. In summary, any factor or thought process influential in the military transportation mode selection decision could be incorporated into this model, and solutions, even with large transportation networks, are computationally tractable.

1.7   References

[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network flows. Upper Saddle River, NJ: Prentice Hall., 1st edition, 1993.

[2] K. Cullinane and N. Toy. Identifying influential attributes in freight route/-mode choice decisions: a content analysis. Transportation Research Part E, 36:4153, 2000.

[3] Kathryn Dobie. Core shipper concept. Transportation Journal, 44(2):3753, 2005.

[4] R. L. Keeney and H. Raiffa. Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Wiley, New York, NY, 1976.

[5] Ralph L. Keeney. Value-Focused Thinking: A Path to Creative Decisionmaking. Harvard University Press, Cambridge, MA, 1992.

[6] Craig W. Kirkwood. Strategic Decision Making. Duxbury Press, Belmont, CA, 1996.

[7] Seongkyun Lee. Onward movement transportation mode selection study. Masters thesis, Air Force Institute of Technology, 2003.

[8] M.J. Liberatore and T. Miller. A decision support approach for transport

carrier and mode selection. Journal of Business Logistics, 16(2):85115, 1995.

[9] Joint Chiefs of Staff. Joint Publication 4-09, Distribution Operations, 05 February 2010.

[10] Gregory S. Parnell. Methods for Conducting Military Operational Analysis: Best Practices in Use Throughout the Department of Defense, chapter 19, Value-Focused Thinking Using Multiple Objective Decision Analysis. Military Operations Research Society, 2007.

[11] Mark Shoviak. Decision analysis methodology to evaluate integrated solid waste management alternatives for a remote alaskan air station. Masters thesis, Air Force Institute of Technology, 2001.

[12] USTRANSCOM. Mission brief. as of 1 December 2009.

Multiobjective Decision Programming for the Multiobjective Minimum Cost Flow Problem

Jeremy D. Jordan, Jeffery D. Weir, and Doral E. Sandlin

Air Force Institute of Technology

2950 Hobson Way, Wright-Patterson AFB, Ohio, 45433,

USA, +1 937-255-3636, jeremy.jordan@afit.edu

jeffery.weir@afit.edu

doral.sandlin@afit.edu

Abstract

Transportation mode selection decisions are becoming increasingly difficult as supply chains expand and the transportation domain adapts. In order to handle these growing complexities, we combine network optimization, multiobjective programming, and multiobjective decision analysis. Specifically, we define Multiobjective decision programming from multiobjective programming and multiobjective decision analysis and apply it to the multiobjective minimum cost network flow problem. The result is a more accurate representation of the transportation mode selection decision environment, and a unique methodology applicable to a variety of problems.

Keywords

Multiobjective Programming, Optimization, Decision Analysis, Transportation Mode Selection

1. Introduction

Interest in transportation mode selection is increasing as the world becomes more interconnected and supply chains expand globally. Operations research techniques are becoming more popular in logistics, yet much work is still needed. Modeling tradeoffs between cost, speed, and reliability functions is especially important and is typically accomplished through Multiobjective programming (MOP), but other intangible factors exist as well. These intangible factors, such as perceived quality of customer service, shipment tracking capabilities, and long-term partnership potential [1] for instance, are best captured using Multiobjective decision analysis (MODA), specifically utility or value functions. The general methodology developed hereafter was motivated by attempts to model the transportation mode selection problem in its entirety.

In this paper, we merge MOP and MODA and define Multiobjective Decision Programming (MODP). The concept of combining a multiobjective programming problem with an additive utility or value function is not new [2-5], we are simply offering a formal definition of a methodology used sparingly in the literature in hopes of spurring future research in this fertile area. In addition to a formal definition, the literature is missing the application of value focused thinking (VFT) [6-8] as a means of soliciting the required information from the decision maker. VFT assists the decision maker in uncovering the value each alternative provides through determining what the decision makers values in the decision. This is accomplished through a logical breakdown of the problem into objectives and measures. We provide the VFT model. Finally, we apply MODP to a network optimization problem and create

154

a formulation not yet discussed in the literature.

Due to the inherent structure present in transportation problems, a minimum cost flow (MCF) optimization model best captures the dynamics of the system. Accounting for multiple objectives in the MCF problem allows us to capture the conflicting criteria present in transportation planning problems, we denote the multiple objective minimum cost flow problem MMCF. Combining this with MOP and MODA creates a unique methodology we call Multiobjective decision programming (MODP) for the multiobjective minimum cost flow (MMCF) problem, filling a gap in the literature. Hamacher et al [47] present a review of the MMCF including theory and algorithms for solving; their all inclusive reference list doesn't include any papers formulating this problem. Furthermore, very few authors have combined MOP and MCF, although transportation planning frequently requires multiple objective functions such as minimizing cost, minimizing arrival time, minimizing deterioration of goods, and maximizing safety [47].

Applying MODP to the MMCF allows us to convert multiple linear objective functions into a single objective function, capturing any qualitative and quantitative factors or objective functions needed, while maximizing value to the decision maker in a network environment. As with typical multiobjective programming, determining the efficient frontier allows a decision maker to examine tradeoffs and make good choices among pareto optimal solutions. However, it becomes difficult to choose which solution is best within this efficient frontier, this is where MODA comes in. Of particular interest to decision makers is the value of their decisions. How much value does a decision contribute based on their specific utility or value function? The idea is then to determine the value each of the objective functions is contributing to the

155

overall decision. This in turn shrinks the size of our efficient frontier to consider only tradeoffs that add value to the decision. The next section gives a background and further motivation for the problem. Section 3 defines our methodology and Section 4 discusses implications of the proposed application to the transportation mode selection problem.

2. Background

The transportation problem was initially formulated by Hitchcock in 1941 [10]. The transportation problem is simply a special case of the minimum cost network flow problem where goods are shipped only from a finite number of origins to some number of destinations. This is not an accurate representation of today's transportation environment. The minimum cost network flow problem, on the other hand, allows goods to flow through intermediate nodes. In addition, most problems involve multiple competing objectives, see [47] for a summary of the multiobjective minimum cost network flow literature. Research in the area of Multiobjective programming, in general, flourished in the 70's and 80's, a few papers use additive value models to assign preferences to objective functions, see [2] for an example. An excellent resource for research on network optimization for transportation in general is given in [11]. The rest of this section presents background from each of the methodology's being combined.

2.1 Multiobjective Programming

Multiobjective programming is a popular operations research tool and several good textbooks exist, see [35, 13] for MOP theory and further detail of the material presented in this section. Generally, when solving multiobjective programming (or multicriteria optimization) problems, unique optimal solutions are unattainable. In-

stead, a decision maker must choose between multiple optimal solutions. These pareto optimal or efficient solutions occur when an increase in one objective function yields a certain decrease in another objective function. Recall the general multiobjective program:

$$\min(f_1(x), ..., f_k(x))$$

$$\text{st} \ \ g_c(x) \leq 0$$

(42)

for $c = 1, 2, ..., m$ constraints and k objective functions.

Definition 7 *A feasible solution $\hat{x} \in X$ is called efficient or pareto optimal if there is no other $x \in X$ such that $f(x) \leq f(\hat{x})$. If $\hat{x}$ is efficient, $f(\hat{x})$ is called a nondominated point. The set of all efficient solutions $\hat{x} \in X$ is denoted X_E and is called the efficient set.*

Since a decision maker must ultimately choose one solution to the problem, the research in MOP is primarily concerned with generating the efficient frontier X_E and choosing the best solutions from X_E. This can be done in a number of ways. Scalarization techniques are one such method that generate good solutions by: assigning some weight λ_k to each function in formulation 42(Weighted sum method); converting or relaxing the objective functions into constraints (epsilon constraint method, elastic constraint method, Benson's method); or by minimizing distances from some desired reference point (Compromise programming, Achievement function method). In each of these, the idea is to get as close as possible to the decision makers' preferred decision without knowledge of his preference function. Why? The argument from Steuer [13] and others is the decision makers preference function is rarely known.

If it was, there would be no need for MCO techniques. Wouldn't a better method be to attempt to collect the function, and spend time with the sensitivity analysis? The next section presents the additive value function of value-focused thinking, an ideal method for gathering a decision makers preference function.

2.2 Multiobjective Decision Analysis

Multiobjective Decision Analysis (MODA) is referred to as "a set of quantitative methods for analyzing decisions based on the axioms of consistent choice" [14]. That is, DA is used to make rational decisions. Value Focused Thinking is one such technique, it's the process of understanding what is valued in a decision. As Keeney states, "Values are what we care about. As such, value should be the driving force for our decision making" [7]. In order to do this, we need to know the measures important to the decision maker, how the decision maker values levels of the measures, and the weights of each of these measures on the overall decision.

The general additive value model is given by:

$$v_j(x) = \sum_{i=1,j=1}^{n,m} w_i v_i(x_{ij}) \tag{43}$$

for $i = 1, 2, ..., n$ measures and $j = 1, 2, ..., m$ alternatives, where $v_i(x_{ij})$ is the single dimension value function of measure i evaluated for alternative j, w_i is the weight of measure i, and $v_j(x)$ is the multiobjective value for alternative j. The objective is to maximize value over the j alternatives. The value of a particular alternative j, or $v_j(x)$, is simply the value of an alternative in relation to what is important to the decision. A $v_j(x) = 1$ is indicative of an alternative that achieves everything desired in a decision, whereas $v_j(x) = 0$ indicates an alternative achieving no value in the

decision. What VFT contributes is the value of an alternative, both in relation to what is valued in the decision and in relation to the other alternatives.

Several methods can be used to determine w_i for each of the measures, direct assessment being the most straightforward. More involved decision maker input is needed for $v_i(x_{ij})$. The exponential and piecewise linear SDVF's are equipped to handle preferences on a continuous or discrete scale and are easy to explain to a decision maker. Initially, both high and low levels are needed and given a value of 1 and 0, respectively. For instance, in transportation mode selection lower lead times are normally desired. It may be the case that that 4 days has the same value as 3 days or 2 days. In this case, any lead time less than 4 days receives a value of 1, increased performance in lead time is not beneficial.

For a monotonically increasing measure, the continuous exponential function is

$$
v_i(x_i) = \begin{cases} \frac{1 - e^{-(x_i - x_L)/\rho}}{1 - e^{-(x_H - x_L)/\rho}} & \rho \neq \text{Infinity}; \\ \frac{x_i - x_L}{x_H - x_L}, & \text{otherwise}. \end{cases} \tag{44}
$$

where x_H is the most preferred level (assigned a value of 1), x_L is the least preferred level (assigned a value of 0), x_i is the level of the ith measure, ρ is an unknown parameter, and $v_i(x_i)$ is the value of the ith measure at level x_i.

VFT is a powerful methodology able to model rational decisions, yet several assumptions accompany the application of the additive value model, see chapter nine of [8]. Ignoring these assumptions can lead to a poor model, see [15] for common pitfalls in value modeling. With the MOP and MODA methodology defined, we transition to network optimization in the next section.

2.3 Multiobjective Minimum Cost Flow

Network optimization is a well established OR methodology and is applicable to a variety of problems, see [16] for theory of networks. One highly studied network model is the minimum cost flow problems, a connected directed graph. The MCF is directed in that flow is directed from one node to another node and connected because a path exists between every node. The MCF allows a cost to be associated with each arc, and bounds to be set on the flow between nodes. By minimizing cost, the MCF prevents cycles by shipping goods through the network via the minimum cost route. Because of these reasons and its competing objectives, the transportation environment is best modeled through an MCF.

Consider a graph $G(N, E)$ as a directed network. The nodes $i \in N$ could represent various locations available for shipment through, to, or from, including bases, ports, airports, and rail stations. The edges $i, j \in E$ are the connections between these nodes. These may be connected in either direction. Given this graph $G(N, E)$, the general minimum cost network flow (MCF) problem is

$$\min z = \sum_{(i,j) \in E} c_{i,j} x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = s(i) \ \ \forall \ \ i \in N$$

$$l_{i,j} \leq x_{i,j} \leq u_{i,j} \ \ \forall \ \ (i,j) \in E$$

$$(45)$$

where $x_{i,j}$ is the flow from vertex i to j, $c_{i,j}$ is the cost of a unit of flow from i

160

to j, $l_{i,j}$ is the lower bound on the flow from i to j, $u_{i,j}$ is the upper bound on the flow from i to j, and $s(i)$ is the supply or demand at node i. Converting this to a multiple criteria optimization problem gives the multiple objective minimum cost flow (MMCF) problem

$$\min z_1 = \sum_{(i,j) \in E} c^1_{i,j} x_{i,j}$$

$$\min z_2 = \sum_{(i,j) \in E} c^2_{i,j} x_{i,j}$$

$$\vdots$$

$$\min z_p = \sum_{(i,j) \in E} c^p_{i,j} x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = s(i) \ \ \forall \ \ i \in N$$

$$l_{i,j} \leq x_{i,j} \leq u_{i,j} \ \ \forall \ \ (i,j) \in E$$

$$(46)$$

for $k = 1, 2, ..., p$ objective functions. The idea with Formulation 46 is to search for efficient solutions within the context of the minimum cost network flow problem. The question becomes what is the best method to obtain $c^k_{i,j}$ for each of the objectives? Consequently, the additive value model in Equation 50 allows the decision maker to make tradeoffs between objectives and solutions, and define the $c_{i,j}$ based on objectives valued in the decision. This contribution is motivated in the next section.

2.4 Motivation for the Methodology

As alluded to earlier, we use the additive value model from VFT as a scalarization technique for the MOP, different from those discussed in Section 2.1. MODP is unique

in that each objective function remains an objective function, unlike the constraint methods, but the entire efficient frontier need not be generated, as in the weighted sum method. MODP is similar to the reference point methods because solutions are compared to an ideal point, however the MODP ideal point is related to the decision maker's value function and may actually be achievable unlike the ideal point of the MOP. It may deviate from the efficient frontier of the MOP as well. Research in this area began with Keeney and Raiffa [6] and is summarized in [2], but has spread into several other literature streams.

Multicriteria Decision Making (MCDM) and Multiattribute Utility Theory (MAUT) are two neighboring disciplines to MODA and MOP. DA is distinct from these in that it "is normative, rather than descriptive. That is, it provides a systematic quantitative approach to making better decisions, rather than a description of how unaided decisions are made"[14]. Nevertheless, MCDM and MAUT contribute important research in the area of decision maker preferences and multiobjective optimization. The majority of the research with decision makers deals with interactive algorithms, that is, defining the efficient frontier and working with the decision maker to determine their preferred efficient solution. In fact, in Wallenius et al [17], the authors state "In multiple criteria optimization, there is usually no attempt to capture the decision maker's utility or value function mathematically. Instead, the philosophy is to iteratively elicit and use implicit information about the decision maker's preferences to help steer the decision maker to her or his most preferred solution". This implies MODA is not a widely accepted technique used with multicriteria optimization, our argument is it should be. MODA appears to be the best approach for strategic decisions such as those faced in the logistics literature.

According to Wallenious [17], the philosophy of multicriteria optimization is to model the decision makers preferences. This objective is different than MODA. MODA attempts to model what a decision maker hopes to achieve by making the decision, and shows how well potential solutions meet their objectives. Attempting to capture a decision makers value function and learning what the decision maker values is just as important as defining the actual value function. After a reasonable value function is decided on, sensitivity analysis can be used to find an acceptable range. This bounds the efficient frontier. Most decision makers can provide a range on the w_i's and v_i's in Equation 50.

Hamacher's review paper on the multiple objective minimum cost flow problem [47] discusses exact and approximate algorithms to solve both the continuous and integer MMCF. In the paper, they state "We found no papers on an exact solution method for MMCF with more than two objectives". The methodology in the next section does just that.

3. Methodology Development

We develop the general multiobjective decision programming (MODP) formulation and then show a specific application of MODP to the multicriteria minimum cost flow (MMCF) problem. The MODP formulation is a general union of multiobjective programming (MOP) and multiobjective decision analysis (MODA). Obviously, multiobjective linear programming (MOLP) is a special case of MOP, so we would call MODP for linear programming multiobjective linear decision programming MOLDP. Since the MMCF is a special case of the MOLP, the application would actually fall under MOLDP, but we stick with the term MODP to limit the new terms presented.

3.1 MODP

As discussed in Section 2.4, VFT can be used as a scalarization technique that measures the value each of the objective functions contributes to the overall decision in a multicriteria optimization problem. So using the objective functions from the MOP in Equation 42 as measures in the additive value model from Equation 50 gives a new VFT formulation:

$$v_j(x) = \sum_{j=1,k=1}^{m,p} w_k v_k(f_k(x_{k,j})), \tag{47}$$

for $k = 1, 2, ..., p$ objective functions where we are seeking to maximize value over all j alternatives. Thus, $v_j(x)$ is the value of the jth alternative, and the highest valued alternative is preferred. $f_k(x_{k,j})$ can be a linear, non-linear, or integer objective function. This is the beauty of the additive value model, all functions are transformed into one function scaled between 0 and 1. The objective functions now have the form:

$$\max w_1 v_1(f_1(x))$$

$$\max w_2 v_2(f_2(x))$$

$$\vdots$$

$$\max w_p v_p(f_p(x))$$

Thus the multiobjective decision programming problem (MODP) now becomes

$$\max v_j(x) = \sum_{n,m,p}^{i,j,k} w_i v_i(f_k(x))$$

$$s.t. g_c(x) \le 0$$

$$\tag{48}$$

for $k = 1, 2, ..., p$ objective functions, $i = 1, 2, ..., n$ measures, $j = 1, 2, ..., m$ alter-

natives, and $c = 1, 2, ..., l$ constraints. We seek to maximize value over all objective functions, measures, and alternatives while satisfying the constraints.

Reiterating, this will tell us how good our solution to the multicriteria optimization problem really is, how much value it's obtaining. It may be the case that the *optimal* solution to MODP is not a *good* solution. Because LP's are prescriptive in nature, they sometimes yield solutions that are impractical from an implementation standpoint as well. This is where the VFT model comes in. The value model is normative in that it provides an ideal solution by which alternatives can be compared. Each pareto optimal solution is gauged against the decision makers values, and the best efficient solutions are realized. This general formulation is capable of handling non-linear, linear, or integer objective functions, which could consist of one or more variables. Solving Equation 48 allows the definition of valuable alternatives within the efficient frontier. Next, MODP is applied to the MMCF, which is a special case of the MOLP and MOP.

3.2 MODP applied to MMCF

Applying MODP to the MMCF problem in Formulation 46 by replacing the constant $c_{i,j}^k$ with values v_k of functions $f_k(*)$, where $f_k(*)$ is the kth cost function of a unit of flow from i to j, gives

$$\min z_1 = -w_1 \sum_{(i,j) \in E} v_1\left(f_1(*)\right) x_{i,j}$$

$$\min z_2 = -w_2 \sum_{(i,j) \in E} v_2\left(f_2(*)\right) x_{i,j}$$

$$\vdots$$

$$\min z_p = -w_p \sum_{(i,j) \in E} v_p\left(f_p(*)\right) x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = s(i) \quad \forall \ i \in N$$

$$l_{i,j} \leq x_{i,j} \leq u_{i,j} \quad \forall \ (i,j) \in E$$

$$b_k \leq f_k \leq h_k \quad \forall \ (i,j) \in E$$

$$(49)$$

for $k = 1, 2, ..., p$ objective functions and where $x_{i,j}$ is the flow from vertex i to j, $l_{i,j}$ is the lower bound on the flow from i to j, $u_{i,j}$ is the upper bound on the flow from i to j, $s(i)$ is the supply or demand at node i, b^k is the lower bound on f_k for the kth objective function from node i to j, and h_k is the upper bound on f_k for the kth objective function from node i to j. b_k and h_k will depend on the variable settings of f_k available at each arc for each function. Reiterating, the negative value is being minimized, which is essentially maximizing value in the same way VFT currently operates. The objective function maximizes value over each of the objective functions for each of the arcs. For example, 49 may minimize cost, maximize speed, and maximize reliability for the sum of all arcs. The value each of these objectives contributes to the decision is maximized, while the arcs capacities and supply and demand are satisfied.

Since the idea is to maximize value, the multiobjective minimum cost flow problem is to minimize negative value. In practice, the functions f_k could be linear, non-linear, or integer. Because the MMCF seeks to minimize costs over the entire network, trade-offs between these functions are calculated prior to the calculation of $x_{i,j}$. Therefore, even if f_k are non-linear, the linear MMCF structure holds and standard linear programming can be used to solve Formulation 49. The $x_{i,j}$ simply changes the function by some magnitude.

Since the MMCF in Formulation 49 is a special case of the MOLP, we know the solution is an efficient solution from the following theorem [35],

Theorem 11 *Let $\hat{x} \in X$ be an optimal solution of the weighted sum LP.*

1. *If $\lambda \geq 0$ then $\hat{x}$ is weakly efficient.*

2. *If $\lambda > 0$ then $\hat{x}$ is efficient.*

Therefore, any value function with a weight $w_i \geq 0$ yields an efficient solution. However, there may be several ways to achieve this optimal solution. Solving Formulation 49 may result in several values of $x_{i,j}$ yielding the same overall value v_j. Formulation 49 is applicable to a variety of network problems, the next section provides details of its application to the transportation mode selection problem.

4 Discussion of the Application to Transportation Mode Selection

Many multiobjective programming articles have minimized cost or both cost and time [18, 19] for the MMCF problem. However, this doesn't consider other factors used in transportation mode selection such as reliability, company quality, and other intangible factors. This is why we've proposed the use of value-focused thinking with

multicriteria optimization. Both qualitative factors and quantitative factors with linear, non-linear, or integer objective functions can be accounted for. In their 2008 review of the transportation mode choice literature, Meixall and Norbis [20] discuss the need for a normative decision making model, MODP for MMCF provides this. Since the MODP for MMCF reduces the number of objective functions to 1, the complexity of solving the multicriteria optimization problem is reduced as well. Rather than choosing one of an infinite number of solutions on the efficient frontier, the decision maker is able to measure the solutions against his best valued option. In this way, several good solutions can be extracted from the MOP efficient frontier, or possibly from outside the MOP efficient frontier. The MODP efficient frontier may be completely different than the MOP efficient frontier.

Defining the variables in Formulation 49 is straightforward. From the literature, the measures affecting the decision makers transportation choice are well known, see Dobie or Cullinane and Toy [21, 22]. The approximate weights w_i of cost, lead time, variability, etc. are known as well, however each company's actual weights will vary slightly. The value functions v_k of the functions f_k are dependent on the decision maker, the goods being shipped, and the goals of the company. These are captured using the model in Section 2.2 and techniques from [7, 8]. The flow, $x_{i,j}$, could represent either pounds or twenty-foot equivalent units (TEU) shipped.

It can be shown that if the supply $s(i)$ and the flow limits $l_{i,j}$, $u_{i,j}$ are integer values, the extreme points of the polyhedra defining the objective space will be an integer vector, meaning $x_{i,j}$ will be integer. For transportation, this means the optimal solution is in TEU's shipped via each arc. If there is a need to define fractional values, this simply results in a less than container load shipment. Upper and lower

limits on $x_{i,j}$ and f_k come from available transportation options on each arc i, j of the network and are dependent on the modes, times, and capacity available. As with any programming problem, a proper sensitivity analysis on the weights given by the value model ensures the implemented solutions are as robust as possible.

5. Conclusions

Modeling the transportation mode selection decision process is a challenging problem. There now exists a methodology for accurately modeling this important decision, multiobjective decision programming for the multiobjective minimum cost flow network. MODP combines multiobjective programming and multiobjective decision analysis to provide a flexible methodological approach to solving the multiobjective optimization problem. Applying multiobjective decision programming to the multiobjective minimum cost flow network problem provides a unique methodology to make tradeoffs among competing objectives in a network environment. Initially, utilizing VFT methodology, a decision maker defines the objective functions most important to the transportation selection decision. Next, he models the transportation environment using the minimum cost network flow problem. Finally, using the multiobjective minimum cost network flow problem and multiobjective decision programming, a solution is generated. This solution is compared against what is truly valued in the decision, and the overall value of the solution is calculated. This paper fills a gap in the multiobjective programming and multiobjective minimum cost flow literature and gives the transportation selection decision maker a powerful decision support tool.

Bibliography

[1] Liberatore, M.J. and Miller, T., 1995, A decision support approach for transport carrier and mode selection, Journal of Business Logistics, 16(2), 85-115.

[2] Evans, G.W., 1984, An Overview of Techniques for Solving Multiobjective Mathematical Programs, Management Science, 30(11), 1268-1282.

[3] Ruiz, F., Luque, M., Miguel, F., and Mar Munoz, M., 2008, An additive achievement scalarizing function for multiobjective programming problems, European Journal of Operational Research, 188, 683-694.

[4] Luque, M., Miettinen, K., Eskelinen, P., and Ruiz, F., 2009, Incorporating preference information in interactive reference point methods for multiobjective optimization, Omega, 37, 450-462.

[5] Ehrgott, M., Klamroth, K., and Schwehm, C., 2004, An MCDM Approach to Portfolio Optimization, European Journal of Operational Research, 155, 752-770.

[6] Keeney, R. L. and Raiffa, H., 1976, Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Wiley, New York, NY.

[7] Keeney, R. L., 1992, Value-Focused Thinking: A Path to Creative Decisionmaking. Harvard University Press, Cambridge, MA.

[8] Kirkwood, C.W., 1996, Strategic Decision Making. Duxbury Press, Belmont, CA.

[9] Hamacher, H.W., Pederson, C.R., and Ruzika, S., 2007, Multiple Objective Minimum Cost Flow Problems: A Review, European Journal of Operational Research, 176, 1404-1422.

[10] Hitchcock, F.L., 1941, The distribution of a product from several sources to numerous localities, Journal of Mathematical Physics, 20:224-230.

[11] Crainic, T.G., and Kin, K.H, 2007, Intermodal transportation. Book Chapter.

[12] Ehrgott, M., 2005, Multicriteria Optimization. Springer, Berlin - Heidelberg.

[13] Steuer, R.E., 1986, Multiple criteria optimization. Wiley, New York.

[14] Keefer, D.L., Kirkwood, C.W., and Corner,J.L., 2004, Perspectives on Decision Analysis Applications, 1990-2001, Decision Analysis, 1(1), 4-22.

[15] Keeney, R.L., 2002, Common mistakes in making value trade-offs. Operations Research, 50(6): 935-945.

[16] Ahuja, R.K., Magnanti, T.L., and Orlin, J.B., 1993, Network Flows: Theory, Algorithms, and Applications, Prentice-Hall, Upper Saddle River, NJ.

[17] Wallenius, J., Dyer, J.S., Fishburn, P.C., Steuer, R.E., Zionts, S., and Deb, K., 2008, Multiple Criteria Decision Making, Multiattribute Utility Theory: Recent Accomplishments and What Lies Ahead, Management Science, 54(7), 1336-1349.

[18] Guelat, J., Florian, M., and Crainic, T.G., 1990, A multimode multiproduct network assignment model for strategic planning of freight flows, Transportation Science, 24(1):25-39.

[19] Chang, T.S., 2008, Best routes selection in international intermodal networks. Computers & Operations Research, 35:2877-2891.

[20] Meixell, M.J. and Norbis, M., 2008, A review of the transportation mode choice and carrier selection literature, The International Journal of Logistics Management, 19(2), 183-211.

[21] Dobie, K., 2005, The Core Shipper Concept: A Proactive Strategy for Motor Freight Carriers, Transportation Journal, Spring, 37-53.

[22] Cullinane, K. and Toy, N., 2000, Identifying infuential attributes in freight route/mode choice decisions: a content analysis, Transportation Research Part E, 36, 41-53.

Multiobjective Shortest Paths for Military Transportation

Jeremy D. Jordan, Jeffery D. Weir, and Doral E. Sandlin

Air Force Institute of Technology

2950 Hobson Way, Wright-Patterson AFB, Ohio, 45433,

USA, +1 937-255-3636, jeremy.jordan@afit.edu

jeffery.weir@afit.edu

doral.sandlin@afit.edu

Abstract

The military transportation environment is becoming increasingly complex. United States Transportation Command (USTRANSCOM) therefore requires advanced methods to assist with transportation mode selection. Combining value focused thinking and network optimization, we create a unique formulation called the multiobjective average shortest path (MASP) problem. An algorithm that solves to optimality is provided and its benefits are discussed. We then present a faster heuristic and examine its performance under a variety of conditions. In addition, it may not be feasible to simply optimize multiobjective value and ignore the path length. Therefore, a multicriteria optimization problem is formulated and solved that allows tradeoffs between path length and value. Pareto optimal solutions are generated and compared.

Keywords

3.1 Introduction

As the military continues to drawdown forces and reduce budgets, attention to transportation costs is increasing. With a $120 billion annual budget, logistics in the Department of Defense is extremely complicated [102]. Transportation costs alone exceed $10 billion and therefore opportunities for improvement must be exploited. The joint command in charge of this budget and responsible for all military transportation during peace and war times is USTRANSCOM. The methodology in this

174

paper is directly applicable to their practices and has the potential to save valuable resources.

The idea is to combine Multiobjective decision analysis [60] and network optimization [3] to create a stronger methodology for general use by a variety of practitioners. Specifically, we use multiobjective values obtained from Value Focused Thinking [61] as arc costs in the shortest path problem. This creates the multiobjective average shortest path (MASP) problem. The power of the MASP lies in its ability to capture multiple quantitative and qualitative factors in a network environment and generate solutions without the use of complicated multicriteria optimization procedures. In fact, we know the use of multicriteria optimization [35] for network problems is NP-hard for more than 3 factors [47].

Additionally, the MASP cannot be solved using existing shortest path algorithms because of the MASP's non-additive objective function. Therefore, two new methods are introduced. Given a graph G with N nodes and V vertices, the first method solves the MASP to optimality in $O(N^2 + N) = O(N^2)$ running time. The algorithm uses a dynamic programming approach to generate the shortest path to each node for every possible sized path and divides this by the path size to obtain the shortest average path. The second method estimates the best average path through scaling each of the network arcs. This linear time heuristic solves in $O(V) + O(V) = O(V)$.

3.2   Background and Formulations

A short background on multiobjective decision analysis and the shortest path problem is given. We give the necessary equations and then combine the two methodologies to create the multiobjective average shortest path problem.

3.2.1 Multiobjective Decision Analysis.

Value Focused Thinking is a very popular multiobjective decision analysis technique; understanding what is valued in a decision is important. As Keeney states, "Values are what we care about. As such, value should be the driving force for our decision making" [61]. In order to do this, we need to know the measures important to the decision maker, how the decision maker values levels of the measures, and the weights of each of these measures on the overall decision.

The general additive value model is given by:

$$v_j(x) = \sum_{i=1,j=1}^{n,m} w_i v_i(x_{ij}) \tag{50}$$

for $i = 1, 2, ..., n$ measures and $j = 1, 2, ..., m$ alternatives, where $v_i(x_{ij})$ is the single dimension value function of measure i evaluated for alternative j, w_i is the weight of measure i, and $v_j(x)$ is the multiobjective value for alternative j. The objective is to maximize value over the j alternatives. The value of a particular alternative j, or $v_j(x)$, is simply the value of an alternative in relation to what is important to the decision. A $v_j(x) = 1$ is indicative of an alternative that achieves everything desired in a decision, whereas $v_j(x) = 0$ indicates an alternative achieving no value in the decision. For more detailed information on this, see [61, 60, 65]. Alternatively, another popular multiobjective technique is the analytic hierarchy process (AHP). The AHP function could easily be substituted for the VFT function in this methodology if desired.

3.2.2 Shortest Path Problem.

Assume a graph G $= (V, A)$ to be a directed acyclic graph (DAG) with $|V| = n$ vertices and $|A| = m$ arcs. Each arc $(i, j) \in A$ has some cost $c_{i,j}$ and flow $x_{i,j}$ associated with its use. The source and sink vertices are designated s and t, respectively. Then the general shortest path problem is:

$$\min z = \sum_{(i,j) \in E} c_{i,j} x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = \begin{cases} 1, & \text{if i=s;} \\ -1, & \text{if i=t;} \\ 0, & \text{otherwise} \forall i \end{cases}$$

$$x_{i,j} \geq 0 \ \ \forall \ (i, j) \in E$$

$$(51)$$

where $x_{i,j}$ is the flow from vertex i to j and $c_{i,j}$ is the cost of a unit of flow from i to j. This is easily solved using the simplex method or any other LP algorithm. However, since it maintains a unique structure, faster algorithms are available. Furthermore, because we specified the network to be a directed acyclic graph, linear time algorithms are available to solve to optimality. Dijkstra's algorithm [3] is the most common shortest path algorithm and solves in O(N^2). This algorithm solves any shortest path problem, not just DAG's. An often overlooked dynamic programming algorithm that solves the DAG in linear time O(V) is called the Reaching Method. It begins with a topological sort of the nodes and labels each vertex successively.

Step 1: Topologically order the DAG G

Step 2: For $i = 1, ..., N$, set dist(i)=0

Step 3: For $i = 1, ..., N - 1$, for each edge $V(i), u$ outgoing from $V(i)$, if $\text{dist}(V_i) + G(V_i, u) > \text{dist}(u)$, then set $\text{dist}(u) = \text{dist}(V_i) + G(V_i, u)$

Step 4: $\text{dist}(N)$ is the shortest path to N

The algorithm above is easily changed to gather the longest path to every node rather than the shortest path. This is the underlying dynamic programming approach behind the proposed heuristic.

3.2.3 MASP Formulation.

Combining the shortest path problem with the additive value model results in a unique formulation. Simply substituting the $c_{i,j}$ in Formulation 51 with the $v_j(x)$ in Equation 50 ignores the fact that we are seeking to find the best average value through the network. Such a substitution simply results in a longest path rather than a highest average. Therefore, the multiobjective average shortest path (MASP) problem is:

$$\max z = \frac{\sum\limits_{(i,j) \in E} \left(\sum\limits_{k=1}^{l} w_k v_{i,j,k} \right) x_{i,j}}{\sum y_{i,j}}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{j,i} = \begin{cases} 1, & \text{if i=s;} \\ -1, & \text{if i=t;} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{i,j} \leq y_{i,j} \quad \forall \ i,j \in E$$

$$x_{i,j} \geq 0 \quad \forall \ i,j \in E$$

$$\sum_{k=1}^{l} w_k = 1 \tag{52}$$

$$y_{i,j} \in \{0,1\} \tag{53}$$

$$\tag{54}$$

where w_k is weight of measure k, $v_{i,j,k}$ is the value function at measure k between nodes i and j, and $x_{i,j}$ is the flow from vertex i to j. Solution methods for the MASP are given in Section 3.3.

3.2.4 A Multicriteria Multiobjective Average Shortest Path Problem.

Even with optimal algorithms, normal average shortest path algorithms may add arcs to increase the overall average. However, in some applications such as transportation mode selection, adding on arcs may cause unseen problems. Increased handling normally results in increased costs, increased damage/loss, and increased chance of mistakes. The multiobjective optimization 55 for the MASP allows the decision maker to tradeoff value with additional arcs in the network path by generating all pareto optimal solutions. Because multiple objectives are handled by the additive value model, the problem remains a bi-objective multicriteria optimization problem,

the first criteria being the value model and the second the number of arcs in the solution path. This is important because including more than three criteria makes the problem intractable as noted above. The multicriteria multiobjective average shortest path problem is

$$\min z = \lambda_1 \left(\sum_{(i,j) \in E} v_{i,j} x_{i,j} \right) + \lambda_2 \sum y_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = \begin{cases} 1, & \text{if i=s;} \\ -1, & \text{if i=t;} \\ 0, & \text{otherwise.} \end{cases}$$

$$\sum \lambda_k = 1$$

$$x_{i,j} \leq y_{i,j} \quad \forall \ i,j \in E$$

$$x_{i,j} \geq 0 \quad \forall \ (i,j) \in E$$

$$y_{i,j} \in \{0,1\} \tag{55}$$

$$\tag{56}$$

where $x_{i,j}$ is the flow between node i and j (1 in the shortest path problem), $v_{i,j}(x)$ is the multiobjective value of using node i and j, and λ_k is the multicriteria weight of criteria k. The formal multicriteria approach this resembles is the weighted sum problem [35]. This problem generates and allows us to obtain efficient solutions, also referred to as pareto optimal solutions. Taken together, this group of solutions comprises the efficient frontier. A single solution that simultaneously optimizes both variables is not attainable. The decision maker must make tradeoffs between pareto solutions along the pareto front. An increase in one variable yields a certain decrease in another. This problem is easily solved using data gathered from the application of

the average shortest path algorithms. We discuss in section 3.4.2.

3.3 MASP Solution Methods

Three methods are available to solve the MASP. The first solution method, Wimer's algorithm [108], solves to optimality in $O(N^3)$; the method and algorithm are summarized in this section. The second solution method uses a dynamic programming approach similar to Wimer's algorithm, but solves in $O(N^2)$. The final solution method is a heuristic that utilizes the dynamic programming reaching method twice, and therefore runs in $O(V) + O(V) = O(V)$ linear time.

3.3.1 Wimer's Algorithm.

[108] lays out the foundation for determining the optimal average path length through the path length minimization algorithm. This algorithm pursues the optimal path by determining the best average path of cardinality j at each node. First, each vertex is assigned a rank according to its maximum cardinality (number of arcs) of a path from s to the vertex. The source, s, is obviously assigned a rank of 0 and the sink, t has the highest rank. Vertices are numbered according to their rank, starting with s, and numbering vertices with equal ranks arbitrarily. So, s is numbered 1, t is numbered $|U|$ and for every arc $e(u, v)$ the vertex u is assigned a smaller number than the vertex v.

Define $G = (U, E)$. Let u be a vertex on some path from source s to sink t. Only the shortest path with cardinality j can be part of the shortest average arc length path from s to t. Each vertex $u \in U$ is next assigned a vector $L(u)$ of length $|U|$. The jth element of $L(u)$, $L_j(u)$ with $(0 \leq j \leq |U| - 1)$, is the minimum length of any path from s to u with cardinality j. $\Pi_j(u)$ denotes the minimum length path or paths. Since $G(U, E)$ is acyclic, the cardinality of a path cannot be greater than $|U| - 1$. If

no path exists for a cardinality, the path is assigned ∞. Another vector P_u of length $|U|$ is associated with u, whose jth element $P_j(u)$ is the last vertex preceding u on $\Pi_j(u)$. This is the vertex v for which $L_{j-1}(v) + l(e(v, u)) = L_j(u)$. If $L_j(u) = \infty$, then $P_j(u) = 0$

Starting at s, a new vertex is marked at each iteration until t is reached. Once a vertex u is labeled, the length of the shortest path from s to u is known for every cardinality between 0 and $|U| - 1$. The sets of arcs entering and leaving $u \in U$ are denoted $\Gamma^{in}(u)$ and $\Gamma^{out}(u)$. The algorithm is as follows:

Step 0: Initialization. Set $L_0(s) = 0$ and $L_j(s) = \infty$, $1 \leq j \leq |U| - 1$. Mark s and set $T = U - \{s\}$. For every $u \in T$ set $L_j(u) = \infty$, $0 \leq j \leq |U| - 1$. For every $u \in U$ define $P_j(u) = \emptyset$, $0 \leq j \leq |U| - 1$.

Step 1: New Vertex Selection. Find a vertex $u \in T$ for which all the tail vertices of the arcs in $\Gamma^{in}(u)$ are already marked. Such a vertex must exist since $G(E, U)$ is an acyclic digraph with a single source and a single sink whose vertices are numbered as described above.

Step 2: Updating the minimum path lengths. Determine the shortest path length vector L_u by considering every vertex v for which $e(v, u) \in \Gamma^{in}(u)$ as follows.

$$L_j(u) = min\{L_{j-1}(v) + l(e(v, u)) | e(v, u) \in \Gamma^{in}(u)\}, \quad 1 \leq j \leq |U| - 1. \qquad (57)$$

Let v^* be the vertex obtained when solving Equation 57 for given u and j. Then, set $P_j(u) = v^*$.

Step 3: Updating the set of marked vertices. Mark u and set $T = T - \{u\}$.

Step 4: Termination Test. If $u = t$ then go to Step 5, else go to Step 1.

Step 5: Retrieving the minimum average arc length path. Upon termination, every $L_j(t) \leq \infty$ is the length of the shortest path from s to t among all the paths of

182

cardinality j. For every j satisfying $L_j(t) = \infty$ there exists no path of cardinality j from s to t. Evidently, $\min\{L_j(t)/j|1 \leq j \leq |U| - 1$ yields the minimum average arc length for any path from s to t. Let j^* be the cardinality of the path for which the minimum average arc length was obtained. Then, the desired path is retrieved by traversing backwards from t to s as follows. We start from t and go backwards to the vertex stored in $p_{j^*}(t)$. We then go backwards to the vertex stored in $P_{j^*-1}[P_{j^*}(t)]$ and continue in the same manner until s is reached.

3.3.2   Optimal Algorithm.

Dynamic programming is an optimization technique that breaks a problem into multiple smaller problems in order to obtain solutions more efficiently. The average shortest path problem for a directed acyclic graph can also be solved using a dynamic programming approach (J.B. Orlin, personal communication, November 16, 2011). "Assuming you are finding a shortest path from node s to node t, you can let d(j, k) be the shortest path from node s to node j with exactly k arcs. (This does not work in networks with cycles because it actually computes the shortest walk from s to j with k arcs.) The values of d(,) can be computed using dynamic programming. The shortest average length of a path from s to t is min d(t, k)/k for k = 2 to n-1. The same technique computes a shortest average path from node s to each other node."

The algorithm is easily coded for a maximum average shortest path. Preliminary results are provided in Section 3.4.2.

3.3.3   Scaling Heuristic.

Following the methodology of [108], we first assign each vertex according to its maximum cardinality (number of arcs) of a path from s to the vertex. The source,

s, is obviously assigned a rank of 0 and the sink, t has the highest rank. Vertices are numbered according to their rank, starting with s, and numbering vertices with equal ranks arbitrarily. So, s is numbered 1, t is numbered $|U|$ and for every arc $e(u,v)$ the vertex u is assigned a smaller number than the vertex v. Let the rank of a particular node be denoted $R(j)$. Hence $c_{i,j}$, the cost of a unit of flow from i to j, need only be scaled by

$$\frac{R(j) - R(i)}{R(t)}. \tag{58}$$

Let

$$c^*_{i,j} = c_{i,j} \left(\frac{R(j) - R(i)}{R(t)} \right), \tag{59}$$

the average shortest path problem becomes

$$\min z = \sum_{(i,j) \in E} c^*_{i,j} x_{i,j}$$

$$s.t. \sum_{(i,j) \in E} x_{i,j} - \sum_{(j,i) \in E} x_{i,j} = \begin{cases} 1, & \text{if i=s;} \\ -1, & \text{if i=t;} \\ 0, & \text{otherwise.} \end{cases}$$

$$x_{i,j} \geq 0 \ \forall \ (i,j) \in E$$

$$\tag{60}$$

Formulation 60 can now be solved using any known shortest path algorithm. Using the Reaching Algorithm from Dynamic Programming solves the problem in linear time. The following is a step by step methodology for the heuristic:

Step 1: Temporarily replace each $c_{i,j} \ ¿ \ 0$ with 1.

Step 2: Use the Reaching Algorithm to solve the longest path to each vertex giving its maximum cardinality

Step 3: Assign each vertex a rank according to its maximum cardinality (number of arcs) of a path from s to the vertex

Step 4: Vertices are numbered according to their rank, starting with s, and numbering vertices with equal ranks arbitrarily. Let the rank of a particular node be denoted $R(k)$, for $k = 1, 2, ..., n$ vertices.

Step 5: Scale the original $c_{i,j}$'s by $\frac{R(j)-R(i)}{R(t)}$. Let $c_{i,j}^* = c_{i,j} \left(\frac{R(j)-R(i)}{R(t)} \right)$

Step 6: Replace $c_{i,j}$ with $c_{i,j}^*$ in original formulation

Step 7: Solve new formulation with the Reaching Algorithm

In summary, the reaching algorithm is used to obtain the maximum number of arcs to each node. Using this, the original cost coefficients are scaled. The reaching algorithm is then used a second time to solve the scaled network to optimality. The next section presents some preliminary results of the heuristic.

3.4   Results

This section presents some preliminary work on the performance of the heuristic along with pareto optimal solutions using data from the optimal algorithms.

3.4.1   Heuristic Results.

Three experimental factors were used in testing: total nodes in the network, reachability, and range in the arc costs. Reachability is defined as the maximum distance any node can reach ahead. For instance, in a 100 node network, a reachability level equal to .25 indicates that for any particular node in the network, this node can at most reach a node 25 nodes away. Range is the range of values allowed in the mul-

tiobjective value function at each arc. Since values of between 0 and 1 are the only achievable levels, the allowable range cannot be above 1.

The errors in Table 13 are for a completely dense directed acyclic graph. This results in a 49.5% dense network in matrices with a reachability of 1. This is due to the fact that DAG's are upper triangular matrices. In reality, a network will likely be much less dense. In fact, in a DAG, a density of 5% is very realistic. The heuristic has performed much better under more sparse networks but further testing is needed to formalize its performance. We've encountered more variability in sparse networks however. For instance, the heuristic will either find the correct path, or find the 2nd best path which could be 10% away from the optimal. All errors presented here are for a fully dense DAG.

Range appears is the most influential factor; more range in the arc costs yields greater error. Of course, as the range of possible values increases, one expects error to increase as well. In the case of the MASP, all values will be between 0 and 1 and therefore a range of 1 is the worst case. Clearly, the heuristic performs within bounds of the optimal solution divided by 1.05, which is very desirable. Using regression analysis, we determined that # of nodes and range both significantly influenced the error. As nodes increased, a decrease in error was observed. These results are omitted and are expected to be published in a forthcoming paper.

Keep in mind the most important benefit of this heuristic is its speed. The next fastest known algorithm to solve the average shortest past problem is non-linear, $O(N^2)$, this heuristic is linear, $O(V)$. For problems with 100,000 or 1 million nodes, this is a considerable savings. It could make an intractable problem tractable. A separate study is necessary to observe actually practical performance. It is also noteworthy that for a fair amount of the randomly produced matrices in Table 13, the heuristic finds the optimal solution. In many of the sparse networks we've tested, the

Table 13. Scaling Heuristic Worst Case Errors Under Different Network Conditions

# of Nodes	Reachability	Range	Run 1	Run 2	Run 3	Run 4	Run 5	Average
20	0.05	0.2	1.00	1.00	1.00	1.00	1.00	1.00
60	0.05	0.2	1.01	1.01	1.00	1.00	1.01	1.01
100	0.05	0.2	1.00	1.00	1.00	1.00	1.00	1.00
20	0.25	0.2	1.00	1.00	1.00	1.00	1.00	1.00
60	0.25	0.2	1.01	1.01	1.01	1.00	1.01	1.01
100	0.25	0.2	1.01	1.01	1.01	1.01	1.00	1.01
20	1	0.2	1.00	1.01	1.00	1.00	1.00	1.00
60	1	0.2	1.01	1.01	1.02	1.00	1.01	1.01
100	1	0.2	1.01	1.01	1.03	1.00	1.01	1.01
20	0.05	0.6	1.00	1.00	1.00	1.00	1.00	1.00
60	0.05	0.6	1.01	1.00	1.00	1.00	1.02	1.01
100	0.05	0.6	1.02	1.00	1.01	1.01	1.01	1.01
20	0.25	0.6	1.02	1.01	1.02	1.00	1.00	1.01
60	0.25	0.6	1.00	1.01	1.01	1.04	1.01	1.01
100	0.25	0.6	1.00	1.01	1.00	1.05	1.02	1.02
20	1	0.6	1.00	1.00	1.00	1.01	1.00	1.00
60	1	0.6	1.00	1.03	1.00	1.13	1.02	1.04
100	1	0.6	1.00	1.10	1.00	1.00	1.03	1.03
20	0.05	1	1.00	1.00	1.00	1.00	1.00	1.00
60	0.05	1	1.00	1.02	1.02	1.00	1.03	1.01
100	0.05	1	1.03	1.04	1.08	1.02	1.03	1.04
20	0.25	1	1.05	1.04	1.10	1.00	1.08	1.05
60	0.25	1	1.17	1.02	1.01	1.03	1.04	1.05
100	0.25	1	1.05	1.03	1.07	1.01	1.03	1.04
20	1	1	1.00	1.13	1.00	1.00	1.02	1.03
60	1	1	1.02	1.00	1.00	1.14	1.08	1.05
100	1	1	1.02	1.00	1.10	1.05	1.00	1.04

heuristic obtains the optimal solution even more frequently. However, more variance in the error exists.

Finally, a better approach to testing is design of experiments [77]. In future testing, we plan to incorporate the three factors above along with network density and multiobjective arc cost probability distributions. In this way, experimental conditions where the heuristic performs well can be determined with statistical certainty. We use the term statistical certainty because each of the arc costs are randomly generated.

If all variables were fixed, the results would be deterministic.

3.4.2 Analysis.

Optimal algorithms are now used to create optimal and non-optimal paths and values to choose from. The optimization problem from formulation 55 in Section 3.2 underlie the graphs in this section. Each of the graphs are generated with the distance matrix from Orlin's or Wimer's algorithm. Because the shortest path to each node is generated, we can easily get the shortest average path to each node, along with the number of arcs.

Figures 71, 72, 73, 74 show the surfaces created by random graphs. Each surface is different. In Figure 71 for instance, the optimal solution is somewhere around 7 arcs. However, the decision maker could choose to sacrifice .7 value to go from source to sink in 1 arc. In contrast, going from the source to sink in 1 step in Figure 72 creates a significant loss of value, .5. Figure 74 shows a discontinuous efficient frontier. For arc lengths around 4, the solution is strictly dominated by the arc length of 1, but also dominated by arc lengths greater than 5.

The optimal solution to the graph in Figure 75 is a 10 arc path (1 10 18 28 39 47 48 84 91 94 100) with an optimal average value of 0.9765. Table 14 shows all paths shorter than the optimal path. All paths longer than the optimal solution in this case give a worse solution. The optimal path and all paths in Table 14 strictly dominate the longer paths to the right. In this case, the user may choose the path with 2 arcs and a value of .9223 rather than choosing the optimal value of .9765 with 10 arcs. In many cases, an added value of .05 or .1 may not be worth adding multiple arcs. In other cases, transfers may require minimal effort and therefore the user may choose to optimize value and ignore arc length.
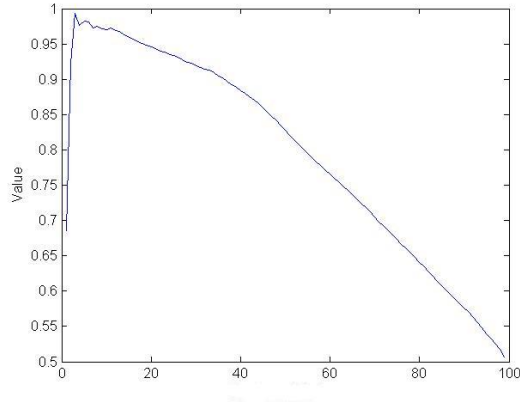
Table 14. Arcs vs Value

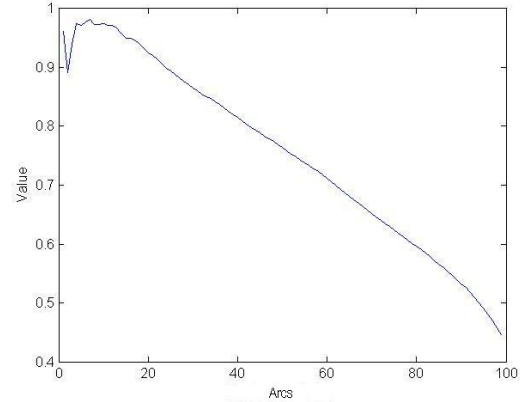| Number of arcs	1	2	3	4	5	6	7	8	9
Value	0.6905	0.9223	0.9549	0.9604	0.9631	0.9665	0.9685	0.9733	0.9765
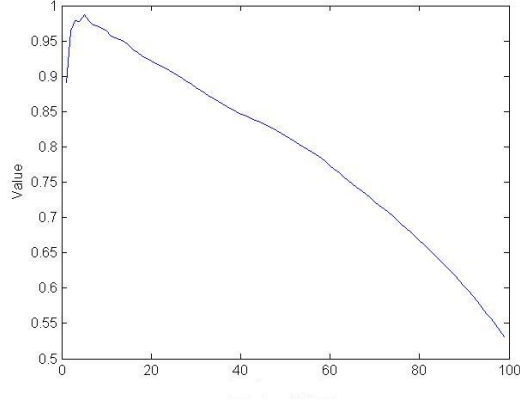
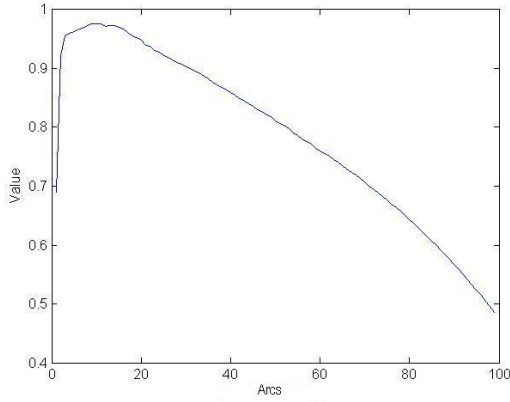**Figure 71. Values vs Arcs**
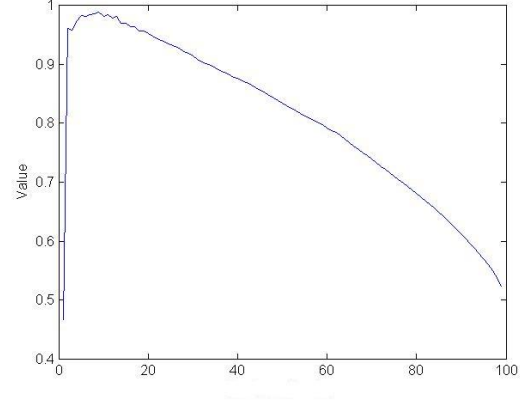
Figure 72. Values vs Arcs

Figure 73. Values vs Arcs
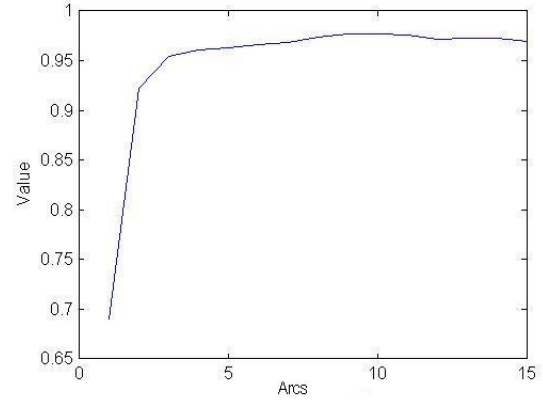

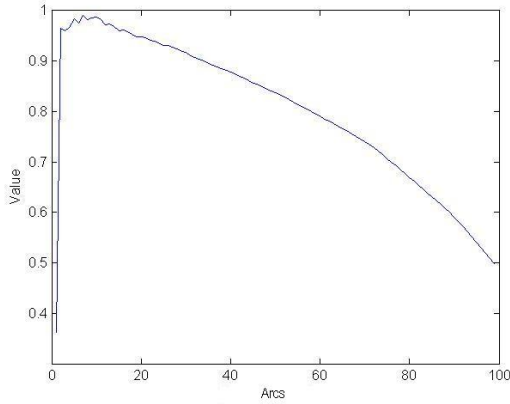**Figure 74. Values vs Arcs**

3.5 Conclusions

The scaling heuristic performs very well under all the conditions we've encountered. Future work includes further testing under additional conditions and running the heuristic on known problems. We hypothesize the heuristic will perform well regardless of the conditions. However, optimality is still important. Wimer's algo-
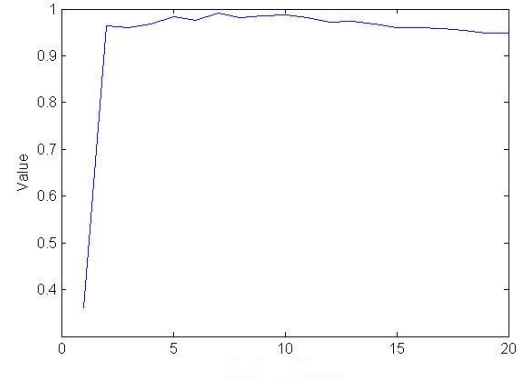
189

Figure 75. Values vs Arcs

Figure 76. Closeup of Figure 75

Figure 77. Values vs Arcs

Figure 78. Closeup of Figure 77

rithm solves to optimality and provides similar information to the algorithm suggested by Dr. Orlin. A decision maker can use this information to make tradeoffs between adding value to the network and adding arcs to the network. In many cases, a decision maker is willing to give up small amounts of value for a decrease in arcs and transfers. Therefore, it's important not only to determine the optimal average path through the network, but also to generate all possible paths through the network. This allows the user to exercise preference through the use of the multicriteria optimization weighted sum method.

190

The MASP provides a basis for applying these methods to USTRANSCOM; it allows us to determine the best path for a single product to flow through the system. In actuality, USTRANSCOM is responsible for shipping numerous products. Implementation of the MASP to the processes of USTRANSCOM will certainly require additional constraints, making the problem a variation of the minimum cost network flow model. The resulting model is called the multiobjective maximum value network flow problem and its formulation can be found in [56]. Results from this problem are forthcoming.

Bibliography

[1] USTRANSCOM. Mission brief. as of 1 December 2009.

[2] R. L. Keeney and H. Raiffa. Decisions with Multiple Objectives: Preferences and Value Tradeoffs. Wiley, New York, NY, 1976.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. Network flows. Upper Saddle River, NJ: Prentice Hall., 1st edition, 1993.

[4] Ralph L. Keeney. Value-Focused Thinking: A Path to Creative Decisionmaking. Harvard University Press, Cambridge, MA, 1992.

[5] M. Ehrgott. Multicriteria Optimization. 2005.

[6] H.W. Hamacher, C.R. Pederson, and S. Ruzika. Multiple objective minimum cost flow problems: A review. European Journal of Operational Research, 176:14041422, 2007.

[7] Craig W. Kirkwood. Strategic Decision Making. Duxbury Press, Belmont, CA, 1996.

[8] S. Wimer, I. Koren, and I. Cederbaum. On paths with the shortest average arcs lengths in weighted graphs. Discrete Applied Mathematics, 45:169179, 1993.

[9] D. C. Montgomery. Design and Analysis of Experiments. Wiley, 3rd edition, New York:1991.

[10] J.D. Jordan, J.D. Weir, and D.E. Sandlin. Multiobjective decision programming for the multiobjective minimum cost flow problem. In proceedings of the 2011 Industrial Engineering Research Conference.

Bibliography

[1] Technical report.

[2] J. Acevedo. An algorithm for multiparametric mixed-integer linear programming problems. *Operations Research Letters*, 24(2):139–148, 1999.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows*. Upper Saddle River, NJ: Prentice Hall., 1st edition, 1993.

[4] A. Arbel and L. G. Vargas. Preference simulation and preference programming: robustness issues in priority deviation. *European Journal of Operational Research*, 69:200–209, 1993.

[5] Masood A. Badri. Combining the analytic hierarchy process and goal programming for global facility location-allocation problem. *International Journal of Production Economics*, 62:237–248, 1999.

[6] W.J. Baumol and H.D. Vinod. An inventory theoretic model of freight transport demand. *Management Science*, 16(7):413–421, 1970.

[7] J.F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252, 1962.

[8] M. Beuthe, C. Bouffioux, J.D. Maeyer, G. Santamaria, M. Vandresse, E. Vandaele, and F. Witlox. *Advances in Spatial Sciences: Methods and Models in Transport and Tele-Communications, Cross Atlantic Perspectives*, chapter A Multi-Criteria Methodology for Stated Preferences among Freight Transport Alternatives, pages 163–179. Springer, 2005.

[9] C. E. Blair and R. G. Jeroslow. The value function of an integer program. *Mathematical Programming*, 23:237–273, 1982.

[10] T.S. Chang. Best routes selection in international intermodal networks. *Computers & Operations Research*, 35:2877–2891, 2008.

[11] H. Chen and D. F. Kocaoglu. A sensitivity analysis algorithm for hierarchical decision models. *European Journal of Operational Research*, 185:266–288, 2008.

[12] Y.L. Chen. The minimal average cost flow problem. *EJOR*, 81:561–570, 1995.

[13] M. Chouman and T.G. Crainic. A mip-tabu search hybrid framework for multicommodity capacitated fixed-charge network design. CRT-2010-31 University of Montreal, CIRRELT, 2010.

[14] P.T. Chuang. Combining the analytic hierarchy process and quality function deployment for a location decision from a requirement perspective. *The International Journal of Advanced Manufacturing Technology*, 18(11):842–849, 2001.

[15] R. T. Clemen and T. Reilly. *Making Hard Decisions with DecisionTools*. Duxbury Press, Belmont, CA, 1997.

[16] Robert T. Clemen. *Making hard decisions : An Introduction to decision analysis*. Duxbury Press, 2nd edition edition, 1996.

[17] W. Cook, A.M.H. Gerards, A. Schrijver, and E. Tardos. Sensitivity theorems in integer linear programming. *Mathematical Programming*, 34:251–264, 1986.

[18] L. Cooper and C. Drebes. An approximate solution method for the fixed charge problem. *Naval Research Logistics Quarterly*, 8:101–113, 1967.

[19] James L. Corner and Craig W. Kirkwood. Decision analysis applications in the operations research literature, 1970-1989. *Operations Research*, 39:206–219, 1991.

[20] A.M. Costa. A survey on benders decomposition applied to fixed-charge network design problems. *Computers & Operations Research*, 32:1429–1450, 2005.

[21] T.G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2), 2000.

[22] T.G. Crainic, A. Frangioni, and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1-3), 2001.

[23] T.G. Crainic and K.H. Kim. *Transportation*, chapter Intermodal Transportation, pages 467–537. North Holland-Elsevier, 2007.

[24] A. Crema. Average shadow price in a mixed integer linear programming problem. *Operations Research Letters*, 85:625–635, 1995.

[25] K. Cullinane and N. Toy. Identifying influential attributes in freight route/mode choice decisions: a content analysis. *Transportation Research Part E*, 36:41–53, 2000.

[26] J.R. Current and H. Min. Multiobjective design of transportation networks: Taxonomy and annotation. *European Journal of Operational Research*, 26:187–201, 1986.

[27] J.R. Current, C.S. ReVelle, and J.L. Cohon. An interactive approach to identify the best compromise solution for two objective shortest path problems. *Computers & Operations Research*, 17(2):187–198, 1990.

[28] S. Dasgupta, C. Papadimitriou, and U. Vazirani. *Algorithms*. McGraw Hill, 2007.

[29] M. W. Dawande and J. N. Hooker. Inference-based sensitivity analysis for mixed integer/linear programming. *Operations Research*, 48(4):623–634, 2000.

[30] L. de Boer, E. Labro, and P. Morlacchi. A review of methods supporting supplier selection. *European Journal of Purchasing & Supply Management*, 7:75–89, 2001.

[31] D.R. Denzler. An approximate algorithm for the fixed charge problem. *Naval Research Logistics Quarterly*, 16:411–416, 1969.

[32] Kathryn Dobie. Core shipper concept. *Transportation Journal*, 44(2):37–53, 2005.

[33] Vivek Dua and Efstratios N. Pistikopoulos. An algorithm for the solution of multiparametric mixed integer linear programming problems. *Annals of Operations Research*, 99:123–139, 2000.

[34] Riccardo Dulmin and Valeria Mininno. Supplier selection using a multi-criteria decision aid method. *Journal of Purchasing & Supply Management*, 9, 2003.

[35] M. Ehrgott. *Multicriteria Optimization*. 2005.

[36] P. Erds and A. Rnyi. On random graphs (I). *Publicationes Mathematicae (Debrecen)*, 6:290–297, 1959.

[37] A.V. Fiacco. *Introduction to sensitivity and stability analysis in nonlinear programming*. 1983.

[38] Ernest H. Forman and Saul I. Gass. The analytic hierarchy process - an exposition. *Operations Research*, 49(4):469–486, 2001.

[39] H. Friesdorf and H. Hamacher. Weighted min cost flows. *European Journal of Operational Research*, 11:181–192, 1982.

196

[40] S.A. Gabriel and D. Bernstein. The traffic equilibrium problem with nonadditive path costs. *Transportation Science*, 34(4):337–348, 1997.

[41] S.A. Gabriel and D. Bernstein. Nonadditive shortest paths: Subproblems in multi-agent competitive network models. *Computational & Mathematical Organization Theory*, 6:29–45, 2000.

[42] M. Garey and D. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W.H. Freeman and Company, New York, NY, 1979.

[43] A.M. Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory*, 10(4):237–260, 1972.

[44] A.M. Geoffrion and R. Nauss. Parametric and postoptimality analysis in integer linear programming. *Management Science*, 23(5):453–466, 1977.

[45] S.H. Ghodsypour and C. OBrien. A decision support system for supplier selection using an integrated analytic hierarchy process and linear programming. *Int. J. Production Economics*, 56-57:199–212, 1998.

[46] J. Guelat, M. Florian, and T.G. Crainic. A multimode multiproduct network assignment model for strategic planning of freight flows. *Transportation Science*, 24(1):25–39, 1990.

[47] H.W. Hamacher, C.R. Pederson, and S. Ruzika. Multiple objective minimum cost flow problems: A review. *European Journal of Operational Research*, 176:1404–1422, 2007.

[48] Mike Hewitt, George L. Nemhauser, and Martin W.P. Savelsbergh. Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing*, 22(2), 2010.

[49] W.M. Hirsch and G.B. Dantzig. The fixed charge problem. *Naval Research Logistics Quarterly*, 9:413–424, 1968.

[50] William Ho. Integrated analytic hierarchy process and its applications - a literature review. *European Journal of Operational Research*, 186:211–228, 2008.

[51] S. Holm and D. Klein. Discrete right hand side parametrization for linear integer programs. *European Journal of Operational Research*, 2(1):50–53, 1978.

[52] Ronald A. Howard. Decision analysis: Practice and promise. *Management Science*, 34:679–695, 1998.

[53] Larry Jenkins. Parametric mixed integer programming: An application to solid waste management. *Management Science*, 28(11):1270–1284, 1982.

[54] Larry Jenkins. Parametric methods in integer linear programming. *Annals of Operations Research*, 27:77–96, 1990.

[55] A. Jimenez, S. Rios-Insua, and A. Mateos. A decision support system for multiattribute utility evaluation based on imprecise assignments. *Decision Support Systems*, 36:65–79, 2003.

[56] J.D. Jordan, J.D. Weir, and D.E. Sandlin. Multiobjective decision programming for the multiobjective minimum cost flow problem. In *proceedings of the 2011 Industrial Engineering Research Conference*.

[57] J.D. Jordan, J.D. Weir, D.E. Sandlin, and S.P. Dillenburger. The multiobjective average longest path problem. *Submitted to European Journal of Operational Research*, 2012.

[58] Richard M. Karp. *Reducibility Among Combinatorial Problems, in Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, 1972.

[59] D. L. Keefer, C. W. Kirkwood, and J. L. Corner. Perspectives on decision analysis applications, 1990-2001. *Decision Anal.*, 1(1):4–22, 2004.

[60] R. L. Keeney and H. Raiffa. *Decisions with Multiple Objectives: Preferences and Value Tradeoffs.* Wiley, New York, NY, 1976.

[61] Ralph L. Keeney. *Value-Focused Thinking: A Path to Creative Decisionmaking.* Harvard University Press, Cambridge, MA, 1992.

[62] Ralph L. Keeney. Common mistakes in making value trade-offs. *Operations Research*, 50(6):935945, 2002.

[63] Mohammad H. Khakbaz, Amir H. Ghapanchi, and Madjid Tavana. A multicriteria decision model for supplier selection in portfolios with interactions. *Int. J. Services and Operations Management*, 7(3):351–377, 2010.

[64] S. Kim and S. Cho. A shadow price in integer programming for management decision. *European Journal of Operational Research*, 37:328–335, 1988.

[65] Craig W. Kirkwood. *Strategic Decision Making.* Duxbury Press, Belmont, CA, 1996.

[66] D. Klein and S. Holm. Integer programming post-optimal analysis with cutting planes. *Management Science*, 25(1), 1979.

[67] L.S. Lasdon. *Optimization Theory for Large Systems.* Macmillian, 1st edition, 1970.

[68] E.L. Lawler. *Combinatorial Optimization: Networks and Matroids,.* Holt, Rinehart, and Winston, 1976.

[69] M.J. Liberatore and T. Miller. A decision support approach for transport carrier and mode selection. *Journal of Business Logistics*, 16(2):85–115, 1995.

[70] T. Magnanti and R. Wong. Network design and transportation planning: models and algorithms. *Transportation Science*, 18(1), 1984.

[71] J. Mangan, C. Lalwani, and B. Gardner. Combining quantitative and qualitative methodologies in logistics research. *International Journal of Physical Distribution & Logistics Management*, 34(7):565–578, 2004.

[72] Ernesto Queiros Vieira MARTINS. On a multicriteria shortest path problem. *European Journal of Operational Research*, 16:236–245, 1984.

[73] A. Mateos, A. Jimenez, and S. Rios-Insua. Monte Carlo simulation techniques for group decision making with incomplete information. *European Journal of Operational Research*, 174:1842–1864, 2006.

[74] M.A. McGinnis. A comparative evaluation of freight transportation choice models. *Transportation Journal*, Winter:36–46, 1989.

[75] S. Melkote and M.S. Daskin. An integrated model of facility location and transportation network design. *Transportation Research Part A: Policy and Practice*, 35(6):515–538, 1985.

[76] D. Mohammaditabar and E. Teimoury. Integrated freight transportation carrier selection and network flow assignment: Methodology and case study. *Journal of Applied Sciences*, 8(17):2928–2938, 2008.

[77] D. C. Montgomery. *Design and Analysis of Experiments*. Wiley, 3rd edition, New York:1991.

[78] K.G. Murty. Solving the fixed charge problem by ranking extreme points. *Operations Research*, 16:268–279, 1968.

[79] J. Mustajoki, R. P. Hämäläinen, and M. R. K. Lindstedt. Using intervals for global sensitivity and worst-case analyses in multiattribute value trees. *European Journal of Operational Research*, 174:278–292, 2006.

[80] Jeffrey Ogden. *Supplier Rating and Supply Base Optimization at Deere & Company*. National Assocation of Purchasing Management (NAPM), 2000.

[81] James B. Orlin. private communication. November 2011.

[82] J.B. Orlin. Private communication. *November*, 2011.

[83] Gregory S. Parnell. *Methods for Conducting Military Operational Analysis: Best Practices in Use Throughout the Department of Defense*, chapter 19, Value-Focused Thinking Using Multiple Objective Decision Analysis. Military Operations Research Society, 2007.

[84] Christopher J. Piper and Andris A. Zoltners. Implicit enumeration based algorithms for postoptimizing zero-one programs. *Naval Research Logistics Quarterly*, 22(4):791–809, 1975.

[85] C.J. Piper and A.A. Zoltners. Some easy postoptimality analysis for zero-one programming. *Management Science*, 22(7):759–765, 1976.

[86] Gary M. Roodman. Postoptimality analysis in integer programming by implicit enumeration: The mixed integer case. *Naval Research Logistics Quarterly*, 21(4):595–607, 1974.

[87] G.M. Roodman. Postoptimality analysis in zero-one programming by implicit enumeration. *Naval Research Logistics Quarterly*, 19(3), 1972.

[88] G. Ruhe. Complexity results for multicriterial and parametric network flows using a pathological graph. *Operations Research*, 32(1):9–27, 1988.

[89] T. L. Saaty. *The Analytic Hierarchy Process*. McGraw-Hill, New York, NY, 1980.

[90] T. L. Saaty. Axiomatic foundation of the analytic hierarchy process. *Management Science*, 32(7):841–855, 1986.

[91] T. L. Saaty. How to make a decision: the analytic hierarchy process. *EJOR*, 48(1):9–26, 1990.

[92] Joseph Sarkis and Srinivas Talluri. A model for strategic supplier selection. *Journal of Supply Chain Management*, 38(1):18–28, 2002.

[93] L.E. Schrage and L.A. Wolsey. Sensitivity analysis for branch and bound integer programming. *Operations Research*, 33:1008–1023, 1985.

[94] P. Serafini. *Some considerations about computational complexity for multi objective combinatorial problems*, chapter In J. Jahn and W. Krabs, editors, Recent advances and historical development of vector optimization, volume 294 of Lecture Notes in Economics and Mathematical Systems,, pages 222–232. Springer Verlag, Berlin, 1986.

[95] J.F. Shapiro. Sensitivity analysis in integer programming. *Annals of Discrete Mathematics*, 1:467–477, 1977.

[96] Mark Shoviak. Decision analysis methodology to evaluate integrated solid waste management alternatives for a remote alaskan air station. Master's thesis, Air Force Institute of Technology, 2001.

[97] D.I. Steinberg. The fixed charge problem. *Naval Research Logistics Quarterly*, 17:217–236, 1970.

[98] R.E. Steuer. *Multiple criteria optimization*. Wiley, New York, 1986.

[99] J.R. Stock and B.J. La Londe. The transportation mode decision revisited. *Transportation Journal*, Winter:51–59, 1977.

[100] Zbigniew Tarapata. Selected multicriteria shortest path problems: An analysis of complexity, models, and adaptation of standard algorithms. *International Journal of Applied Mathematics and Computer Science*, 17(2):269–287, 2007.

[101] Shin-Chan Ting and Danny I. Cho. An integrated approach for supplier selection and purchasing decisions. *Supply Chain Management: An International Journal*, 13(2):116127, 2008.

[102] USTRANSCOM. Mission brief. as of 1 December 2009.

[103] Omkarprasad S. Vaidya and Sushil Kumar. Analytic hierarchy process: An overview of applications. *European Journal of Operational Research*, 169(1):1 – 29, 2006.

[104] Luis G. Vargas. An overview of the analytic hierarchy process and its applications. *European Journal of Operational Research*, 48:2–8, 1990.

[105] M.D. Voss, T.J. Page Jr., S.B. Keller, and J. Ozment. Determining important carrier attributes: A fresh perspective using the theory of reasoned action. *Transportation Journal*, Summer:7–19, 2006.

[106] W.E. Walker. A heuristic adjacent extreme point algorithm for the fixed charge problem. *Management Science*, 22(5), 1976.

[107] Ge Wang, Samuel H. Huang, and John P. Dismukes. Product-driven supply chain selection using integrated multi-criteria decision making methodology. *Int. J. Production Economics*, 91, 2004.

[108] S. Wimer, I. Koren, and I. Cederbaum. On paths with the shortest average arcs lengths in weighted graphs. *Discrete Applied Mathematics*, 45:169–179, 1993.

[109] Frank Witlox and Els Vandaele. Determining the monetary value of quality attributes in freight transportation using a stated preference approach. *Transportation Planning and Technology*, 28(2):77–92, 2005.

[110] L. Wolsey. Integer programming duality: Price functions and sensitivity analysis. *Mathematical Programming*, 20:173–195, 1981.

[111] Laurence A. Wolsey. The b-hull of an integer program. *Discrete Applied Mathematics*, 3:193–201, 1981.

[112] Jiaqin Yang and Huei Lee. An ahp decision model for facility location selection. *Facilities*, 15(9/10):241–254, 1997.

1. REPORT DATE *(DD-MM-YYYY)* 13-09-2012	2. REPORT TYPE **PhD Dissertation**	3. DATES COVERED *(From – To)* Aug 2009 – Sep 2012

4. TITLE AND SUBTITLE THE MULTIOBJECTIVE AVERAGE NETWORK FLOW PROBLEM: FORMULATIONS, ALGORITHMS, HEURISTICS, AND COMPLEXITY	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Jordan, Jeremy, D., Captain, USAF	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)
  Air Force Institute of Technology
  Graduate School of Engineering and Management (AFIT/EN)
  2950 Hobson Street, Building 642
  WPAFB OH 45433-7765
	8. PERFORMING ORGANIZATION
	    REPORT NUMBER
	
	    AFIT/DS/ENS/12-09

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
  Amy A. Papas   USTRANSCOM/TCAC
  508 Scott Drive   Phone: 618-220-7758
  Scott AFB, IL 62225-5357   e-mail: amy.pappas@ustranscom.mil
	10. SPONSOR/MONITOR'S ACRONYM(S)

	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT   APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

   Integrating value focused thinking with the shortest path problem results in a unique formulation called the multiobjective average shortest path problem. We prove this is NP-complete for general graphs. For directed acyclic graphs, an efficient algorithm and even faster heuristic are proposed. While the worst case error of the heuristic is proven unbounded, its average performance on random graphs is within 3% of the optimal solution. Additionally, a special case of the more general biobjective average shortest path problem is given, allowing tradeoffs between decreases in arc set cardinality and increases in multiobjective value; the algorithm to solve the average shortest path problem provides all the information needed to solve this more difficult biobjective problem. These concepts are then extended to the minimum cost flow problem creating a new formulation we name the multiobjective average minimum cost flow. This problem is proven NP-complete as well. For directed acyclic graphs, two efficient heuristics are developed, and although we prove the error of any successive average shortest path heuristic is in theory unbounded, they both perform very well on random graphs. Furthermore, we define a general biobjective average minimum cost flow problem. The information from the heuristics can be used to estimate the efficient frontier in a special case of this problem trading off total flow and multiobjective value. Finally, several variants of these two problems are discussed. Proofs are conjectured showing the conditions under which the problems are solvable in polynomial time and when they remain NP-complete.

15. SUBJECT TERMS   Decision Analysis, Network Optimization, Heuristics, Complexity Theory

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON Dr. Jeffery Weir (ENS)
a. REPORT	b. ABSTRACT	c. THIS PAGE			19b. TELEPHONE NUMBER *(Include area code)* (937) 255-6565, ext 4523; e-mail: Jeffery.weir@afit.edu
U	U	U	UU	212	