12-10-2014

Network Flexibility for Recourse Considerations in Bi-Criteria Facility Location

Jeremy North
University of Missouri-St. Louis, jnorth@murraystate.edu

Network Flexibility for Recourse Considerations in Bi-Criteria Facility Location

Jeremy W. North
M.B.A., General Administration, Pittsburg State University, 2008
B.B.A., Management, Pittsburg State University, 2007

A dissertation submitted to The Graduate School at the University of Missouri – St. Louis in partial fulfillment of the requirements for the degree Ph.D. in Business Administration with an emphasis in Logistics and Supply Chain Management

<u>Advisory Committee</u>
Dr. Ray Mundy, Ph.D. (Chair)
Dr. Robert Nauss, Ph.D.
Dr. Keith Womer, Ph.D.
Dr. Choong Lee, Ph.D.

Table of Contents

Network Flexibility for Recourse Considerations in Bi-Criteria Facility Location

Abstract

What is the best set of facility location decisions for the establishment of a logistics network when it is uncertain how a company's distribution strategy will evolve? What is the best configuration of a distribution network that will most likely have to be altered in the future? Today's business environment is turbulent, and operating conditions for firms can take a turn for the worse at any moment. This fact can and often does influence companies to occasionally expand or contract their distribution networks. For most companies operating in this chaotic business environment, there is a continuous struggle between staying cost efficient and supplying adequate service. Establishing a distribution network which is flexible or easily adaptable is the key to survival under these conditions.

This research begins to address the problem of locating facilities in a logistics network in the face of an evolving strategic focus through the implicit consideration of the uncertainty of parameters. The trade-off of cost and customer service is thoroughly examined in a series of multi-criteria location problems. Modeling techniques for incorporating service restrictions for facility location in strategic network design are investigated. A flexibility metric is derived for the purposes of quantifying the similarity of a set of non-dominated solutions in strategic network design. Finally, a multi-objective greedy random adaptive search (MOG) metaheuristic is applied to solve a series of bi-criteria, multi-level facility location problems.

1. Introduction

1.1 Motivation

Opening production plants and warehouses in a distribution network is very costly. For this reason, the vast majority of the literature approaches this strategic decision from the myopic point of view of cost minimization. However, this approach may not be ideal from the perspective of the end consumer. The customer is a key element and member of the supply chain (Murphy & Wood 2011), yet their needs are rarely addressed in strategic network design. This is unfortunate considering customer service level can usually be significantly improved at a slight increase in cost above the cost minimizing network design solution (Shen & Daskin 2005). This key finding is the primary impetus behind this work. In this research, the trade-off between cost and customer service in muti-criteria strategic network design is explored.

Multi-objective optimization requires the decision maker to select a compromise solution. Usually, the selected compromise solution is one which optimizes no individual objective considered, performs adequately across all criteria, and is not dominated by any other feasible solution. If no strict ordering of preferences or ranking of the criteria can be determined a-priori, then the problem must be addressed by identifying a subset of solutions that are superior to all other alternatives. Pareto optimality or non-dominance is what is required for a solution to be a member of this set. This approach is taken in this research, and the focus is on posteriori solution methods and analyses.

In the field of multi-criteria optimization, several methodologies have been developed to assist the decision maker in the selection of what is often referred to as “the chosen solution.” These methodologies are usually centered on comparing compromise solutions based upon the performance of their respective objective function values. It is argued that although the objective function values are important in that they are what distinguish the efficient solutions from their dominated counterparts in a mathematical model, objective performance doesn’t have to, and perhaps shouldn’t be the only factor determining the selection of an ideal compromise solution for discrete location problems.

The similarity of a network of locations relative to its neighboring solutions on the frontier could also be an important factor to consider when selecting the chosen Pareto efficient solution. The ease and cost of altering a network is related to its similarity to the recourse distribution configurations. In this work, we consider this aspect of decision analysis and lay the foundation for a new multi-criteria decision aid tool for the purposes of identifying solutions which exhibit a high degree of similarity to adjacent efficient solutions on the Pareto frontier.

A network which is flexible in this context is likely to be able to have its structure adjusted more economically in response to evolving logistical strategies concerning level of customer service. Transportation mode shifts, inventory stocking decisions and other considerations can also impact the service level of a company, in addition to location decisions. However, if flexibility is considered at the onset of the network design of a company, then Pareto optimality could be maintained more economically while altering this strategic distribution network in the future.

1.2 Similar Problems in the Literature

Ultimately, this research contributes to the field of multi-criteria decision making in facility location, as evidenced by the literature review given in chapter 3. There have been three prominent surveys on the subject over the years: ReVelle et al. (1981), Current et al. (1990), and Nickel et al. (2005). This indicates that MCDM approaches to facility location continue to be an area of interest to many researchers around the world. However, there are several areas of research other than pure location theory papers which have some similarity with this work. These problems are briefly discussed here.

Conceptually, a problem in the field of management sciences and operations research similar to this work is the Real Options problem. Real options valuation is a technique pioneered in the field of finance as an alternative to discounted cash flow and net present value approaches to capital budgeting problems (Trigeorgis 1996). Real options analysis is defined in Chow & Regan (2011) as the following: “Real options are a corporate finance concept derived from financial options to place a value on the flexibility of an investment decision.”

The technique of real options has been applied in scenarios beyond the purely financial decision analysis world to a variety of decision making problems under uncertainty. In the context of network design, the “real option” is the right to call (expand) or put (contract) the network in reaction to the realization of uncertain business environments. Several studies have been conducted recently applying real options analysis to the network design problem, for example, Chow & Regan (2011), Chow & Regan (2011b), Loureiro et al. (2012).

Although conceptually similar, there are two very important differences between the field of real options research and the approach taken in this research. Firstly, real options problems explicitly consider the evolution of uncertain parameters over time. In other words, they are dynamic optimization problems which incorporate time directly in the model. Secondly, the uncertainty of parameters is modeled as a nonstationary stochastic process, usually as a Brownian motion function. These techniques typically require the use of dynamic programming, while this work is strictly discrete, which is a significant simplification compared to real options work. The consideration of evolving parameters over time and the incorporation of stochastic processes are areas of future research.

Distribution system or supply chain redesign is another problem which has several similarities with this work. In these studies, an established network exists, and a change in the operating environment prompts a business to reevaluate their current distribution strategy. The result of this is usually that their network has been altered based upon senior management wishing to adjust their distribution scheme in response to a changing business environment. An excellent example of one such study is Camm et al. (1997). In that paper, the company Proctor & Gamble realized savings of over $200 million dollars per year in a network redesign initiative. Several warehouse locations were closed as a result of this distribution system rationalization.

The main similarity between this work and the discrete distribution system redesign problem is that the parameters are unchanging during the optimization process, and network redesign is the primary theme of the work. Although these techniques are applied because of a direct result of the changing parameters in the operating environment, the key difference between that field and the research here is that the uncertainty of the operating environment is implicitly considered at the distribution system design or redesign phase in the hopes that future redesign initiatives will be less expensive.

The flexibility of distribution strategies are being explored in a relatively new stream of research bearing some ideological similarities to the research developed here. Lin et al. (2007), Shimizu (2006), Cheshmehgaz et al. (2013), Rad et al. (2014) are some examples of this work. However, this stream of research is centered on the analysis of varying distribution amounts across the arcs in a given network. Therefore, these works are developing tactical level distribution strategies given an established, unalterable network of location decisions. This work focuses solely on strategic level location decisions. Integrating that stream of research with ideas developed here is an area of future research.

There are some graph theoretic papers that are conceptually similar to the work developed in this research, (Graham 1987, Lauri 2011, Koutra et al. 2011, Raymond et al. 2012). Specifically, the area of graph similarity and the sub-graph matching problem have some bearing here. These research areas all focus on nodal evaluations and the connectivity of the graph. However, the methodology developed in this thesis does not consider the distribution arcs or graph connectivity. Incorporating some of these more sophisticated analytical techniques in the analysis of network flexibility is an area of future research.

Figure 1.1 depicts a simple example illustration of the research pursed here.

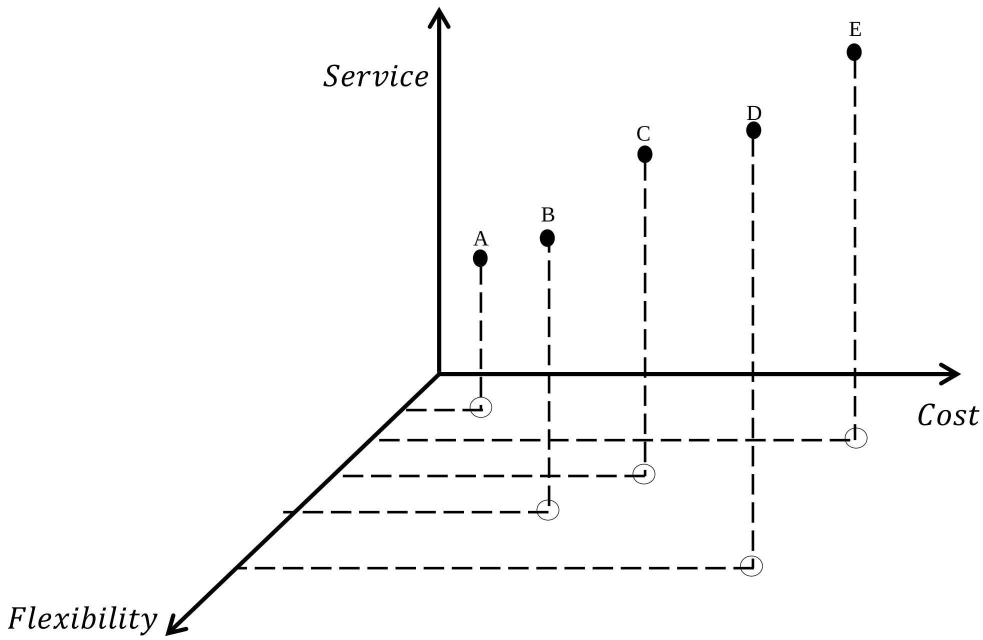


Figure 1.1 Post-optimal flexibility evaluation of the non-dominated set

Points A-E in figure 1.1 comprise the non-dominated set considering the criteria of cost and service level. Solution point D is superior from a network flexibility standpoint, as found in an analysis of the Pareto efficient set. Note that the cost minimizing and service maximizing solutions (A and E respectively) each perform worse than the compromise solutions, B-D on the metric of flexibility. The main point of this simple illustration is to highlight the fact that in discrete location, the relative similarity of a network of location decisions amongst a set of Pareto optimal configurations is a criterion that is not necessarily correlated with either service maximization or cost minimization. In fact, the performance of this criterion can vary quite widely among adjacent points on the frontier. This property is a key driver for this research, where the foundation for the quantifying of a performance metric called "network flexibility" is established.

1.3 Research Purpose

To address this consideration of network flexibility, the following question is considered: *Do subsets of Pareto optimal solutions exist which have similar network structures?* If so, which scenarios tend to generate cost efficient, highly flexible solutions? Not only would a decision maker be more reassured they selected a good compromise solution between cost and customer service, *the increase in cost for a network to improve customer service in the future while possibly maintaining Pareto optimality could be drastically reduced.* If a business sought to improve customer service in the future by altering their distribution network, this would potentially be much more costly to do so if the underlying structure of said network is inflexible in this context. If a compromise solution is selected purely based upon objective value performance, then the optimal locations themselves could very well result in a highly fragile network structure, which is costly to maintain optimality if reconfiguration is desired in the future.

A highly flexible distribution network is analogous to a manufacturing system capable of performing cost effective switches between production runs of varying types of goods. The scale and magnitude of costs are clearly different, but the concept is the same. Choosing a compromise solution can be a difficult task, but it can be simplified if the decision maker is presented with a solution that is *non-dominated and highly flexible*.

This concept of quantifying flexibility through network similarity will be investigated via a Multi-Criteria Decision Analysis (MCDA) tool. MCDA is a technique used to assist decision makers in the selection of a course of action while considering multiple criteria. The MCDA approach taken here is complementary with classical multi-objective combinatorial optimization (MOCO) techniques. A post-optimality analysis for evaluating solution alternatives on the Pareto frontier and generating a set of performance metrics reflecting the network flexibility of each solution is the primary purpose of this research.

In addition to the research quantifying the similarity of a distribution networks, a new approach for modeling service level considerations in strategic network design problems is given. This contribution expands location theory in the general area of facility location in public sector modeling. Lastly, the first implementation of a multi-objective greedy randomized adaptive search (MOG) meta-heuristic for a multi-echelon location problem (Feo & Resende 1989) is provided and implemented.

Section two briefly summarizes some key foundational theory of multi-objective optimization. Pareto optimality and the fundamental mathematics of discrete multi-objective optimization of problems are summarized. In section three, the historical techniques for incorporating service level considerations are reviewed and discussed. Additionally, a unique modeling approach for service levels in distribution location is provided with a series of mathematical formulations for a variety of strategic network design problem. The field of MCDA is discussed in section four, and metrics used to capture network flexibility are derived and explained. In section five, the algorithmic solution approaches for this work are discussed. A simple, preliminary example is provided in section six. Computational exercises and analysis is given in section seven as well as a synthesis of the proposed contributions to the literature of this research.

2. Multi-Criteria Optimization Theory

As opposed to the mono-objective optimization problem, problems with more than one objective or multi-objective optimization problems seek to minimize (or maximize) a vector of k objectives. In most cases, these objectives are conflicting where improving one objective degrades the value of others. The tradeoff inherent amongst objectives leads to an impossibility (usually) of attaining a single global optimum. This complication is addressed by generating a set of *Pareto efficient*, or *Pareto optimal* solutions which can be seen as being prefered to all others in a nonempty *feasible region* S. For this work, bold lower case notation can be assumed to be a column vector and bold upper case notation can be assumed to be a matrix of size $m \times n$ with m rows and n columns.

2.1 Multi-Objective Optimization

The notation used here is inspired by and similar to that used in both Miettinen (1999) and Collette & Siarry (2004).

The multi-objective optimization problem can be generalized as having the following form:

$$\begin{aligned} &minimize \{f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_k(\mathbf{x})\} \\ &subject\ to\ \ \mathbf{x} \in S, \end{aligned} \tag{2.1}$$

where there are $k \geq 2$ *objective functions*, and the decision variable vector $\mathbf{x} = (x_1, x_2, \dots, x_n)^T$ belongs to S which is a subset of the decision variable space R^n. In problem (2.1), the *constraint functions* defining the region of feasibility are not provided so S can be referred to hereafter in a more generalizable context. For ease of exposition regarding the mathematical notation used in this work, it is assumed that all objectives are to be minimized unless otherwise stated.

In problem (2.1), the vector of objective functions is denoted by $\mathbf{f}(\mathbf{x}) = \{\boldsymbol{f_1}(\mathbf{x}), \boldsymbol{f_2}(\mathbf{x}), \dots, \boldsymbol{f_k}(\mathbf{x})\}^T$. The image of the feasible region is a subset of the *objective space* and is denoted by $Z = \mathbf{f}(S)$. This is typically referred to as the *feasible objective region*, where $Z \subset \mathbf{R}^k$. The *objective vectors* or *criterion vectors* are denoted by $\mathbf{z} = (z_1, z_2, \dots, z_k)^T$ or $\mathbf{f}(\mathbf{x})$, where $f_i(\mathbf{x}) = z_i \;\; \forall\, i = 1,2, \dots, k$. The elements of Z are referred to as the *criterion values* or the *objective function values*.

To provide more clarity as to the structure of S, it is convenient to further refine problem (2.1) in the following vectored notation:

$$\begin{aligned} &minimize\; \vec{f}(\mathbf{x}) \\ &s.t. \qquad \vec{g}(\mathbf{x}) \leq 0 \\ &\qquad\quad\;\; \vec{h}(\mathbf{x}) = 0 \\ &\text{where } \mathbf{x} \in \mathbf{R}^n, \vec{f}(\mathbf{x}) \in \mathbf{R}^k, \vec{g}(\mathbf{x}) \in \mathbf{R}^m \text{ and } \vec{h}(\mathbf{x}) \in \mathbf{R}^p \end{aligned} \qquad (2.2)$$

Problem 2.2 seeks to minimize a *criterion vector* of size k manipulating n decision variables subject to vectors of m *inequality constraints* and p *equality constraints*. Constraints of the form $\vec{h}(\mathbf{x}) = 0$ are often referred to in the literature as *active constraints*.

2.2 Pareto Optimality

The origins of the field of multi-objective optimization can be traced back to the ideas of Irish philosopher and economist, Francis Ysidro Edgeworth, and Italian engineer and economist, Vilfredo Pareto. In 1881, Edgeworth described the concept of isolating criterion vectors of interest where no components or objective values can be improved without simultaneously deteriorating at least one other (Edgeworth 1987). Pareto developed the concept of domination and Pareto efficiency building upon the ideas of Edgeworth in 1896 (Pareto (1964, 1971).

Because attaining global optimality in most MOCO problems is generally not feasible, one must restrict the focus to identifying a subset of decision vectors in *S* which are better than the others. These "prefered" solutions map to objective values which are *Pareto efficient*, and they can be found on the leading edge of the *k* dimensional solution space. Solutions on this leading edge that are *non-dominated* are called *Pareto optimal*. The domination relation defined below is the key to identifying the subset of solutions which are Pareto Optimal.

> Definition 1. *Domination Relation*
> A vector $\mathbf{x}_1$ $dominates\ solution\ vector$ $\mathbf{x}_2$ if the following property holds:
> $\mathbf{x}_1 is\ at\ least\ as\ good\ as\ \mathbf{x}_2\ for\ all\ the\ objectives$ and
> $\mathbf{x_1} is\ strictly\ better\ than\ \mathbf{x}_2\ for\ at\ least\ one\ objective\ in\ \mathbf{f}(\mathbf{x})$.

The domination relation and Pareto optimality go hand in hand, as seen in definition 2, which provides a formal definition for global Pareto Optimality.

> Definition 2. *Global Pareto Optimality*
> A decision vector $\mathbf{x}^* \in S\ is\ Pareto\ optimal$ if there does not exist another decision vector $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)\ \ \forall\ i = 1, 2, \ldots, k$ and
> $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)\ for\ at\ least\ one\ index\ j \neq i$.

The globally Pareto optimal set contains the set of *prefered alternatives* in *Z*. However, there are also some solution values on the leading edge which do not belong to the Pareto optimal set yet still exist on the frontier. These objective values are often refered to as *weak Pareto optimal* solutions. The Pareto optimal set is a subset of the weak Pareto optimal set, meaning that the entire leading edge in solution space is weakly Pareto optimal. Definition 3 formally defines weak Pareto optimality.

Definition 3. *Weak Pareto Optimality*
A decision vector $\mathbf{x}^* \in S$ *is weakly Pareto optimal* if there does not exist another decision vector $\mathbf{x} \in S$ such that $f_i(\mathbf{x}) < f_i(\mathbf{x}^*)\ \ \forall\, i = 1, 2, \ldots, k$.

Figure 2.2.1 presents an example mapping of a two variable constraint defined feasible space to an objective space of dimension $\mathbf{R}^2$.

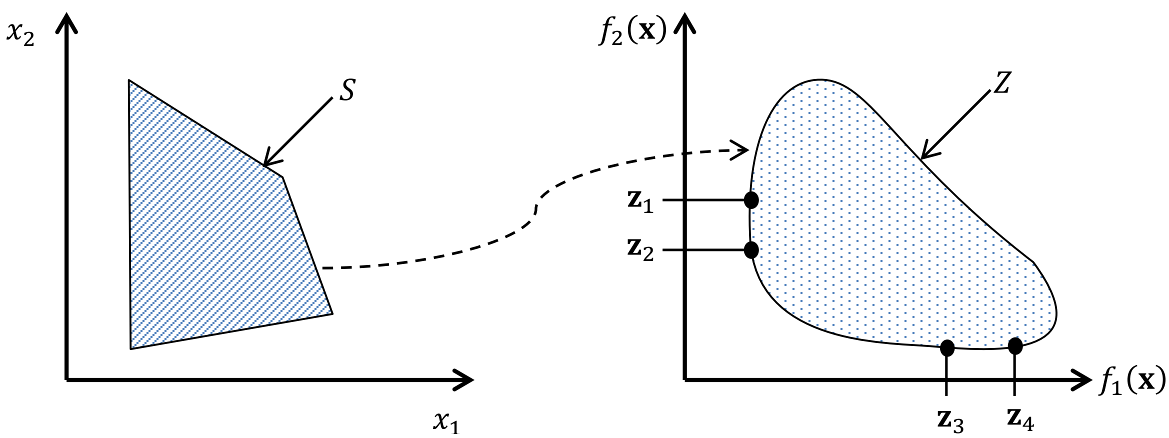


Figure 2.1 The feasible region *S*, its mapping to *Z*, and the Pareto frontier

If the case where minimization is sought for both objectives, the entire Pareto frontier depicted in Figure 2.1 is identified by the edge in *Z* space closest to the origin connecting solutions $\mathbf{z}_1$ and $\mathbf{z}_4$. The $(\mathbf{z}_2, \mathbf{z}_3)$ edge of the frontier represents the set of Pareto optimal solutions. The solutions on the frontier from $\mathbf{z}_3$ (but not including) to $\mathbf{z}_4$ are only weakly Pareto optimal and are therefore dominated by the Pareto optimal edge. The same applies to the solutions between $\mathbf{z}_2$ (but not including) and $\mathbf{z}_1$ as they are also merely weakly Pareto optimal.

3. Bi-Criteria Distribution Network Design

Location problems are characterized by a space or geographic region, a distance metric, and a set of known points. In discrete problems, these points typically represent demand originating sites and/or established facilities in a system or candidate facility locations. The manner in which demand is represented in the problem, either discretely, stochastically, or as a continuous function of density over an area is another defining trait of any location problem. Lastly, the facilities themselves can be sited at discrete candidate locations, or continuously at derived locations on a plane. The decision maker's task is to locate at least one new facility and allocate demand such that a desired objective is achieved, whether that may be the minimization of costs or a surrogate measure of costs, the maximization of a customer service metric, or something else entirely.

Since the seminal paper Geoffrion and Graves (1974), multi-echelon distribution location problems have been studied with ever increasing interest in the literature. Recently, several surveys have been provided on the topic or a related area recently; Bilgen & Ozkarahan (2004), Klose & Drexl (2005), Sahin & Sural (2007), Shen (2007), Melo et al. (2009), and Farahani et al. (2014). This indicates that modeling distribution networks continues to be a thriving area of research. Additionally, multi-criteria approaches have been consistently investigated for these problems throughout the years, as evidenced by the following surveys on the subject: ReVelle et al. (1981), Current et al. (1990), Nickel et al. (2005), and Farahani et al. (2010).

The problem of designing a distribution network is typically approached as a facility location problem. In these types of models, a series of location decisions reflecting the opening of a facility and the assignment of demand is made. The production-distribution hierarchical location problem and the strategic network or supply chain design problem are two different modeling approaches to distribution network design, with supply chain papers typically including additional decisions beyond location-allocation.

In this chapter, the seminal papers from location theory pertinent to this discussion will be highlighted, and the literature on multi-criteria facility location in distribution network design is synthesized and summarized. In addition, a series of discrete facility location problems will be provided throughout as well as a discussion on modeling service levels in distribution network design. The notation used in this chapter is based on Daskin (2013).

3.1 Model Assumptions

Several simplifying assumptions are required by the modeling approach taken in this work. A brief discussion of the more prominent assumptions required in discrete facility location follows.

- Static Model

Many real world scenarios in facility location are not static in nature. Therefore, non-dynamic modeling approaches to strategic location problems can signify a simplifying assumption in the form of static model parameters.

- Deterministic Parameters.

Some model parameters in most strategic network design problems are inherently stochastic. Deterministic approaches are usually a simplification.

- Single, Homogenous Product.

Very few real world distribution systems produce, move and store a single product. For this reason, approaches which model a single aggregated product is a significant simplification.

- Inelastic Demand

In many real world scenarios, demand is responsive to the level of service provided. For this work however, an assumption of inelastic demand is made. However, the development of profit-based objectives incorporating the elasticity of demand for distribution systems is an especially interesting area of future research.

- Closest Facility Assignments

In uncapacitated facility location modeling, demands are typically assigned to the closest open facility. For this work, this assumption will be dropped for some models. Multiple echelon distribution systems exhibit interesting properties in regards to closest assignment. This is discussed in depth later in this chapter.

- Demand Allocation Made in Isolation for each Demand-Facility Pair

Often, subsets of demand points require service from the same facility. Therefore, this is usually a simplifying assumption in many real world problems.

- Uncapacitated Models

For this research, capacity considerations will not be considered. This can be seen as a simplifying assumption for some real world scenarios. However, the interest here is strictly limited to the spatial impact of location decisions in distribution networks on cost and service level. For this reason, the arbitrary incorporation of capacity considerations is not necessary. Additionally, adding capacity can lead to seemingly strange customer facility assignments and can significantly increase the difficulty of solving these problems, (Waston et al 2013). Capacity considerations in this stream are an area of future research.

- Direct Shipment

In the work provided here, multi-stop routes or other more sophisticated distribution strategies are not modeled. The simplifying assumption is that transportation costs can be adequately approximated via a linear function of distance between a customer and their assigned facility.

- No Economies of Scale Shipment Savings

The per unit distribution cost differences are minimal in the work, reflecting that there is no economies of scales savings on any long haul arcs due to varying modes or cost structures.

3.2 Background

The location of warehouses in distribution networks has been studied since Kuehn & Hamberger (1963). The *P*-Median location problem is the forbearer of discrete modeling approaches in strategic network design. The seminal paper Hakimi (1964) originally defined the *P*-Median problem on a network seeking to locate *P* facilities such that the total demand-weighted distance to service all customers is minimized in a graph theoretic contribution. An integer programming formulation of the problem is provided in ReVelle & Swain (1970) and has been widely used ever since. This formulation is provided below.

Inputs

h_i = demand at node $i \in I$

d_{ij} = distance between demand node $i \in I$ and candidate facility location $j \in J$

P = number of facilities to locate

Decision Variables

$$X_j = \begin{cases} 1 \text{ if location } j \in J \text{ is selected} \\ 0 \text{ otherwise} \end{cases}$$

$$Y_{ij} = \begin{cases} 1 \text{ if demand at node } i \in I \text{ is served by a facility at } j \in J \\ 0 \text{ otherwise} \end{cases}$$

Using this notation, the *P*-median problem is formulated as follows:

Minimize $\sum_{i \in I} \sum_{j \in J} h_i d_{ij} Y_{ij}$ (3.1)

Subject To: $\sum_{j \in J} Y_{ij} = 1$ $\forall i \in I$ (3.2)

$\sum_{j \in J} X_j = P$ (3.3)

$Y_{ij} \leq X_j$ $\forall i \in I; j \in J$ (3.4)

$X_j \in \{0,1\}$ $\forall j \in J$ (3.5)

$Y_{ij} \in \{0,1\}$ $\forall i \in I; j \in J$ (3.6)

Objective (3.1) minimizes the total demand-weighted distance between the demand nodes and the closest facility. Constraint (3.2) forces all demands to be satisfied. Constraint (3.3) states that exactly *P* facilities are to be located. Constraint (3.4) prevents the assignment of any demand to a facility that is not selected. Constraint (3.5) and (3.6) are binary restrictions on the location and customer assignment decision variables.

This model tends to locate facilities close to large clusters of demand. Therefore, the minimization of the weighted average distance (*P*-median) performs well on the metrics of transportation costs and aggregate customer service in the form of average response time. For this reason, the *P*-Median facility location model remains today as one of the most useful approaches to network design It is often the best starting point, or all that is needed when considering a strategic network design or supply chain design problem (Watson et al. 2013). Secondly, this approach highlights the value of additional facilities. As the number of facilities to locate *P* is increased, a decreasing marginal improvement in transportation costs and customer service can be seen. However, as pointed out in Daskin (2013), the marginal improvement in demand weighted total distance isn't always strictly monotonically decreasing. This characteristic is present in most real networks.

In the case where a direct consideration of the fixed facility location costs is essential to incorporate in a modeling approach, the simple facility location model or the uncapacitated facility location problem (UFLP) can be considered. It is usually prudent to do so when the differences in the costs to open a facility differ significantly across candidate locations. Otherwise, there is no need to include this cost parameter and complicate the model unnecessarily. The reason for this is that the sum product of fixed costs and location variables would be roughly proportional to the total number of facilities chosen, in which case, it would be beneficial to examine the demand weighted total distance at varying levels of the amount of total locations to select.

Determining the number and optimal location of warehouses in a distribution network was first considered in Kuehn & Hamberger (1963). Efroymson & Ray (1966) expanded upon this work and provided a branch and bound algorithm for the optimal location of factories. Finally, Erlenkotter (1978) is considered a seminal paper in this field as well. In that work, the authors provide an efficient dual-based procedure which is still heavily used today. The formulation for the UFLP is given below.

Inputs

f_j = annualized fixed cost of opening a facility at $j \in J$

c_{ij} = cost to service demand $i \in I$ from candidate facility location $j \in J$

Decision Variables

Y_{ij} = proportion of demand at node $i \in I$ served by a facility at $j \in J$

Using this additional notation, the UFLP is formulated as follows:

Minimize $\sum_{j \in J} f_j X_j + \sum_{i \in I} \sum_{j \in J} c_{ij} Y_{ij}$ (3.7)

Subject To: $\sum_{j \in J} Y_{ij} = 1 \quad \forall i \in I$ (3.8)

$Y_{ij} \leq X_j \quad \forall i \in I; j \in J$ (3.9)

$X_j \in \{0,1\} \quad \forall j \in J$ (3.10)

$Y_{ij} \geq 0 \quad \forall i \in I; j \in J$ (3.11)

Objective (3.7) minimizes the total costs. Constraint (3.8) forces all demands to be satisfied. Constraint (3.9) prevents the assignment of any demand to a facility that is not selected. Constraint (3.10) is a binary restriction on the location variables. Constraint (3.11) enforces non-negativity on the demand allocation variables. Note that due to constraint (3.8), the demand allocation variables Y_{ij} have an upper bound of 1.

In the UFLP, the tradeoff inherent in fixed facility location costs and variable transportation costs are considered directly in the objective. As the number of facilities is increased, transportation costs decrease. Usually, the optimal solution for the UFLP is one which balances this tradeoff.

The previous models are both concerned with cost based performance metrics (or a surrogate measure of cost). In network design problems, customer service is an oft overlooked factor in modeling distribution systems. For this work, the metric of customer service is considered directly in a multi-criteria framework, with the intent of adequately capturing and analyzing the trade-off between service level and cost in distribution system design.

An alternate approach to incorporating customer service restrictions on a distribution problem would be to simply add a maximum service level constraint to the model. An early paper taking this approach is Holmes et al. (1972), and an old review on these modeling approaches is given by Moon & Chaudry (1984).

These types of models are usually referred to as time definite or maximum service restriction problems. Using the previously defined notation, a service level constraint for the P-Median problem is provided below.

$$d_{ij}Y_{ij} \leq d_c \qquad \forall i \in I; j \in J \tag{3.12}$$

where d_c is a maximum coverage distance. In concert with constraint (3.2), this approach forces all customers to be assigned to a facility within a certain distance, mandating a maximum response time. When incorporated in a P-Median problem, this effectively minimizes total demand weighted distance given a minimum service level for all demands. This is essentially a single iteration of a multi-criteria scalarization technique resulting in the identification of a single point on the trade-off frontier between cost and customer service. In this work, a broader viewpoint is considered, where the entire Pareto frontier is of interest.

Before proceeding further with the historical coverage based modeling approaches, a brief discussion of the various factors influencing service level in private sector distribution systems follows.

- Proximity to customer

The proximity of a distribution location to a customer is the primary determinant of service level, (Murphy & Wood 2011). The approach taken in this work is to focus solely on this contributing factor of service level via the use of discrete location models.

- Inventory levels

Lead time to customer is heavily impacted by the inventory levels at stocking locations throughout a distribution system. Generally, higher inventory levels lead to increased levels of customer responsiveness.

- Reorder Policies

The reorder policy or strategy at any given echelon of a distribution system also impacts the customer responsiveness of the system. The reason for this is that the reorder policy of a product determines the stocking levels of a product over a time horizon, which in turn affects customer responsiveness/service level.

- Model Choice

The mode of transportation used to deliver product to a customer is a factor that has a massive impact on customer service level. After all, what is reachable by a plane in one day is much larger than the distance achievable by an automobile in a day.

- Direct Shipment from Warehouses

A common assumption in location allocation models in distribution networks is that demands can only be serviced by warehouses. In a multi-echelon problem when demand can potentially be serviced by different types of facilities (like plants for example), this is a rather restrictive assumption. In some models considered in this work, this assumption will be dropped, and customers can be serviced by plants or warehouses.

- Employee Training

Many intangible factors also contribute to customer service level. The level of training or experience level of the employees affecting the movement and storage of goods and materials is one of the more prominent of these intangible factors.

The use of covering based objectives in this work is driven by the fact that in most private sector distribution problems, service levels are naturally partitioned by delivery windows, usually in days. What this means is that customers (retailers) don't necessarily care if they can receive their shipment within one hour of issuing an order or one day. To them, service level is essentially one day. This factor naturally leads to the performance metric of customer service to be measured in days elapsed since the issuance of an order. Provided below is an example illustration from Murphy & Wood (2011) depicting varying service levels in distribution.

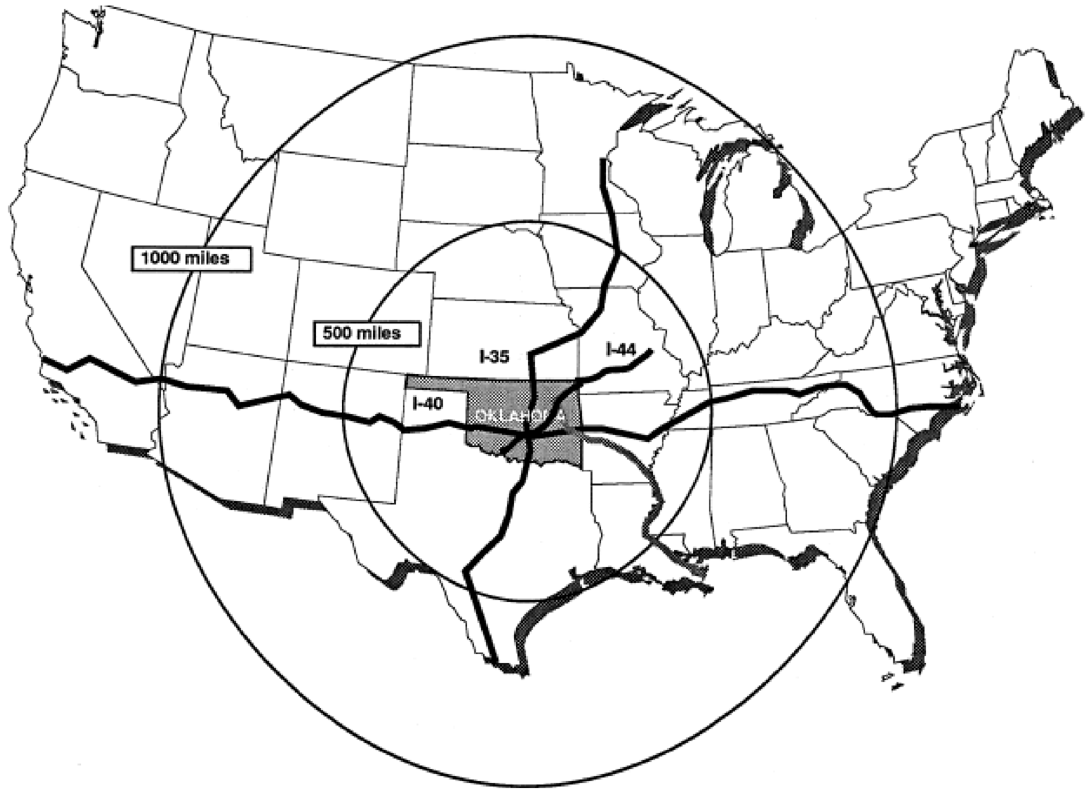


Figure 3.1 Distribution radii of a warehouse in Central Oklahoma

Depicted above in figure 3.1 is an example of one and two day delivery zones for a hypothetical warehouse located in Central Oklahoma. Under the assumption that any demand within the radius of 500 miles from this location has an adequate road network such they can be reached in time, their service level is effectively measured at one day. Therefore, the service level to a customer located at Tulsa, Oklahoma is the essentially the same as a customer located in Omaha, Nebraska, because both locations can be reached in one day. However, a customer located at Atlanta, Georgia would receive a lower level of service if they received product from the same location. It is this reality in physical distribution which led us to take coverage based approaches in this work, where a customer is covered if they are within a given distance from their assigned facility.

Another important consideration leading to the use of coverage base approaches is the recent emphasis on same day delivery in business. The explosion of ecommerce has fueled the idea of reducing the time elapsed between order and delivery to point where it is a heated area of competition. Covering based modeling approaches in facility location can be used to help address these considerations in distribution system design.

In covering models, there are two core approaches. The first is set covering, or what is called the set covering location problem (SCLP). Once again, the initial discovery of this location problem is attributed to Seifollah Louis Hakimi in his seminal graph theoretic work Hakimi (1965). An integer programming formulation of this problem was given in Toregas et al. (1971). This model is provided below.

Inputs

$$a_{ij} = \begin{cases} 1 \text{ if candidate facility } j \in J \text{ can cover demand } i \in I \\ 0 \text{ otherwise} \end{cases}$$

Using this additional notation, the SCLP is formulated as follows:

Minimize $\sum_{j \in J} f_j X_j$ (3.13)

Subject To: $\sum_{j \in J} a_{ij} X_j \geq 1 \quad \forall i \in I$ (3.14)

$X_j \in \{0,1\} \quad \forall j \in J$ (3.15)

Objective (3.13) minimizes the fixed costs of all selected facilities. Constraint (3.14) forces all demands to be covered at least once. Constraint (3.15) is a binary restriction on the location variables.

The set covering location model does not consider the assignment of demands to locations. In scenarios where capacity is a concern, other approaches modeling the decision of customer assignment may be necessary. For ease of further discussion, the binary coverage matrix is redefined below using the previously given notation.

$$a_{ij} = \begin{cases} 1 \text{ if } d_{ij} \le d_c \\ 0 \text{ otherwise} \end{cases}$$

This implies that the number of facilities needed in any given realization of a SCLP is controlled to a large extent by the maximum coverage distance parameter d_c. As this value increases, the number of facilities needed to cover all demands decreases, and vice versa. Note that if the cost to locate a facility is the same for all candidate locations $j \in J$, the fixed cost parameters f_j may be dropped from the model.

In the SCLP, all customers must be served within a maximum service level and transportation costs are largely ignored. However, similar to the relationship of the *P*-Median versus the UFLP, it may be more beneficial to examine the incremental impact of adding facilities to the network, or if budgetary concerns are present restricting the maximum number of facilities required. These issues were addressed in the maximal covering location problem (MCLP) in Church & ReVelle (1974). This model is provided below.

Decision Variables

$$Z_i = \begin{cases} 1 \text{ if demand node } i \in I \text{ is covered} \\ 0 \text{ otherwise} \end{cases}$$

Using this additional notation, the MCLP problem is formulated as follows:

Maximize $\sum_{i \in I} h_i Z_i$ (3.16)

Subject To:

$Z_i \le \sum_{j \in J} a_{ij} X_j \quad \forall i \in I$ (3.17)

$\sum_{j \in J} X_j \le P$ (3.18)

$$X_j \in \{0,1\} \qquad \forall j \in J \qquad (3.19)$$

$$Z_i \in \{0,1\} \qquad \forall i \in I \qquad (3.20)$$

Objective (3.16) maximizes the amount of covered demand. Constraint (3.17) states that a customer cannot be covered if a facility is not within the coverage radius. Constraint (3.18) disallows more than *P* facilities to be opened. Constraints (3.19) and (3.20) are binary restrictions on the location and coverage variables.

The MCLP maximizes the amount of demand covered given that no more than *P* facilities can be opened. Like the SCLP, the coverage matrix a_{ij} is heavily dependent upon the allowable coverage distance d_c. An alternative formulation for this problem can be given by minimizing the uncovered demand, which results in a model structure that is very similar to the *P*-Median problem, Daskin (2013). This model is given below.

Decision Variables

$$W_i = \begin{cases} 1 \text{ if demand node } i \in I \text{ is not covered} \\ 0 \text{ otherwise} \end{cases}$$

Using this additional notation, the MCLP problem can be reformulated as follows:

Minimize $\sum_{i \in I} h_i W_i$ (3.21)

Subject To:

$$\sum_{j \in J} a_{ij} X_j + W_i \geq 1 \qquad \forall i \in I \qquad (3.22)$$

$$\sum_{j \in J} X_j \leq P \qquad (3.23)$$

$$X_j \in \{0,1\} \qquad \forall j \in J \qquad (3.24)$$

$$W_i \in \{0,1\} \qquad \forall i \in I \qquad (3.25)$$

Objective (3.21) minimizes the amount of uncovered demand. Constraint (3.22) states that if a customer cannot be covered by a facility within the allowable radius, then the uncoverage variable is positive. Constraint (3.23) disallows more than *P* facilities to be opened. Constraint (3.24) and (3.25) are binary restrictions on the location and uncoverage variables.

A key difference to note between the SCLP and the MCLP is that in the SCLP, no individual demand is favored amongst any other demand. Therefore, the magnitudes of the demands have no influence on the optimal location decisions.

The models given in this section provide the foundation for the approaches taken in this work. All subsequent models presented here are based upon these seminal papers. What follows is a survey of the literature on multi-objective distribution system design problems.

3.3 Literature Review

In this review, the literature highlighting the multi-objective approaches to facility location in distribution system design will be reviewed or discussed, with a focus on multi-echelon systems consisting of at least three levels (usually plants, distribution centers or warehouses and demands or customers). These papers include those works categorized as facility location in supply chain design and hierarchical production-distribution location.

Multi-objective approaches to facility location have been consistently pursued since the inception of the field. ReVelle et al. (1981), Current et al. (1990), Farahani et al. (2010) are the prominent surveys on the subject. The role of facility location in supply chain design was also recently reviewed by Melo et al. (2009). What follows is a review and categorization of those multi-objective facility location problems in supply chain design.

3.3.1 Multi-Objective, Multi-Echelon Supply Chain Design Papers

This portion of the literature is targeting research classified as being multi-objective supply chain design for multi-echelon systems. The applicable papers found in the existing literature surveys on the subject of location in supply chains were included as, well as a total of 42 additional works identified. These papers are primarily applied, featuring scenarios across a number of industries from typical retail networks to biofuel supply chains.

The most commonly used problem structure in the literature on facility location in supply chain design features deterministic parameters, a single commodity, and a single time period. Table 3.1 categorizes the selected literature according to the number of location echelons, data type, number of products and time periods, and the inclusion of reverse logistics activities. The papers are further partitioned within the table by their respective number of location echelons.

Table 3.1: Problem Structure

The number of location echelons, data type (deterministic/stochastic), amount of products and time periods (single/multiple), and whether reverse logistics is present

Article	Location Echelons	Data	Products	Periods	Reverse
Altiparmak et al. (2006)	1	D	S	S	N
Azaron et al. (2008)	1	S	M	S	N
Cardona-Valdes et al. (2011)	1	S	S	S	N
Cardona-Valdes et al. (2014)	1	S	S	S	N
Dehghanian & Mansour (2009)	1	D	S	S	Y
Du & Evans (2008)	1	D	M	S	Y
Farahani & Asgari (2007)	1	D	S	S	N
Hertwin et al. (2014)	1	D	S	S	N
Hugo & Pistikopoulos (2005)	1	D	M	M	N
Jabal-Ameli & Mortezaei (2011)	1	D	S	S	N
Liao et al. (2011)	1	S	M	S	N
Makui et al. (2006)	1	D	S	S	N
Melachrinoudis et al. (2005)	1	D	S	S	N
Olivares-Benitez et al. (2012)	1	D	S	S	N
Olivares-Benitez et al. (2013)	1	D	S	S	N
Paksoy & Chang (2010)	1	D	S	M	N
Pinto-Varela et al. (2008)	1	S	M	S	N
Wang et al. (2011)	1	D	M	S	N
Yazdian & Shahanaghi (2011)	1	S	S	S	N
You & Grossman (2011a)	1	S	S	S	N
Amin & Zhang (2013)	2	S	M	S	Y
Bhattacharya & Bandyopadhyay (2010)	2	D	M	M	N
Chen et al. (2007)	2	S	M	M	N
Giarolo et al. (2011)	2	D	M	M	N
Guillen et al. (2005)	2	S	M	M	N
Kim & Moon (2008)	2	S	M	S	N
Ramudhin et al. (2010)	2	D	M	S	N
Rezazadeh & Farahani (2010)	2	D	M	S	N
Selim & Ozkarahan (2008)	2	S	M	S	N
Xu et al. (2008)	2	S	S	S	N
You & Grossman (2011b)	2	S	M	M	N
You et al. (2011)	2	D	M	M	N
Chaabane & Paquet (2010)	>2	D	M	M	Y
Erkut et al. (2008)	>2	D	S	S	Y
Fonseca et al. (2010)	>2	S	M	S	Y
Hiremat et al. (2013)	>2	D	M	S	N
Khajavi et al. (2011)	>2	D	S	M	Y
Pati et al. (2008)	>2	D	M	S	Y
Pishvaee & Torabi (2010)	>2	S	S	M	Y
Pishvaee et al. (2010)	>2	D	S	S	Y
Ramezani et al. (2013)	>2	S	S	S	Y
Wang et al. (2013)	>2	D	S	S	Y

Roughly half of the articles reviewed are single location echelon problems. About a third of the articles site facilities in two echelons, the majority of them being production facilities and DCs. Of these two location echelon papers, all but one of them also accounted for multiple product types. Ten of the 42 articles (20%) located facilities in more than two echelons. Every one of these papers included reverse logistics activities except Hiremat et a. (2013).

About 40% of the surveyed articles incorporated stochastic parameters in their modeling approach. This is a surprising finding, which may suggest a general trending away from deterministic modeling approaches toward stochastic ones, as far as MO problems are concerned. Another interesting finding was that half of the papers model multiple product types. Only 11 of the 42 articles were dynamic models. Additionally, only about 28% of the papers were deterministic, single product type, and single period models. This finding suggests that most MO supply chain design papers, contrary to their single objective counterparts, tend to incorporate other more realistic, complicating factors in their models.

The objectives considered in these papers were categorized into five different types: cost, profit, customer service, environmental, and other. Table 3.2 displays the selected articles and their respective objectives. The papers are further subdivided into bi-objective and multi-objective papers.

Table 3.2: Objectives Considered

Article	Cost	Profit	Service	Environmental	Other
Bi-Objective					
Amin & Zhang (2013)	✓			✓	
Cardona-Valdes et al. (2011)	✓		✓		✓
Cardona-Valdes et al. (2013)	✓		✓		
Chaabane & Paquet (2010)	✓			✓	
Du & Evans (2008)	✓		✓		
Farahani & Asgari (2007)	✓				✓
Fonseca et al. (2010)	✓				✓
Giarolo et al. (2011)		✓		✓	
Hertwin et al. (2014)	✓		✓		
Hugo & Pistikopoulos (2005)		✓		✓	
JabalAmeli & Mortezaei (2011)	✓✓				
Khajavi et al. (2011)	✓		✓		
Kim & Moon (2008)	✓				✓
Liao et al. (2011)	✓		✓		
Olivares-Benitez et al. (2012)	✓		✓		
Olivares-Benitez et al. (2013)	✓		✓		
Pinto-Varela et al. (2008)		✓		✓	
Pishvaee & Torabi (2010)	✓		✓		
Pishvaee et al. (2010)	✓		✓		
Ramudhin et al. (2010)	✓			✓	
Rezazadeh & Farahani (2010)	✓		✓		
Wang et al. (2011)	✓			✓	
Xu et al. (2008)	✓		✓		
Yazdian & Shahanaghi (2011)	✓				✓
You & Grossman (2011a)	✓		✓		
You & Grossman (2011b)	✓	✓	✓		
Multi-Objective					
Altiparmak et al. (2006)	✓		✓		✓
Azaron et al. (2008)	✓				✓✓
Bhattacharya & Bandyopadhyay (2010)	✓✓				
Chen et al. (2007)	✓		✓		✓✓
Dehghanian & Mansour (2009)		✓		✓	✓
Erkut et al. (2008)	✓			✓	✓✓✓
Guillen et al. (2005)		✓	✓		✓
Hiremath et al. (2013)	✓		✓		✓
Makui et al. (2006)	✓✓✓✓				✓✓
Melachrinoudis et al. (2005)	✓		✓		✓
Paksoy & Chang (2010)	✓✓✓				
Pati et al. (2008)	✓			✓	✓
Ramezani et al. (2013)	✓		✓		✓
Selim & Ozkarahan (2008)	✓✓		✓		
Wang et al. (2013)	✓			✓✓	
You et al. (2011)	✓			✓	✓

As seen in Table 3.2, every paper included at least one economic objective, with the majority of them considering costs. The most noteworthy of these papers are those that suggest that an integrated total cost function may not be ideal. For example, in Bhattacharya & Bandyopadhyay (2010), the authors consider an entire location echelon's cost separately from total cost. They show that this method results in lower total costs as opposed to minimizing an aggregate total cost function.

Half of the selected articles incorporated a service level objective. There is a wide variety of types of objectives considered in this category. The different objectives considered as being service oriented are maximize volume fill rate, maximize the amount of demand served, maximize supply chain responsiveness, minimize lateness of delivery, minimize maximum lead time, enforce a guaranteed service level, minimize cycle tardiness, minimize delivery tardiness, and minimize total transportation.

Twelve papers included environmentally oriented objectives as a part of their model. These papers generally sought to minimize damages to the environment associated with supply chain activates via the reduction of greenhouse gases (GHG). However, in Pati et al. (2008), the objective was to maximize the amount of recyclable material collected in a reverse logistics network, and in Wang et al. (2013), energy consumption and waste generation was minimized.

Finally, 17 papers featured objectives that didn't fall neatly into any of the other four categories. Some of these objectives are risk or robustness based, while others, like Fonseca et al. (2010) sought obnoxious facility location based objectives for the location of refuse or collection centers in reverse logistics.

3.3.2 Multi-Objective Hierarchical Production-Distribution Location Papers

Location decisions of hierarchical systems of facilities require the consideration of several aspects not found in single echelon location problems. This distinction is primarily due to the added complexity of multiple facility types capable of providing varying services to demands. Typically, a separate location echelon is modeled for each facility type with categories based upon the variety of service available. There have been several reviews on the subject with Sahin & Sural (2007) and Farahani et al. (2014) being the latest.

Several classification schemes for the problem have been proposed. The scheme employed here is consistent with Sahin & Sural (2007), which features four distinct characteristics or attributes. This categorization scheme builds upon that proposed in Narula (1982) and Daskin (1995). These four problem defining attributes are *flow pattern* or *flow discipline*, *service varieties*, *coherency*, and *objective*. Before categorizing the selected papers, a detailed description is provided for each of these attributes.

The flow discipline of a hierarchical location problem describes the manner in which demand is satisfied via the movement of goods or customers. There are two types of flow discipline, *single-flow* and *multi-flow*. In single-flow systems, flow can either originate from the demands or the highest location level in a given problem. In multi-flow systems, demand can be directly satisfied by more than one location echelon. When the direction of travel of a flow originates from the demand nodes, this typically represents customers visiting service facilities in public sector location problems. For example, the referral of a customer in a hospital system to a larger facility featuring a greater service variety often occurs. It is important to note that this referral behavior in service hierarchies is not generalizable to all demands and usually applies to a subset of customers. Reverse logistics in distribution is a private sector example of product flow originating from the demands. A production distribution logistics network can feature either single-flow or multi-flow behavior. Both single-flow and multi-flow systems will be examined in this dissertation.

The service variety of a problem can be described as either being *nested* or *non-nested*. This distinction refers to the specific types of demand, if there are varying types of demand, any given location echelon is capable of satisfying. If higher level facilities in a problem scenario are capable of providing all the goods or services of lower level facilities and at least one additional good or service, then the system is classified as being nested. A hierarchical system is classified as being nested if it can offer all of the services of lower echelon locations. The US postal system is an example of a nested system of hierarchical facilities, where the lowest echelon consists of customer mailboxes and the higher echelons are postal centers and local mail depository boxes.

Production-distribution networks with multiple goods can be either nested or non-nested, depending upon the product variety, their production locations, and/or stocking points. Additionally, the flow discipline of a system affects to a large degree the service variety of a system. Multi-flow systems with flow originating from either the demands or the highest location echelon can be considered as being nested or non-nested. However, if the flow is originating from the highest echelon in a single flow system, like the typical production-distribution location problem, that system can never be nested despite the product variety offered. This can be easily seen by noting that if demand can only be satisfied by the lowest location echelon, then a nested implementation is not possible.

This characteristic is usually restricted to the production-distribution problem, and not necessarily to the other applications in hierarchical location. For example, health care applications can be nested, single flow systems, per Marianov & Serra (2001).

A *coherent* hierarchy is one in which demands that are assigned to a particular lower level facility must be assigned to the exact same higher level facility as across all location echelons where customer demand satisfaction can occur. In other words, single sourcing restrictions at each echelon are enforced in coherent systems. The impact on network configuration this attribute has is immense. Coherent systems can have widely differing network structures compared to non-coherent systems. Either single-flow or multi-flow systems can have coherent structures. A hospital system receiving patients at smaller, satellite service facilities may refer patients to a larger facility with a wider variety of services available. This would be an example of a public sector application to a single-flow, coherent system.

Sahin & Sural (2007) categorize the various objectives used by papers in their literature review under the categories of *median*, *covering*, and *fixed charge objectives*. Obviously, this classification attribute isn't unique to the hierarchical location problem. The scenarios modeled in this dissertation focus on the trade-off of cost (median) and customer service (maximal covering) at varying levels of amount of facilities to open.

Depicted below in Figure 3.2 is an example of a single-flow production-distribution system.

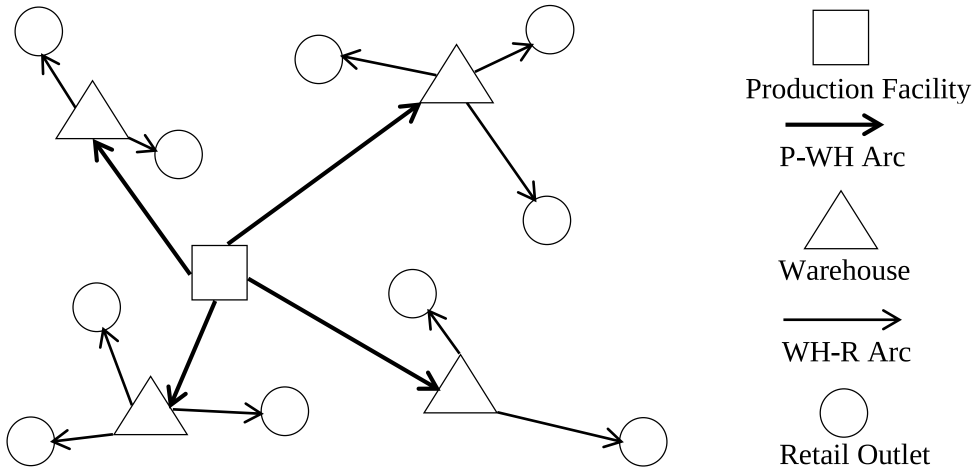


Figure 3.2 A single-flow hierarchical production-distribution system

Figure 3.2 shows a production distribution location problem modeled as a hierarchical location problem consisting of production facility and warehouse location decisions, as well as customer to retail outlet assignment decisions. The structure of this network is single-flow with no nestedness or coherency considerations. The majority of applications in the production-distribution location problem feature this scenario. However, if customer demand is allowed to be satisfied directly by a production facility, then this scenario no longer applies, as the system would be multi-flow with possible nested and/or coherent structures.

Depicted below in Figure 3.3 is an example of a multi-flow production-distribution system.

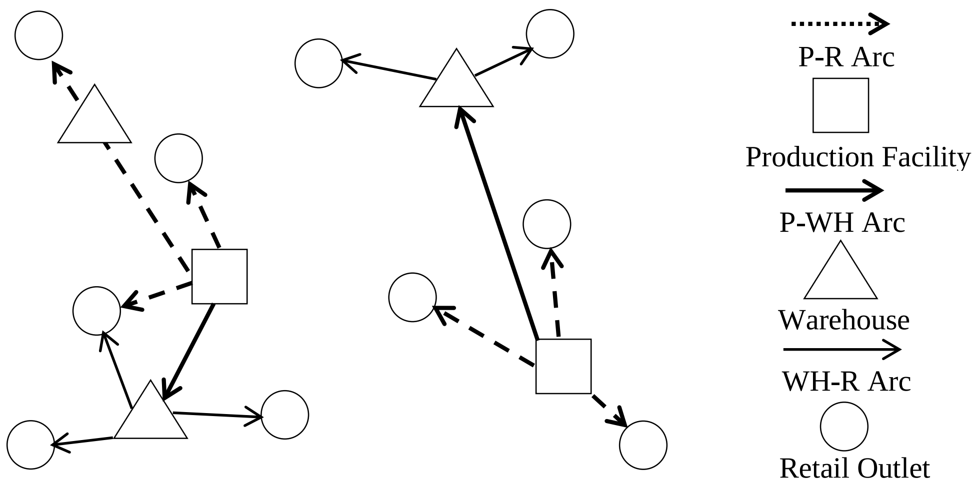


Figure 3.3 A multi-flow hierarchical production-distribution system

A multi-flow variant of the production-distribution location problem can be seen in figure 3.3. This scenario captures the additional complication that occurs when direct delivery to customers from production locations is allowed. An assumption in this case is that tax considerations or other ancillary factors that influence location decisions are not considered.

Tables 3.3-3.5 below list and categorize the production-distribution location problems, and hierarchical location problems in the literature featuring coverage based objectives or multiple objectives. The papers are described by the attributes discussed above. Flow pattern (FP) is indicated as being either S (single) or M (multiple). In the service availability column (SA), a paper is classified as being nested (N) or (-), meaning that the nestedness property was either not present or not considered. The spatial configuration column (SC) shows either (CO) for coherent or (-) for non-coherency or no consideration of this property. (LO) refers to the types of location objectives considered, with the options being fixed charge (F), coverage (CV), or multi-objective (MO). The last column is a brief description of key facets of the paper.

Table 3.3 Categorization of production-distribution hierarchical location papers

Reference	FP	SA	SP	LO	Comments
Kaufman et al. (1977)	M	-	-	F	Distribution; First assignment-based formulation; Branch and bound
Wirashinghe & Waters (1983)	M	N	CO	F	Production-distribution; Modified Kaufman et al. (1977)
Ro & Tcha (1984)	M	-	-	F	Distribution; Branch and bound
Tcha & Lee (1984	M	-	-	F	Production-distribution; Petrochemical company application
Van Roy (1989)	M	N	-	F	Production-distribution; Price elastic demand; algorithm from Dokmeci
Gao, R. (1992)	M	-	CO	F	Distribution; Primal-dual & branch and bound; 25 node problem
Barros & Labbe (1994)	S	-	CO	F	Warehouse-depot; Generalizes Tcha & Lee (1984) & Gao & Robinson
Gao & Robinson (1994)	M	-	CO	F	Distribution; Generalizes Gao & Robinson (1992)
Pirkul & Jayaraman (1996)	S	-	-	F	Production-distribution; Capacitated; 100 demand problems
Aardal et al. (1996)	S	-	-	F	Distribution; Valid inequalities and cuts; Comparison with continuous
Tragantalerngsak et al. (1997)	S	-	CO	F	Distribution; Capacitated upper facilities; 100 demand problems
Aardal (1998)	M	-	-	F	Production-distribution; Capacitated; Cutting plane; 50 demand
Pirkul & Jayaraman (1998)	S	-	-	F	Production-distribution; Single sourcing; similar to Pirkul & Jayaraman
Marin & Pelegrin (1999)	S	-	-	F	Distribution; Capacitated; Compares flow and assignment formulation
Hinojosa et al. (2000)	S	-	-	F	Production-distribution; Multi-period opening and closing decisions
Kantor & Peleg (2006)	S	-	CO	F	Distribution; Steiner tree; Approximation algorithms
Asadi et al. (2008)	S	-	CO	F	Distribution; Outlier demands not served; Approximation algorithms
Mo et al. (2011)	M	-	-	F	Production-distribution location; Two stage Stochastic program
Litvinchev & Espinosa (2012)	S	-	-	F	Production-distribution; Lagrangian relaxation; Up to 200 demand

The production-distribution hierarchical location papers can be seen in table 3.3. There is a relatively even mix of single-flow versus multi-flow papers, but there are very few nested applications. Additionally, only a handful of papers consider or enforce coherency in their networks. The classic fixed charge objective has been the only criteria considered for all of the production-distribution papers in the hierarchical location literature. The majority of these papers feature a single product, which explains the lack of nestedness being an explicit modeling consideration. Table 3.4 provides a categorization of the covering-based hierarchical location papers.

Table 3.4 Categorization of covering-based hierarchical location papers

Reference	FP	SA	SP	LO	Comments
Charnes & Storbeck (1980)	M	N	-	CV	EMS; 16 node problem; Goal programming
Moore & ReVelle (1982)	M	N	-	CV	Education; Uncoverage Cost; Binary model with LP relaxation; Honduras
Ruefli & Storbeck (1982)	M	N	-	CV	EMS; Similar to Charnes & Storbeck (1980); behaviorally linked systems
Church & Eaton (1987)	M	N	-	CV	Health care; Survey included; Referral systems; Zarzal, Columbia
Desai & Storbeck (1988)	M	N	CO	CV	Behaviorally and technologically linked systems; 21 node problem
Vernekar et al. (1990)	S	-	CO	CV	Computer networks; Horizontal relations & resource deployment
Serra et al. (1992)	M	N	-	CV	Competitive environment; 55 node problem
Gerrard & Church (1994)	M	N	-	CV	Health care; Referral systems; Zarazal, Columbia & Uganda
Mandell (1996)	M	N	-	CV	EMS; Probabilistic version of Moore & ReVelle (1982)
Espejo et al. (2003)	M	N	-	CV	Binary model; Lagrangean; 700 demand problems
Jayaraman et al. (2003)	M	N	-	CV	Extends Moore & Revelle (1982) with capacitated facilities
Johnson et al. (2005)	M	N	-	CV	Health care; Nursing home system; Max demand served s.t. capacity and budget
Shavandi et al. (2006)	M	N	CO	CV	Fuzzy queuing structure; Based on Marianov & Serra (2001); 15 node problem;
Sahin et al. (2007)	M	N	-	CV	Health care; Sequential optimization procedure pq-Median to set cover
Yasenovskiy & Hodgson (2007)	M	N	-	CV	Health care; Max customer welfare; Spatial interaction-based model, Ghana
Shavandi & Mahlooji (2008)	M	N	-	CV	Health care; Congested models; Fuzzy framework and queuing system

Lee & Lee (2010)	M	N	-	CV	Partial coverage; Heuristic solution; Up to 150 demand nodes

As seen in Table 3.4, all application papers featuring covering objectives are public sector papers. Because covering based approaches typically neglect the distribution variables, they are not commonly applied on private sector problems. This work explores a new modeling approach expanding the concepts of facility coverage with distribution considerations. In a logistics network, customer service is usually measured in days elapsed. This effectively creates varying bands of service levels, as described previously in section 3.2. Coverage based objectives can be used to model these service level bands reflecting proximity to each customer. This approach will be used in this work to model customer service in multi-criteria distribution systems.

All but one paper in table 3.4 is a multi-flow, nested version of the problem. This suggests that public sector papers, especially health services, dominate the use of coverage in hierarchical location modeling. A common cause of this is that the less restrictive flow discipline (multi-flow) is much more prevalent in public sector applications, where flow typically originates from the demands.

Table 3.5 is a categorization of the multi-objective hierarchical location papers found in the literature.

Table 3.5 Categorization of multi-criteria hierarchical location papers

Reference	FP	SA	SP	LO	Comments
Calvo & Marks (1973)	M	N	-	MO	Health care; First flow-based formulation
Dokmeci (1979)	M	N	-	MO	Health care; Bi-criteria variant of Dokmeci (1973)
Schilling et al. (1979)	M	N	-	MO	Fire protection; 120 node problem; Baltimore
Flynn & Ratick (1988)	M	N	CO	MO	Airline management; Nested and non-nested levels
Serra & ReVelle (1993)	M	N	CO	MO	Health care; first use of coherence; referral system; 25 node
Serra & ReVelle (1994)	M	N	CO	MO	Health care; Solution algorithms for Serra & ReVelle (1993)
Serra (1996)	M	N	CO	MO	Health care; Covering & median objectives
Alminyana et al. (1998)	M	N	CO	MO	Directed branching procedure for Serra & ReVelle (1993)
Marianov & Serra (2001)	S	N	CO	MO	Health Care; Covering and fixed charge; Congested
Galavao et al. (2006)	M	N	-	MO	Health care; Capacitated; Load balancing objective; Rio de
Mitropoulos et al. (2006)	M	N	-	MO	Health care; P-Median and covering objectives; Greece
Pahlavani & Mehrabad	M	N	-	MO	Min average travel time; Max coverage; Fixed cost; Fuzzy;
Gu & Wang (2012)	M	N	-	MO	Static and mobile vehicle location; Geo-spatial; Centering
Baray & Cliquet (2013)	M	N	-	MO	Health care; Median and max covering; Maternity hospitals in

Smith et al. (2013)	M	N	-	MO	Health care; Median and max covering; Equity criteria; India

As shown in Table 3.5, there has yet to be any multi-objective approaches utilized for the production-distribution location problem. The reason for this is the growing popularity of casting this problem as a supply chain design problem. Additionally, multi-criteria applications are greatly outnumbered by single criterion papers. The vast majority of the multi-criteria models in the field of hierarchical location have been health care applications. These public sector models all consider equity objectives, usually in some form of covering, with direct consideration of budgetary limitations. All applications have been nested, and all but one, Marianov & Serra (2001), have been multi-flow. As discussed earlier, this is because multi-flow systems are quite prevalent in health care applications.

3.4 Research Opportunities

Firstly, in private sector applications, a customer cannot be deemed "covered" if they are not being serviced by a facility within the coverage radius. The mere existence of a facility within the coverage radius (as in public sector applications) is insufficient for coverage in these problems. The allocation of demand or customer assignment determines coverage level. More exploration is needed on appropriate modeling approaches for covering in private sector applications.

Starting with the supply chain literature, there appears to be several areas of potential future research. These potential research areas are described below.

- Profit oriented objectives

Profit maximization is usually the main goal of most businesses. Therefore, the incorporation of revenue oriented objectives within supply chain design models should occur more frequently than presently found in the literature.

- Robustness and Risk

Robustness was an objective that was sought by very few articles in this summary, while financial risk was considered by only three. Maximizing the robustness of a solution, or minimizing its riskiness, is a factor that an increasing number of real-world businesses wish to consider, especially during these periods of economic turmoil.

- Metaheuristics

The genetic algorithm is easily the most frequently used metaheuristic in multi-objective supply chain network design models. More work comparing and evaluating the performance of other metaheuristics for this problem is needed.

Several promising areas of potential future research have resulted from touring the literature on the hierarchical production-distribution location problem. These areas are listed and described below.

- Multi-criteria approaches

All multi-criteria based approaches in the hierarchical location literature have been public sector problems. This can be explained by the natural way in which public sector problems, like health care and education systems, can be easily and accurately modeled as a hierarchical location problem. Although the production-distribution location problem is a private sector application, the amount of public sector application areas that naturally fit hierarchical location modeling techniques seems to greatly outnumber the potential private sector applications. Despite this, more work is clearly needed on multi-criteria private sector applications to the production-distribution location problem.

- Nestedness and/or coherency

Through the incorporation of multiple products in production-distribution location, further exploration of the properties of nestedness and coherency is a promising area of future research.

- Metaheuristics

There is very little in the literature regarding metaheuristic algorithms and the hierarchical location problem, both private and public sector.

3.5 Closest Assignment and Multi-Level Facility Location

A key property motivating this research is that closest assignment is not necessarily the most economical option. This property was first discovered in Rojeski & ReVelle (1970). This aspect is examined in a multi-level location problem setting in this work. In simpler location problems where there is no intermediary stocking location in the system (i.e. two echelon problems), the closest assignment policy is usually the cheapest option from a transportation cost perspective. However, as seen in Figure 3.4 below, this isn't always true in multi-echelon distribution systems.

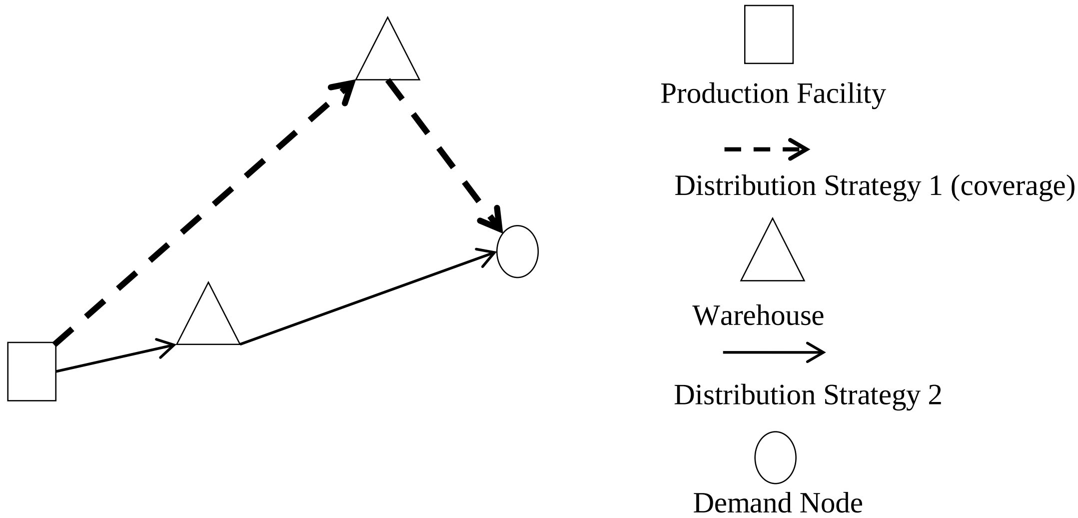


Figure 3.4 Cost versus coverage in multi-echelon distribution. Scenario A

As seen in Figure 3.4 above, Distribution Strategy 1 ensures coverage of the demand node. However, Distribution Strategy 2 is the cost minimizing approach, clearly invalidating the assumption that closest assignment is the cheapest. This is especially significant in a multi-criteria problem setting, where multiple solutions on the Pareto frontier will feature assignments of demand to warehouses outside of their respective coverage radii. A similar situation where this issue can arise in these problems can be seen in scenario B presented in Figure 3.5 below.

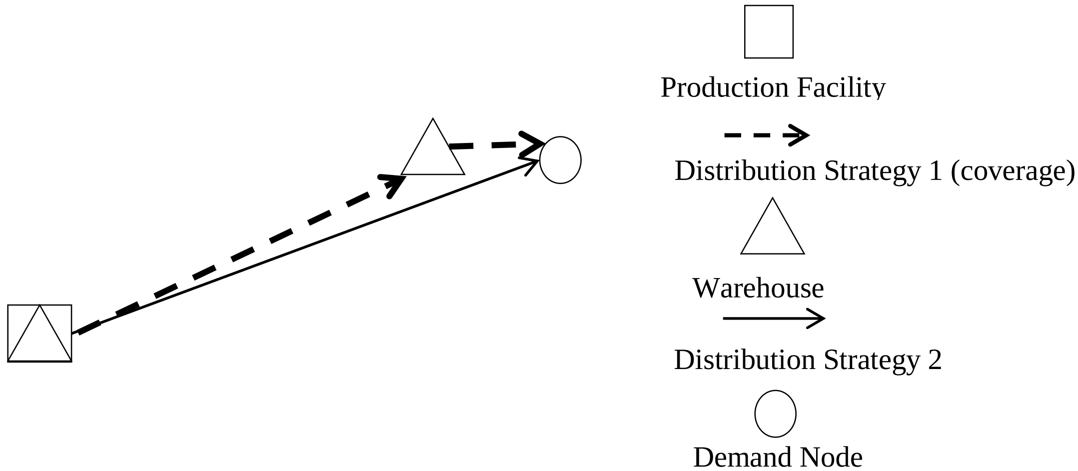


Figure 3.5 Cost versus coverage in multi-echelon distribution. Scenario B

In scenario B depicted in Figure 3.5, there is a nominal difference in total transportation cost as determined by total path length between Distribution Strategies 1 and 2. However, Distribution Strategy 2 is the cost minimizing one, and is likely to be selected more than once during the generation of the Pareto frontier.

Finally, Figure 3.6 illustrates why the closest assignment restrictions that can be found in the literature on public sector location problems, initially presented in Rojeski & ReVelle 1970, does not properly address this issue in a multi-echelon distribution setting considering the conflicting criteria of cost and service level.

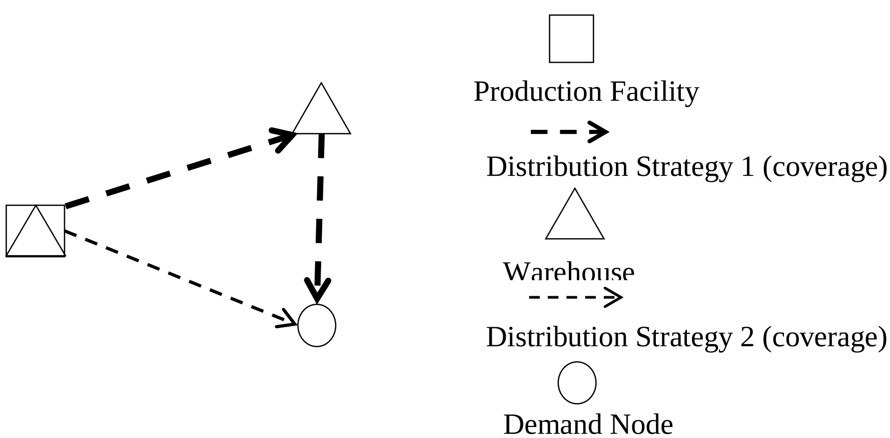


Figure 3.6 Cost versus coverage in multi-echelon distribution. Scenario C

In Figure 3.6, Distribution Strategy 1 is nearly twice as costly as Distribution Strategy 2, despite the fact that either strategy is adequate for coverage of the demand in scenario C. However, if closest assignment restrictions were enforced via any of the approaches summarized in Gerrard & Church (1996), then cost would be unnecessarily increased in a mutli-echelon distribution scenario.

Another issue potentially arises with the scenarios depicted in Figures 3.4 - 3.6. If the cost minimizing solution is chosen, then the customer may very well become dissatisfied at not receiving better service, under the assumption that they are aware of the distribution capabilities of their upstream supply chain partner. For example, suppose a customer is receiving a service level of one week from their upstream distributor. However, they are aware that there is a warehouse well within one day's drive of their location. Given that they are being serviced by a warehouse well outside of their service radius, this customer may become dissatisfied. These types of issues can be addressed at the onset of the network design phase with the modeling approach given in this work.

For several public sector applications, the approach outlined in Rojeski & ReVelle (1970) is the prudent one to take. A good example would be in the case of emergency vehicle location and the impact on response times to patients in need. In that scenario, the closest vehicle should always respond to the call or be dispatched for service. However, for private sector applications, especially in multi-echelon distribution, the problem hasn't been adequately addressed. This research contributes to the literature by addressing this gap through a series of multi-echelon location formulations for the distribution network design problem with mandatory service considerations.

3.6 Mandatory Service in Multi-Echelon Distribution System Design

The conflicting criteria to be considered in all of the models provided here will be the minimization of the demand weighted average distance (P-median) and the maximization of the amount of demand within a given coverage radius (maximal covering). In the computational exercises, experiments will be conducted with increasing amounts of locations to select. Additionally, all of the models given here can be simplified to a single objective problem subject a minimum performance level on the other objective. Lastly, these flow based models can all by formulated as path-based, with three indexes on the distribution variables. These models are omitted here.

The first model presented in this work is a single echelon warehouse location problem given a set of plants and demand generating nodes. This scenario reflects the all too common problem of determining the best number and location of warehouses in a distribution network. With the set of demands indexed by $i \in I$, warehouses by $j \in J$, plants by $k \in K$, and the set of all nodes given by $N = I \cup J \cup K$, this first scenario can be formulated as Model 3.1. Note that the superscripts on the distribution variables are there for clarity. They represent the origin of the shipment, not a third index.

Model 3.1

Inputs

$h_i =$ demand at node $i \in I$

$d_{ij} =$ distance between nodes $i \in N$ and $j \in N$

$P =$ number of facilities to locate

$$a_{ij} = \begin{cases} 1 \text{ if candidate facility } j \in J \text{ can cover demand } i \in I \\ 0 \text{ otherwise} \end{cases}$$

Decision Variables

$$X_j = \begin{cases} 1 \text{ if candidate location } j \in J \text{ is selected} \\ 0 \text{ otherwise} \end{cases}$$

Y_{jk}^p = amount of product shipped from plant $k \in K$ to warehouse $j \in J$

Y_{ij}^w = amount of product shipped from warehouse $j \in J$ to customer $i \in I$

$$Z_i = \begin{cases} 1 \text{ if demand } i \in I \text{ is covered} \\ 0 \text{ otherwise} \end{cases}$$

Minimize $\sum_{i \in I} \sum_{j \in J} d_{ij} Y_{ij}^w + \sum_{j \in J} \sum_{k \in K} d_{jk} Y_{jk}^p$ (3.26)

Minimize $\sum_{i \in I} \sum_{j \in J} (1 - a_{ij}) Y_{ij}^w$ (3.27)

Subject To:

$a_{ij} X_j \leq Z_i$	$\forall i \in I, j \in J$	(3.28)
$h_i Z_i \leq \sum_{j \in J} a_{ij} Y_{ij}^w$	$\forall i \in I$	(3.29)
$Y_{ij}^w \leq h_i X_j$	$\forall i \in I,\ j \in J$	(3.30)
$\sum_{j \in J} Y_{ij}^w = h_i$	$\forall i \in I$	(3.31)
$\sum_{i \in I} Y_{ij}^w = \sum_{k \in K} Y_{jk}^p$	$\forall j \in J$	(3.32)
$\sum_{j \in J} X_j = P$		(3.33)
$X_j \in \{0,1\}$	$\forall j \in J$	(3.34)
$Y_{ij}^w \geq 0$	$\forall i \in I,\ j \in J$	(3.35)
$Y_{jk}^p \geq 0$	$\forall j \in J, k \in K$	(3.36)
$Z_i \in \{0,1\}$	$\forall i \in I$	(3.37)

Objective (3.26) minimizes total demand weighted distance. Objective (3.27) minimizes total uncovered demands. Constraint (3.28) states that if a facility is selected within the coverage radius, then customer $i \in I$ is covered. If customer $i \in I$ is covered, then the entirety of its demand must be satisfied by a facility within the coverage radius (3.29). Constraint (3.30) links the distribution variables to the facility selection variables and prevents any shipments from warehouses that aren't selected. Additionally, a tight upper bound of h_i is enforced for each ${Y_{ij}}^W$ with this constraint. Demand must be satisfied at all customers (3.31). The total amount of product entering a warehouse is equal to the total amount of product leaving a warehouse (3.32). A total of P facilities are to be located (3.33). Constraints (3.34) and (3.37) are binary restrictions on the facility location and coverage variables, while (3.35) and (3.36) are non-negativity restrictions on the distribution variables.

Model 3.1 is a bi-objective warehouse location problem in a distribution system. This problem could also be referred to as a strategic network design problem or a supply chain design problem. However, the mandatory service considerations discussed throughout this chapter have been incorporated. With these restrictions in place, if a customer could be serviced within the coverage radius (by virtue of being close enough to an open facility that can service them), then they must be serviced by a facility within the coverage radius (but not necessarily by the closest selected location). This property will hold true at every iteration of a generating technique in multi-criteria optimization.

The next model provided in this work is a multi-flow version of Model 3.1. Given the following additional notation, this problem can be formulated as Model 3.2.

Model 3.2

Inputs

$$a_{ij} = \begin{cases} 1 \text{ if any plant } k \in K \text{ or candidate warehouse } j \in J \text{ can cover demand } i \in I \\ 0 \text{ otherwise} \end{cases}$$

Decision Variables

Y_{ij} = amount of product shipped from facility $j \in J \cup K$ to customer $i \in I$

Minimize $\sum_{i \in I} \sum_{j \in J \cup K} d_{ij} Y_{ij} + \sum_{j \in J} \sum_{k \in K} d_{jk} Y_{jk}^{p}$ (3.38)

Minimize $\sum_{i \in I} \sum_{j \in J \cup K} (1 - a_{ij)} Y_{ij}$ (3.39)

Subject To:

$$a_{ij} X_j \leq Z_i \qquad \forall i \in I, j \in J \qquad (3.40)$$

$$h_i Z_i \leq \sum_{j \in J \cup K} a_{ij} Y_{ij} \qquad \forall i \in I \qquad (3.41)$$

$$Y_{ij} \leq h_i X_j \qquad \forall i \in I,\ j \in J \qquad (3.42)$$

$$Y_{ik} \leq M X_k \qquad \forall i \in I \cup J,\ k \in K \qquad (3.43)$$

$$\sum_{j \in J \cup K} Y_{ij} = h_i \qquad \forall i \in I \qquad (3.44)$$

$$\sum_{i \in I} Y_{ij} = \sum_{k \in K} Y_{jk}^{p} \qquad \forall j \in J \qquad (3.45)$$

$$\sum_{j \in J} X_j = P \qquad (3.46)$$

$$X_j \in \{0,1\} \qquad \forall j \in J \qquad (3.47)$$

$$Y_{ij} \geq 0 \qquad \forall i \in I,\ j \in J \cup K \qquad (3.48)$$

$$Y_{jk}^{p} \geq 0 \qquad \forall j \in J, k \in K \qquad (3.49)$$

$$Z_i \in \{0,1\} \qquad \forall i \in I \qquad (3.50)$$

Objective (3.38) minimizes total demand weighted distance. Objective (3.39) minimizes total uncovered demands. Constraint (3.40) states that if a facility is selected within the coverage radius, then customer $i \in I$ is covered. If customer $i \in I$ is covered, then the entirety of its demand must be satisfied by a facility within the coverage radius (3.41). Constraint (3.42) links the distribution variables to the warehouse selection variables and prevents any shipments from warehouses that aren't selected. Additionally, a tight upper bound of h_i is enforced for each Y_{ij} with this constraint. Constraint (3.43) links the distribution variables to the plant selection variables and prevents any shipments from plants that aren't selected. Demand must be satisfied at all customers (3.44). The total amount of product entering a warehouse is equal to the total amount of product leaving a warehouse (3.45). A total of P facilities are to be located (3.46). Constraints (3.47) and (3.50) are binary restrictions on the facility location and coverage variables, while (3.48) and (3.49) are non-negativity restrictions on the distribution variables.

The final scenario modeled here is a bi-objective hierarchical production-distribution location problem, where there are two location echelons (P warehouses and Q plants). Additionally, plants as well as warehouses can deliver product to demand (multi-flow). A simplification of the following model for the single flow scenario will not be given. With the following additional notation, this problem can be formulated as Model 3.3.

Model 3.3

Inputs

$Q =$ number of plants to locate

$M = \sum_{i \in I} h_i$

$$a_{ij} = \begin{cases} 1 \text{ if candidate facility } j \in J \cup K \text{ can cover demand } i \in I \\ 0 \text{ otherwise} \end{cases}$$

Decision Variables

$$X_j = \begin{cases} 1 \text{ if candidate location } j \in J \cup K \text{ is selected} \\ 0 \text{ otherwise} \end{cases}$$

$Y_{jk}^p =$ amount of product shipped from plant $k \in K$ to warehouse $j \in J$

$Y_{ij} =$ amount of product shipped from facility $j \in J \cup K$ to customer $i \in I$

$$Z_i = \begin{cases} 1 \text{ if demand } i \in I \text{ is covered} \\ 0 \text{ otherwise} \end{cases}$$

Minimize $\sum_{i \in I} \sum_{j \in J \cup K} d_{ij} Y_{ij} + \sum_{j \in J} \sum_{k \in K} d_{jk} Y_{jk}^{p}$ (3.51)

Minimize $\sum_{i \in I} \sum_{j \in J \cup K} (1 - a_{ij}) Y_{ij}$ (3.52)

Subject To:

$a_{ij} X_j \leq Z_i \quad \forall i \in I, j \in J \cup K$ (3.53)

$h_i Z_i \leq \sum_{j \in J \cup K} a_{ij} Y_{ij} \quad \forall i \in I$ (3.54)

$Y_{ij} \leq h_i X_j \quad \forall i \in I,\ j \in J$ (3.55)

$Y_{ik} \leq M X_k \quad \forall i \in I \cup J,\ k \in K$ (3.56)

$\sum_{j \in J \cup K} Y_{ij} = h_i \quad \forall i \in I$ (3.57)

$\sum_{i \in I} Y_{ij} = \sum_{k \in K} Y_{jk}^{p} \quad \forall j \in J$ (3.58)

$\sum_{j \in J} X_j = P$ (3.59)

$\sum_{k \in K} X_k = Q$ (3.60)

$X_j \in \{0,1\} \quad \forall j \in J \cup K$ (3.61)

$Y_{ij} \geq 0 \quad \forall i \in I,\ j \in J \cup K$ (3.62)

$Y_{jk}^{p} \geq 0 \quad \forall j \in J, k \in K$ (3.63)

$Z_i \in \{0,1\} \quad \forall i \in I$ (3.64)

Objective (3.51) minimizes total demand weighted distance. Objective (3.52) minimizes total uncovered demands. Constraint (3.53) states that if a facility is selected within the coverage radius, then customer $i \in I$ is covered. If customer $i \in I$ is covered, then the entirety of its demand must be satisfied by a facility within the coverage radius (3.54). Constraint (3.55) links the distribution variables to the warehouse selection variables and prevents any shipments from warehouses that aren't selected. Additionally, a tight upper bound of h_i is enforced for each Y_{ij} with this constraint. Constraint (3.56) links the distribution variables to the plant selection variables and prevents any shipments from plants that aren't selected. Demand must be satisfied at all customers (3.57). The total amount of product entering a warehouse is equal to the total amount of product leaving a warehouse (3.58). A total of P warehouses and Q plants are to be located (3.59) and (3.60) respectively. Constraints (3.61) and (3.64) are binary restrictions on the facility location and coverage variables, while (3.62) and (3.63) are non-negativity restrictions on the distribution variables.

4. Multi Criteria Decision Analysis

4.1 Introduction

Multi-criteria decision analysis (MCDA) or multi-criteria decision aid is the process of applying a practical methodology to assist a decision maker in selecting a compromise solution to a difficult multiple criteria problem. Many techniques developed in MCDA are also capable of addressing the Condorcet paradox, a phenomenon more commonly known as intransitivity, which often arises in multi-criteria decision making, (Collet & Siarry 2004). Classic multi-objective optimization methodologies are incapable of addressing this issue, which necessitated the use of alternative measures to do so. Intransitivity can best be described as follows:

> "Given three actions A, B and C, we can have $A \geq B, B \geq C, and\ C \geq A$ (here, the symbol $\geq$ corresponds to the preference relation)," (Collet & Siarry 2004).

MCDA techniques can be used as an alternative to multi-objective optimization techniques, or a complementary aid tool to further assist the decision maker. The approach taken in this work is complimentary with multi-objective optimization techniques by applying an MCDA methodology for the final selection of a Pareto efficient solution.

MCDA methodologies require as inputs a discrete set of possible courses of action for a given problem. Either a complete or partial ordering of these solutions is done based upon one or multiple criteria. Partial orderings of this set of actions (solutions) are accomplished only if some solution alternatives are incomparable. Definition 4 formalizes the preference relations to be used, (Collete & Siarry 2004).

> Definition 4. *Preference Relations*
> Given courses of action a and b:
> $a\ P\ b$ indicates that action a is preferred to b or $a \geq b$
> $a\ I\ b$ indicates that action a is indifferent to b or $a = b$
> $a\ R\ b$ indicates that actions a and b are not comparable or neither $a\ P\ b$ nor $a\ I\ b$

For this work, the criterion of network flexibility is going to be formally given and applied in a post-optimization procedure. The result of this analysis will be a complete ordering of the Pareto frontier with regards to this new criterion through the establishment of the preference relations between every pairing of solutions in the optimal set. A decision maker may then select a solution to implement based upon performance across all objectives in addition to overall network flexibility.

In the event that the size of the Pareto efficient set is too great to efficiently construct a complete ordering, a reduction in the amount of solutions to consider during the application of this MCDA tool should be conducted. Perhaps a classical MCDA approach like the ELECTRE or PROMETHEE methods can be applied prior to the use of this decision aid to assist in this process. A thorough discussion of these MCDA techniques can be found in Collette & Siarry (2004).

4.2 Network Flexibility for Pareto Efficient Solution Selection

The flexibility of a distribution network, in the context of multi-criteria location analysis, is determined by its relative distance from its neighboring solutions on the Pareto frontier. The term “distance” when used here indicates the degree to which varying sets of location decisions differ. This distance reflects the relative costs inherent in changing an established network to an alternate Pareto efficient solution. This can also be interpreted as the cost of adjusting the weight associated with the objectives in a MOCO optimization applying a scalar solution approach. This work focuses solely on the interactions of economic and customer responsiveness objectives, therefore, this is a post-optimality procedure of a bi-criteria optimization problem. However, the MCDA technique given here can be just as easily applied in circumstances where there are more than two objective criterions.

The distance measurement used here is inspired by the Hamming distance, (Hamming 1950). The origins of the Hamming distance stems from information theory and the comparison of bit strings of equal length to determine the number of positions which corresponding elements differ between two strings. The procedure used here is ideologically the same by comparing the set of location decisions between two solutions and calculating the total amount of differing decisions. Hamming distance as applied in this work is defined below in definition 5, (Hamming 1950):

Definition 5. *Hamming Distance*
Given two *binary* variable vectors $\mathbf{x}^1, \mathbf{x}^2 \in S$, their *Hamming distance* is given by the the following: $d_H(\mathbf{x}^1, \mathbf{x}^2) = \sum_i \left| \mathbf{x}_i^1 - \mathbf{x}_i^2 \right|$

Using the Hamming distance calculation given in definition 5, and the previously defined set of candidate facility locations F, the network percentage difference ($NPD_{i,j}$) metric between binary location variable vectors $\mathbf{x}^i$ and $\mathbf{x}^j$, each of length N is presented below:

$$NPD_{i,j} = \frac{\sum_{n\in N}\left|\mathbf{x}_n^{i\in L} - \mathbf{x}_n^{j\in L}\right|}{|F|} \tag{4.2.1}$$

The metric above is the Hamming distance between any two feasible solutions in the set of Pareto optimal alternatives $(\mathbf{x}^{\mathrm{i}}, \mathbf{x}^j) \in L$ divided by the cardinality of the set of location variables F. This metric gives the percentage difference in locations between any two logistics networks. Decision makers interested in preserving network flexibility will desire values for these metrics to be as close to zero as possible when compared to all other solutions on the Pareto frontier. An assumption here is that closing and opening costs are symmetric across all candidate locations. If this isn't the case, a cost based approach should be used.

If there are $|L|$ Pareto optimal solutions, then there are a total of $\frac{|L| \times (|L|-1)}{2}$ NPD values to compute, with $l - 1$ values for each member of the Pareto optimal set. In order to facilitate meaningful comparisons amongst the members of the Pareto optimal set, an aggregate NPD score capturing the network flexibility of an individual solution must be utilized. For each member of the

Pareto optimal set, a mean of its $|L| - 1$ NPD scores is used for these purposes. The mean network percentage difference (μNPD) metric is given below.

$$\mu NPD_i = \frac{1}{|L|-1} \sum_{j \neq i \in L} \frac{\sum_{n \in N} \left| \mathbf{x}_n^i - \mathbf{x}_n^j \right|}{|F|} = \frac{\sum_{j \neq i \in L} NPD_{i,j}}{|L|-1} \quad (4.2.2)$$

For the purposes of creating a complete ordering of the Pareto optimal set with this metric, the following holds true for solutions $i, j \in L$:

$$i\ P\ j \vdash \mu NPD_i < \mu NPD_j$$

and

$$i\ I\ j \vdash \mu NPD_i = \mu NPD_j$$

In the above, $i\ P\ j$ implies that $\mu NPD_i < \mu NPD_j$, while $i\ I\ j$ implies that $\mu NPD_i = \mu NPD_j$.

The μNPD_i metric given by (4.2.2) may be used in the circumstances where the decision maker has absolutely no preference to any individual objective being pursued, and the level of objective performance degradation present when moving from solution $i \in L$ to $j \in L$ is of no concern.

However, if a maximum degradation level must be enforced as per the wishes of the DM, then a modification must be made to expression (4.2.2). In order to accommodate this consideration, we define a set for each Pareto efficient solution $B_{i \in L}$ consisting of the subset of all other solutions in L that respect a maximum degradation level for all k objectives in regards to solution i. In other words a constraint is applied across the set of solutions, which filters out those that exceed a maximum allowable change in any objectives. Under the assumption that all $k \in K$ objectives are to be minimized, then this set can be calculated as follows: $B_i = \left\{ j \neq i \in L : \left(f_j^k(\mathbf{x}) - f_i^k(\mathbf{x}) \right) / f_i^k(\mathbf{x}) \leq p_k \ \ \forall\, k \in K \right\}$, where the parameter p_k is a maximum degradation percentage for objective k. Using this additional notation, the μNPD_i metric is redefined below in (4.2.3).

$$\mu NPD_i^{\,p_k} = \frac{1}{|B_i|} \sum_{j \in B_i} \frac{\sum_{n \in N} \left| \mathbf{x}_n^i - \mathbf{x}_n^j \right|}{|F|} = \frac{\sum_{j \in B_i} NPD_{i,j}}{|B_i|} \qquad (4.2.3)$$

This issue brings forth another important consideration that is relevant for this discussion. If there is a maximum acceptable degradation level of objective performance for each solution $i \in L$, then varying sizes of the set B_i could exist amongst the Pareto optimal solutions. Therefore, the cardinality of these sets should also be used as a metric capturing network flexibility. In fact, this concept must be used in conjunction with the $\mu NPD_i^{\,p_k}$ metric. After all, if a solution $i \in L$ has a $\mu NPD_i^{\,p_k} = .1$, then that solution would appear to have a rather strong network flexibility rating, in that its underlying structure is on average only 10% different than its fellow Pareto optimal solutions. However, if $|B_i| = 1$, then that 10% $\mu NPD_i^{\,p_k}$ rating suddenly seems much less impressive. After all, how flexible can a solution be if there is only one other acceptable recourse network structure? For simplicity of further discussion, this second aspect of network flexibility is formally defined in (4.2.4) as the acceptable recourse networks (ARN_i) metric.

$$ARN_i = |B_i|: B_i = \left\{j \neq i \in L: \left(f_j^k(\mathbf{x}) - f_i^k(\mathbf{x})\right)/f_i^k(\mathbf{x}) \leq p_k \ \forall\, k \in K\right\} \quad (4.2.4)$$

The ARN_i metric given above provides the number of acceptable recourse networks available to solution $i \in L$. A decision maker interested in establishing a high degree of network flexibility would desire the value of this metric to be as close to $|L| - 1$ as possible.

For the purposes of creating a complete ordering of the Pareto optimal set in regards to this metric, the following holds true for solutions $i, j \in L$:

$$i\ P\ j \vdash ARN_i > ARN_j$$

and

$$i\ I\ j \vdash ARN_i = ARN_j$$

In the above, $i\ P\ j$ implies that $ARN_i > ARN_j$, while $i\ I\ j$ implies that $ARN_i = ARN_j$. A solution with more recourse solutions has a higher degree of flexibility, hence is preferred.

The μNPD_i metric considers only the differences in location decisions amongst logistics networks, but does not take into account distances. When deciding how similar two networks are spatially, distance might also be taken into account. For example, a network consisting of warehouses in New York, Atlanta, and Oakland is little different from a distance perspective than a network with warehouses in Newark, Birmingham, and San Francisco. However, the μNPD_i

metric will suggest that the two distribution networks are vastly different, when in reality, they really aren't.

One way to address this is to modify the $NPD_{i,j}$ calculations such that a positive hamming distance can only occur if a non-selected location in solution $i \in L$ is further than a set distance from the closest location in solution $j \in L$. Alternatively, comparing customer assignments amongst differing network design solutions could be a promising approach as well. Both approaches require comparing elements of location (or assignment) vectors across differing indices, going beyond the simple methodology presented in this work. However, these modifications are promising avenues of future research in this stream.

4.3 Network Flexibility: A Cost Perspective

In the preceding section, a set of MCDA metrics to rate the relative flexibility of a distribution network by evaluating and comparing the similarity of the prescribed network configurations among a set of Pareto optimal solutions was formally defined. Often, a decision maker or a group of decision makers prefer comparisons and values displaying the costs of business decisions. In this case, network reconfiguration costs can be easily constructed to accompany the previously defined metrics. Building upon the existing notation, the following variables are defined to simplify this discussion.

$$U_f^{l,k} = \left(\mathbf{x}_f^{l\in L} - \mathbf{x}_f^{k\in L}\right) \qquad \forall\, l \neq k \in L, f \in F: \mathbf{x}_f^{l\in L} - \mathbf{x}_f^{k\in L} \geq 0$$

$$V_f^{l,k} = -1\left(\mathbf{x}_f^{l\in L} - \mathbf{x}_f^{k\in L}\right) \qquad \forall\, l \neq k \in L, f \in F: \mathbf{x}_f^{l\in L} - \mathbf{x}_f^{k\in L} \leq 0$$

Recall that $F = I \cup J$. Using these decision variables, and an estimated closing cost for all candidate plant and warehouse locations $(r_{i\in I}, r_{j\in J})$, the following network reconfiguration cost estimation can be used:

$$N_{l,k\in L} = \sum_{i,j\in f}\left(f_{i\in I}U_{i\in I}^{l,k} + r_{i\in I}V_{i\in I}^{l,k} + f_{j\in J}U_{j\in J}^{l,k} + r_{j\in J}V_{j\in J}^{l,k}\right) \quad (4.3.1)$$

where $N_{l,k\in L}$ is the total network reconfiguration cost between Pareto efficient solutions $l \neq k \in L$. An assumption here is that the facility opening and closing costs are the only significant cost components. Rearrangement of distribution assets isn't explicitly considered in (4.3.1). However, these costs can be included in the $(r_{i\in I}, r_{j\in J})$ parameter estimations, if they are significant enough to merit direct consideration within the MCDA approach.

If a decision maker will not allow network reconfiguration costs to exceed a specific value, or if an upper bound is desired on $N_{l,k\in L}$ for the purposes of further reducing the set of candidate solutions for the final decision, these calculations can be utilized in the development of the $ARN_{i\in L}$ metric. Given below in (4.3.2) and (4.3.3) is an updated definition of the B_i sets and the ARN_i network flexibility metric to accommodate these cost considerations.

$$B_i = \left\{j \neq i \in L: \left(f_j^k(\mathbf{x}) - f_i^k(\mathbf{x})\right)/f_i^k(\mathbf{x}) \leq p_k \;\; \forall\, k \in K \text{ and } N_{l,k\in L} \leq b_r\right\} \quad (4.3.2)$$

$$ARN_{i\in L} = |B_i| \quad (4.3.3)$$

4.4 Network Flexibility as an Additional Criterion in MCDA

The process outlined in sections 4.2 and 4.3 apply the MCDA technique derived here as a final solution selection procedure given an efficient subset of alternatives as found by competing criteria in a multi-objective optimization problem (cost and service in this case). However, a decision maker may be interested in incorporating the magnitudes of deviation of criterion performance of each solution in an efficient set from a best possible alternative. In this case, we must normalize criterion performance by using the ideal and nadir points. What follows is a brief summary of these concepts in multi-criteria optimization.

Figure 4.1 Ideal and nadir points in a bi-criteria discrete optimization problem

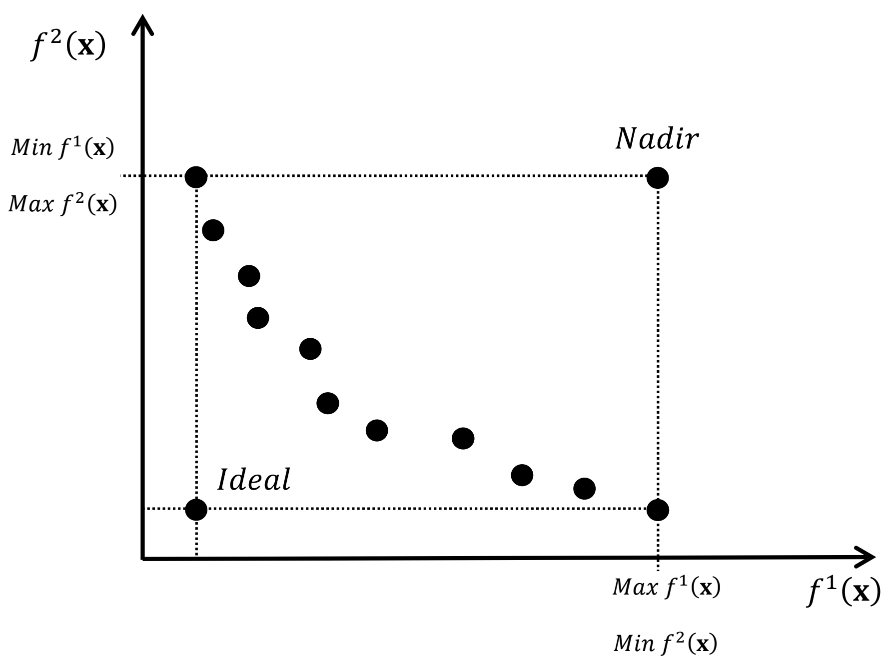


Figure 4.1 is a simple illustration of the ideal and nadir points as determined by a set of efficient solutions in a bi-criteria discrete optimization problem. In this case, minimization is sought for both criterions. As this illustration suggests, the ideal point is

an unattainable point in *Z* space (else we wouldn't need multi-criteria techniques to solve the problem) where all criterion being considered is optimized. The nadir point is that point in *Z* space where the performances of all objectives are at their worst. In many cases, the nadir point is also infeasible.

Let Z^n = the nadir point and Z^{**} = the ideal point. Given that minimization is sought in all cases, an objective vector Z^i can be normalized as follows.

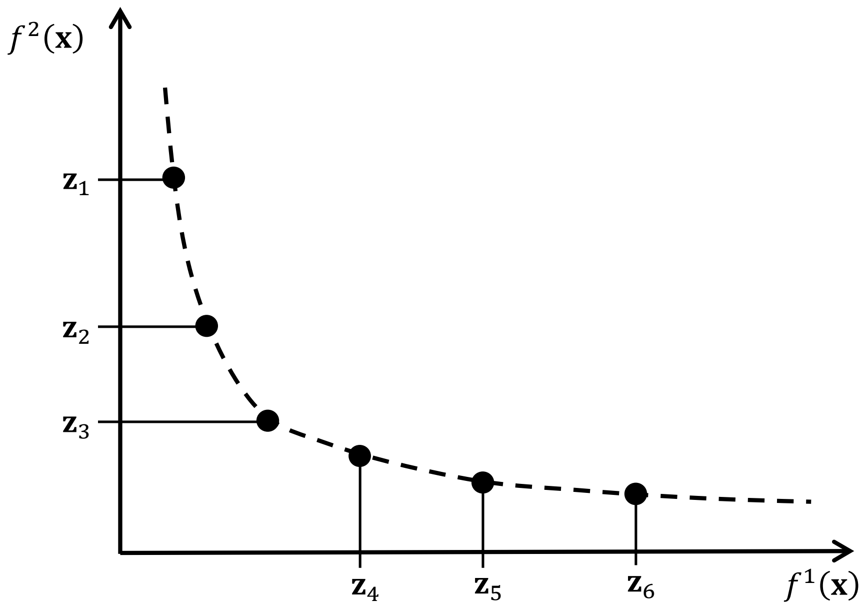
$$\boldsymbol{\widehat{Z^i}} = \frac{\boldsymbol{Z^i} - \boldsymbol{Z}^{**}}{\boldsymbol{Z^n} - \boldsymbol{Z}^{**}}$$

By using these normalized values, MCDA techniques could then be applied to all criterion considered in a decision aid, with or without the inclusion of weights on the individual, normalized objective values.

4.5 Example MCDA Application

In this example, a Pareto optimal set for a multi-criteria location problem with two conflicting minimization objectives is examined. Because this is a discrete optimization problem, the frontier depicted in figure 4.1 is a dotted line, signifying the lack of an infinite set of Pareto efficient solutions.

Figure 4.1 Example set of efficient solutions for a bi-objective location problem



As seen in Figure 4.1, there are six Pareto optimal solutions in this example. Therefore, $|L| = 6$. Table 4.1 is the objective performance of all the solutions on the frontier.

Table 4.1: Objective Performance

Solution	$f^1(x)$	$f^2(x)$
z_1	25	100
z_2	30	60
z_3	40	30
z_4	50	26
z_5	60	23
z_6	75	20

The simple data provided in Table 4.1 is normalized according to the ideal and nadir points below in Table 4.2. Table 4.3 is the set of location decisions for each efficient solution.

Table 4.2: Normalized Objective Values

Solution	$f^1(x)$	$f^2(x)$
z_1	0	1
z_2	.1	.5
z_3	.3	.125
z_4	.5	.075
z_5	.7	.0375
z_6	1	0

Table 4.3: Facility Location Decisions

Facility	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$	$\mathbf{z}_6$
1	X		X	X	X	X
2	X	X	X		X	
3				X		X
4					X	
5	X	X	X	X		X
6						X
7	X	X	X			
8	X	X	X	X	X	
9				X		
10	X	X	X	X	X	X
11						
12						
13	X				X	
14		X	X	X		X
15					X	X
16					X	X
17	X	X	X	X	X	
18				X		X
19		X				X
20	X		X		X	X

In the following analysis, all 6 Pareto optimal solutions will be compared using the μNPD and ARN metrics. For this analysis, the symbols “+”, “=”, and “-“ in cell (i, j) indicates $i\ P\ j$, $i\ I\ j$, and $j\ P\ i$ respectively.

Given below in Table 4.4 are the NPD calculations for all $(i, j) \in L$ pairings in the Pareto optimal set $l \in L$.

Table 4.4:NPD Matrix

Solution	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$
$\mathbf{z}_1$					
$\mathbf{z}_2$	.25				
$\mathbf{z}_3$	.10	.15			
$\mathbf{z}_4$	.40	.35	.30		
$\mathbf{z}_5$	.25	.50	.35	.55	
$\mathbf{z}_6$	.60	.55	.45	.40	.55

By using the example output given in Table 4.4, Table 4.5 can be constructed with the equation given in (4.2.2). Recall that this metric is utilized when there isn't a maximum allowable objective performance degradation restriction (which subsequently renders the ARN metric useless).

Table 4.5: Example μNPD Metrics

Solution	μNPD_i
$\mathbf{z}_1$	.32
$\mathbf{z}_2$	.36
$\mathbf{z}_3$	.27
$\mathbf{z}_4$	.40
$\mathbf{z}_5$	.44
$\mathbf{z}_6$	.51

Normalizing Table 4.5, we have Table 4.6 below.

Table 4.6: Normalized μNPD Metrics

Solution	μNPD_i
$\mathbf{z}_1$	.33
$\mathbf{z}_2$	.375
$\mathbf{z}_3$	0
$\mathbf{z}_4$	.54
$\mathbf{z}_5$	.71
$\mathbf{z}_6$	1

Given the information provided in Tables 4.2 and 4.6, a complete ordering of all solutions in L can be created. This analysis can be seen below in Table 4.7.

Table 4.7: Example $MCDA$ Analysis (μNPD_i)

Rank	μNPD_i	$f^1(\mathbf{x})$	$f^2(\mathbf{x})$
1	$\mathbf{z}_3$	$\mathbf{z}_1$	$\mathbf{z}_6$
2	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_5$
3	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$
4	$\mathbf{z}_4$	$\mathbf{z}_4$	$\mathbf{z}_3$
5	$\mathbf{z}_5$	$\mathbf{z}_5$	$\mathbf{z}_2$
6	$\mathbf{z}_6$	$\mathbf{z}_6$	$\mathbf{z}_1$

As seen in Table 4.7, solution $\mathbf{z}_3$ appears to be the most flexible given this metric. From this table of information, a preference diagram can be created. This is given below in Table 4.8.

Table 4.8: Preference Diagram (μNPD_i,$f^1(\mathbf{x})$,$f^2(\mathbf{x})$)

Solution	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$
$\mathbf{z}_1$					
$\mathbf{z}_2$	(-,-,+)				
$\mathbf{z}_3$	(+,-,+)	(+,-,+)			
$\mathbf{z}_4$	(-,-,+)	(-,-,+)	(-,-,+)		
$\mathbf{z}_5$	(-,-,+)	(-,-,+)	(-,-,+)	(-,-,+)	
$\mathbf{z}_6$	(-,-,+)	(-,-,+)	(-,-,+)	(-,-,+)	(-,-,+)

Using Table 4.8, we can compare solutions $\mathbf{z}_2$ and $\mathbf{z}_5$ in the following fashion:

- Solution $\mathbf{z}_2$ is superior for metric one (μNPD_i).
- Solution $\mathbf{z}_2$ is superior for metric two ($f^1(\mathbf{x})$).
- Solution $\mathbf{z}_5$ is superior for metric three ($f^2(\mathbf{x})$).

Based on these findings, the conclusion that $\mathbf{z}_2\ P\ \mathbf{z}_5$, or solution $\mathbf{z}_2$ is preferred to solution $\mathbf{z}_5$ can be made. Using this procedure, an additional preference diagram can be created under the assumption that all performance metrics are considered to be equally important to the DM. This output is shown below in Table 4.9.

Table 4.9: Preference Diagram (l_i, l_j)

Solution	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$
$\mathbf{z}_1$					
$\mathbf{z}_2$	-				
$\mathbf{z}_3$	+	+			
$\mathbf{z}_4$	-	-	-		
$\mathbf{z}_5$	-	-	-	-	
$\mathbf{z}_6$	-	-	-	-	-

Considering the output given in Table 4.9, the chosen solution the DM should implement would be $\mathbf{z}_3$. Note that this conclusion can only hold if the importance of all three performance metrics is considered to be exactly equal. If weights were applied in this procedure, the outcome could be quite different.

If the DM wishes to enforce a maximum objective performance degradation restriction, then Metrics (4.2.3) and (4.2.4) should be used. Table 4.10 provides the ARN metric for three different p_k values over all 6 efficient solutions. This can be constructed directly from Table 4.4. In the following, it is assumed that $p_k = p_l\ \forall\ l \neq k \in K$. In other words, the allowable percentage degradation is the same across all k objectives. This assumption is made for simplicity of exposition here, but clearly isn't mandatory. In fact, the elements of vector p_k can be any number in the interval [0,1].

Table 4.10:Example ARN_i Metrics

Solution	$p_k = .25$	$p_k = .35$	$p_k = .50$
$\mathbf{z}_1$	3	3	4
$\mathbf{z}_2$	2	3	4
$\mathbf{z}_3$	2	4	5
$\mathbf{z}_4$	0	2	4
$\mathbf{z}_5$	1	2	5
$\mathbf{z}_6$	0	0	2

Table 4.10 shows that at the most restricted level ($p_k = .25$), the solution exhibiting the most flexibility within a range of acceptability is $\mathbf{z}_1$. However, the best solution changes to $\mathbf{z}_3$ (for this metric) when p_k is increased by 10% and 25%.

An updated version of Table 4.5 is given below, reflecting the $\mu NPD_i^{p_k}$ values.

Table 4.11: Example $\mu NPD_i^{p_k}$ Metrics

Solution	μNPD_i^{25}	μNPD_i^{35}	μNPD_i^{50}
$\mathbf{z}_1$	.20	.20	.25
$\mathbf{z}_2$	.20	.25	.313
$\mathbf{z}_3$	.125	.225	.27
$\mathbf{z}_4$	N/A	.325	.363
$\mathbf{z}_5$	.25	.3	.367
$\mathbf{z}_6$	N/A	N/A	.425

A normalization procedure will not be done in this case because the weights will be considered to be equal again in order to simplify this simple demonstration. If a decision maker examined the output for μNPD_i^{25} alone, that person could choose solution $\mathbf{z}_3$ under the impression that they selected the most flexible network configuration. However, the fact that its respective ARN_3 metric is equal to 2 clearly contradicts this conclusion. Therefore, when conducting post optimality analyses of these metrics, a primary sorting of the Pareto optimal solutions should usually be made in regards to the ARN_i metric first, followed by a secondary sub sorting using the $\mu NPD_i^{p_k}$ metric. Tables 4.12 and 4.13 below give the analyses for a $p_k = .25$.

Table 4.12: Preference Diagram $\{(ARN_i \rightarrow \mu NPD_i^{.15}), f^1(\mathbf{x}), f^2(\mathbf{x})\}$

Solution	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$
$\mathbf{z}_1$					
$\mathbf{z}_2$	(-,-,+)				
$\mathbf{z}_3$	(-,-,+)	(+,-,+)			
$\mathbf{z}_4$	(-,-,+)	(-,-,+)	(-,-,+)		
$\mathbf{z}_5$	(-,-,+)	(-,-,+)	(-,-,+)	(+,-,+)	
$\mathbf{z}_6$	(-,-,+)	(-,-,+)	(-,-,+)	(=,-,+)	(-,-,+)

Table 4.13: Preference Diagram ($p_k = .25$)

Solution	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$
$\mathbf{z}_1$					
$\mathbf{z}_2$	-				
$\mathbf{z}_3$	-	+			
$\mathbf{z}_4$	-	-	-		
$\mathbf{z}_5$	-	-	-	+	
$\mathbf{z}_6$	-	-	-	=	-

When comparing solution $\mathbf{z}_2$ and $\mathbf{z}_3$, the following procedure would be taken to reach the conclusion (-,+,-):

1. $ARN_2 = ARN_3$
    a. $\mu NPD_2^{.25} > \mu NPD_3^{.25} \rightarrow$ (-,,)
2. $f^1(\mathbf{x^2}) < f^1(\mathbf{x^4}) \rightarrow$ (-,+,)
3. $f^2(\mathbf{x^2}) > f^2(\mathbf{x^4}) \rightarrow$ (-,+,-)

Therefore, $\mathbf{z}_3\ P\ \mathbf{z}_2$ is the appropriate conclusion.

As seen in Table 4.13, if there is no preference between the performance metrics from the DM, but a maximum objective performance degradation level of 25% is in place, the best solution is $\mathbf{z}_1$ followed by $\mathbf{z}_3$ then $\mathbf{z}_2$.

Tables 4.14 and 4.15 show the final results displaying the preference matrices for both the $p_k = .35$ and $p_k = .50$.

Table 4.14: Preference Diagram ($p_k = .35$)

Solution	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$
$\mathbf{z}_1$					
$\mathbf{z}_2$	-				
$\mathbf{z}_3$	+	+			
$\mathbf{z}_4$	-	-	-		
$\mathbf{z}_5$	-	-	-	+	
$\mathbf{z}_6$	-	-	-	-	-

From Table 4.14, the chosen solution would be $\mathbf{z}_3$, with $\mathbf{z}_1$ being the second best alternative followed by $\mathbf{z}_2$, $\mathbf{z}_5$, $\mathbf{z}_4$, and $\mathbf{z}_6$, in that order.

Table 4.15: Preference Diagram ($p_k = .50$)

Solution	$\mathbf{z}_1$	$\mathbf{z}_2$	$\mathbf{z}_3$	$\mathbf{z}_4$	$\mathbf{z}_5$
$\mathbf{z}_1$					
$\mathbf{z}_2$	-				
$\mathbf{z}_3$	+	+			
$\mathbf{z}_4$	-	-	-		
$\mathbf{z}_5$	-	-	-	-	
$\mathbf{z}_6$	-	-	-	-	-

From Table 4.15, the chosen solution would again be $\mathbf{z}_3$, with $\mathbf{z}_1$ being the second best alternative followed by $\mathbf{z}_2$.

The preceding simple application of this decision aid assumes that the DM has no preference between the performance metrics, and that any weights associated with them are equal. If this isn't the case, then a cost based approach can be implemented applying the weights reflecting the preferences of the decision maker. These cost estimations may then be normalized as well as the objective performance values to facilitate meaningful comparisons in the application of this decision aid technique.

5. Algorithms

5.1 Approach

Two different solution methodologies will be employed in this work. The first solution approach will be an implementation of a state of the art scalarization, multi-criteria methodology. Scalarization algorithmic techniques effectively transform multi-objective optimization problems into mono-objective ones via the use of a scalar multiplier, or through the conversion of all but one objective into constraints. The methodology implemented in this study will be a recent innovation in scalarization algorithms found in Ehrgott (2006). This solution approach combines and captures the strengths of the two most popular scalarization methodologies, the weighted-sum-of-objective-functions-method, (Gass & Saaty 1955), and the ε-constraint methodology, which was first presented in Heimes et al. (1971).

The second solution approach to be used is an implementation of the metaheuristic algorithm commonly known as GRASP, or Greedy Random Adaptive Search Procedures. These algorithms all feature a stochastic component, which guides the search through the feasible region, or the image set Z, followed by a subroutine that finds a locally optimal solution. The output of a GRASP algorithm, in the context of multi-criteria optimization, is a set of solutions approximating the Pareto frontier. Because this is a heuristic, attaining the entire set of Pareto efficient solutions for a given discrete optimization problem is not assured. Given this reality, a series of metrics has been derived in Van Veldhuizen (2000) to evaluate the performance of multiobjective heuristic algorithms, one of which is applied in this work.

5.2. Scalarization Methodologies

The weighted-sum-of-objective solution methodology is a convenient technique for several reasons. Firstly, it is rather simple to implement. Secondly, the outcome at each iteration of the algorithm is a solution that is provably Pareto efficient. Miettinen (1999) provides a more detailed discussion of the merits and various proofs associated with this methodology.

A significant drawback of this solution approach is that if an objective function is not convex, as is the case here and in most combinatorial optimization problems, then it is impossible to locate some points on the Pareto frontier. This can lead to a set of efficient solutions that don't adequately approximate the entire frontier. Therefore, for some discrete optimization problems featuring an image space which is non-convex, it may not be appropriate to use the weighted-sum-of-objectives in an a-postiori methodology, as is the case here.

Given a vector of weights $\boldsymbol{\omega}_k$, one weight for each objective $k \in K$, problem (2.1) can be transformed using the weighted-sum-of-objectives method. This transformation is given below in problem (5.1).

$$\begin{aligned} &minimize \;\; \omega_1 f_1(\mathbf{x}) + \omega_2\, f_2(\mathbf{x}) +, \dots, + \omega f_k(\mathbf{x}) \\ &subject\ to \;\; \mathbf{x} \in S \end{aligned} \tag{5.1}$$

Most implementations of problem (5.1) enforce the following property: $\sum_k \omega_k = 1$, with all $\omega_k > 0$. However, this restriction isn't mandatory, it is usually done as a way of prioritizing or valuing the objectives in regards to each other. In a-priori implementations of this algorithm, these weights can be the product of a pre-optimization analysis of the criteria. In most cases, the weights used typically sum to one.

Implementation of this algorithm is done through a re-optimization procedure where the weights of the objectives are iteratively perturbed. However, incommensurability of units can lead to excessively long solution times if a full exploration of the Pareto frontier is desired. To address the incommensurability of units between the two competing objectives, a normalized version of the objectives can be used (see Section 4.4).

The ε-constraint works by preserving one of the objectives in a multi-objective problem as the objective function, while converting the others into inequality constraints. Problem (2.2) can be transformed using the ε-constraint methodology as follows:

$$\begin{aligned} &Minimize \quad f_1(\mathbf{x}) \\ &s.t. \\ &\qquad \vec{g}(\mathbf{x}) \leq 0 \\ &\qquad \vec{h}(\mathbf{x}) = 0 \\ &\qquad f_2(\mathbf{x}) \leq \varepsilon_2 \\ &\qquad \vdots \\ &\qquad f_k(\mathbf{x}) \leq \varepsilon_k \\ &\text{where } \mathbf{x} \in \mathbf{R}^n, \vec{g}(\mathbf{x}) \in \mathbf{R}^m \text{ and } \vec{h}(\mathbf{x}) \in \mathbf{R}^p \end{aligned} \tag{5.2}$$

To develop the Pareto frontier using the methodology given by problem 5.2, the set of right hand side values associated with the constraintized objectives functions (ε_k) are iteratively perturbed in a re-optimization procedure, effectively finding all non-dominated solutions.

The main advantage of the weighted-sum-of-objectives method is its ease of implementation and the speed with which the Pareto frontier can be generated. The disadvantage of this methodology is that it is not possible to find any non-dominated solutions not located on the convex hull of the objective space. The ε-constraint methodology, however, is able to find all non-dominated solutions, even the "convex dominated" points, but is harder to solve and takes longer to implement (Miettinen 1999). Point $\mathbf{z}_7$ in Figure 5.2.1 below is an example of a convex dominated solution.

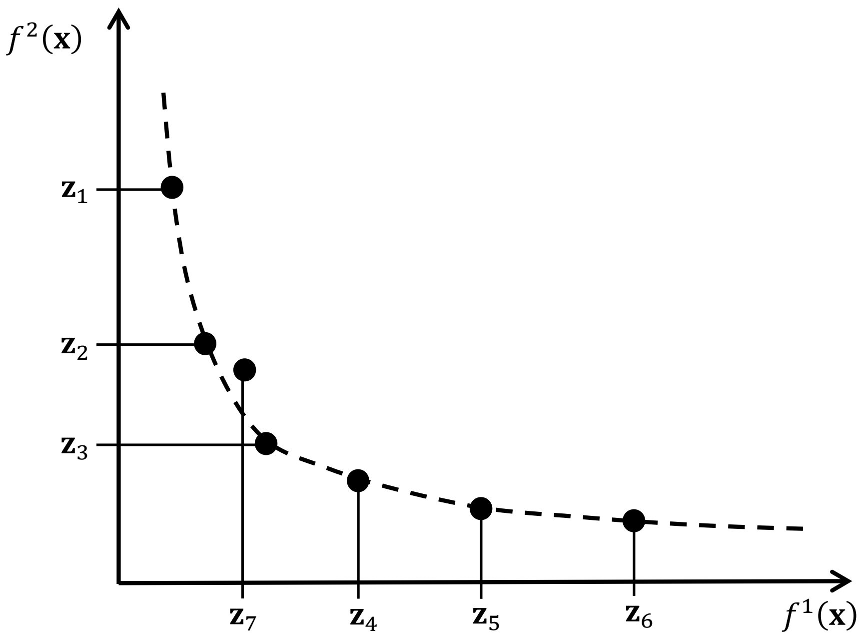


Figure 5.2.1 Example approximation of a Pareto face in bi-objective discrete optimization

Ehrgott (2006) presented a scalarization solution technique which combines the weighted-sum-of-objectives and the ε-constraint methods, capturing the strengths of both methods. This methodology is called the elastic constraint method.

The elastic constraint methodology, on the other hand, is both easy to solve, like the weighted-sum-of-objectives, and capable of finding all efficient solutions, like the ε-constraint method, Ehrgott (2006). This methodology makes the constraintized objectives elastic which results in easier solvability due to the possibility of upper bound violation at a penalty cost. Problem 2.2 is transformed below with the elastic constraint method.

$$\begin{aligned} &Minimize \quad f_1(\mathbf{x}) + \mu_2 L_2 +, \dots, +\mu L_k \\ &s.t. \\ &\vec{g}(\mathbf{x}) \le 0 \\ &\vec{h}(\mathbf{x}) = 0 \\ &f_2(\mathbf{x}) + L_2 - S_2 = \varepsilon_2 \\ &\quad . \\ &\quad . \\ &\quad . \\ &f_k(\mathbf{x}) + L_k - S_k = \varepsilon_k \end{aligned} \tag{5.3}$$

where $\mathbf{x} \in \mathbf{R}^n$, $\vec{g}(\mathbf{x}) \in \mathbf{R}^m$, $\vec{h}(\mathbf{x}) \in \mathbf{R}^p, L_k \ge 0,$ and $S_k \ge 0$

In 5.3 above, the objective minimizes $f_1(\mathbf{x})$ plus the sum product of an additional penalty factor and a positive slack value for all constraintized objectives which fail to meet a required performance level ε_k.

The following procedure is used to implement the elastic constraint methodology in this work.

Figure 5.1: Elastic Constraint efficient set generation algorithm

Procedure ElasticConstraint
1. EfficientSet = []
2. Data Read_Input()
3. Z_1 = [min_cost, worst_uncoverage]
4. **Call** Update(Z_1, EfficientSet)
5. Z_2 = [min_uncoverage, worst_cost]
6. **Call** Update(Z_2, EfficientSet)
7. ε = worst_uncoverage
8. **While** ε > min_uncoverage **do**
9. Z_i = **Call Model**(ε)
10. **Call** Update(Solution, EfficientSet)
11. ε = Z_i(coverage) - 1
12. **End_While**
13. **Return** ParetoSet
End_ElasticConstraint

The algorithm given in Table 5.1 begins by finding the optimal values of each individual objective function by solving the single objective problems. These are then used to calculate the ideal and nadir points. In this case, the constraintized objective function is penalized with the surplus variable (as opposed to the slack variable as seen in problem 5.3). Iterating from the worst performance to the best performance on the objective of coverage, the normalized version of the models are solved and the resulting solutions are stored in the Pareto efficient set. The step increment at each iteration of the algorithm is equal to the previous iteration's uncoverage value minus one. This procedure generates the entire Pareto frontier using an optimal search methodology.

5.3. Heuristic Algorithm

For this thesis, a multi-objective GRASP algorithm (MOG) will be implemented. GRASP is a multi-start metaheuristic involving a construction phase and a local search phase (Feo & Resende 1989). In this methodology, a solution is generated through a greedy construction phase and improved through a local search phase. A facility location problem for multi-echelon location was recently solved using GRASP in Montoya-Torres et al. (2011). The approach taken for this study will be similar, but adapted for a multi-criteria implementation.

Given below in Figure 5.2 is the generic pseudocode for the GRASP procedure employed here.

Figure 5.2: Pseudocode of the GRASP Procedure

Procedure GRASP(Max_Iterations, α)
1. Best_Solution = 0
2. Data Read_Input()
3. **For** k=1,…,Max_Iterations **do**
4. **Call** Update(Solution, Best_Solution)
5. Solution GreedyRandomizedConstruction(Seed)
6. **If** feasible == false **do**
7. Solution == Repair(Solution)
8. **EndIf**
9. Solution LocalSearch(Solution)
10. **Call** Update(Solution, Best_Solution)
8. **EndFor**
9. **Return** Best_Solution
End GRASP

The construction phase of the algorithm terminates with the selection of all candidate facilities to be opened. At this point, a local search takes place resulting in a locally optimal solution. The procedure being taken to implement a MOG for this problem is based upon the concept outlined in the algorithm given in Table 5.1, where the search space is iteratively altered and a series of locally optimal solutions are found. This technique is referred to as iterated domain restriction and has been successfully applied in a number of papers featuring local search or multi-start algorithms in multi-criteria optimization, Murphy et al. (2000), Pasiliao (1998), Deb (2001).

For the MOG applied in this work, the GRASP procedure displayed in figure 5.2 will be repeated in the same fashion as the elastic constraint methodology, by iteratively changing the required uncoverage level and saving the best solution found at each iteration while respecting the constraints on the problem until uncoverage minimization is reached. The psudocode for this procedure is given below in table 5.3.

Figure 5.3: MOG Algorithm

Procedure MOG(max_iterations, α)
1. EfficientSet = []
2. Data Read_Input()
3. Z_1 = [min_cost, worst_uncoverage]
4. **Call** Update(Z_1, EfficientSet)
5. Z_2 = [min_uncoverage, worst_cost]
6. **Call** Update(Z_2, EfficientSet)
7. ε = worst_uncoverage
8. **While** ε > min_uncoverage **do**
9. Z_i = **Call GRASP**(max_iterations, α)
10. **Call** Update(Z_i, EfficientSet)
11. ε = Z_i(coverage) - 1
12. **End_While**
13. **Return** ParetoSet
End_MOG

The MOG is a stochastic local search algorithm that works by generating a restricted candidate list during the construction phase by evaluating the fitness of each candidate element by the performance of the objective function given the previously selected elements. In a greedy fashion, the best element is chosen and added to the list as well as a subset of the other candidate elements, the size of which is driven by the parameter α. The use of this parameter α for the generation of the restricted candidate list (RCL) is given below.

$$RCL = \left\{c \in C \middle| Z_i(c) \leq Z_i^{min} + \alpha\left(Z_i^{max} - Z_i^{min}\right)\right\}$$

where $\alpha \in [0,1]$.

The primary determinant of the performance of the GRASP algorithm is the selection of the α parameter, effectively setting the size of the RCL. If this parameter is set too large, the resulting solutions are too random. If it is set too small, then the solution tends to be consistent with a pure greedy approach. Finding the appropriate value to utilize here is paramount for high performance of the algorithm.

Each iteration of the construction phase of the GRASP heuristic is repeated until the max iterations value is reached. For the implementation used here, the GRASP procedure halts when all of the facilities have been selected (P, and/or Q in this case), and the MOG ends with the generation of the Pareto frontier.

The repair phase of this algorithm is especially important, and also contributes to the performance of the algorithm. For this work, this entails repairing a solution which violates the minimal coverage restrictions. A unique repair procedure was developed for this work. The psudocode for this repair function is given below in Figure 5.4.

Figure 5.4: Repair Function

Procedure Repair(Solution, Candidates)
1. **While** Solution(coverage) > min_uncoverage **do**
2. **Call** Update(c_{out} ∈ Solution, $c_{in} \in Candidates$)
3. **End_While**
4. **Return** Solution

End_Repair

In step 2 of the function given in Figure 5.4, the candidate location c_{in} is added into the solution while c_{out} is removed. The candidate element selected to leave the solution (c_{out}) is the facility in the solution which covers the least amount of demand. The incoming facility (c_{in}) is the one location in the list of candidates not in the solution, which increases demand coverage beyond what was lost via the removal of c_{out} at the smallest increase in cost. If minimal coverage is satisfied, the solution is sent back to the GRASP procedure, otherwise the process is repeated with the remaining locations (not c_{in}) currently in the solution being considered for removal.

For the local search step of the MOG, in this work, a substitution procedure similar to that outlined in Daskin (2013) is implemented. However, unlike the method provided in Daskin (2013), this substitution procedure is done after the construction phase is completed, as is conventional in most GRASP implementations. During the substitution phase, each selected facility is re-evaluated with the change in objective value found if facility $j \in J$ is replaced with any of the $|J| - 1$ facilities not already present in the solution. The best improvement found during this phase, if any, is retained, and the procedure is repeated until all selected locations have been evaluated.

Additionally, in order to inhibit the repeated discovery of previously identified non-dominated solutions, a tabu list (Glover 1986) was incorporated in the local search heuristic. This prevents the algorithm from allowing a previously chosen element in the candidate location pool to re-enter the solution for a set number of iterations. After some experimentation, a tabu list of size 10 was found to be the most effective for the data set considered in this work. Future research in this area will focus on formally evaluating the size of the tabu search lists and comparing other local search techniques in a multi-objective GRASP framework across multiple datasets.

The assignment of demand is a critical step before the evaluation of each candidate solution. In previous approaches to applying GRASP in location problems, a network model linear programming problem was solved as a sub-routine in the solution procedure, (Montoya-Torres et al. 2010, Montoya-Torres et al. 2011). For this work, however, the assignment of demand is done as a subroutine of the algorithm, which enforces the mandatory service restrictions developed in this work for multi-level location problems. The psudocode of this subroutine is below.

Figure 5.5: Solution Evaluation Procedure

```
Procedure SolutionEvaluation(W_Array, P_Array, Flow_Type)
TotalDistance, TotalCoverage = 0, 0
If Flow_Type == 1 do
Combined = W_Array ∪ P_Array
For i in 1:length(Customers)
temp = min(Distance[i,  j in Combined])
TotalDistance += temp
If temp <= 500
TotalCoverage += Demand(i)
End_If
End_For
Else
For i in 1:length(Customers)
temp = min(Distance[i, j in W_Array, k in P_Array])
TotalDistance += temp
If temp <= 500
TotalCoverage += Demand(i)
End_If
End_For
End_IfElse
Return TotalDistance, TotalCoverage
End_SolutionEvaluation
```

The Solution Evaluation procedure described above works by first identifying the flow type (1 == multi-flow) before “assigning” demand prior to criteria performance evaluation. The creation and retention of arrays of binary variables reflecting the allocation of demand for every single evaluation of a candidate solution unnecessarily increases the overhead and runtime of the algorithm. What this procedure does instead is identify the total distance by finding the assignment that is most economical, while iteratively updating the total demand covered, given vectors of selected facilities and plants. The procedure is polynomial in time at the worst case, but it runs even faster in practice due to the fact that the maximum number of comparisons required are 49 x 3 x 2 (number of nodes x largest problem size in this work).

What follows is a formal definition of a metric used to evaluate the performance of the MOG implemented in this work.

5.4. Metaheuristic Solution Method Evaluation

Collete & Siarry (2004) provide a set of performance metrics for evaluating the Pareto frontier of a multi-objective optimization problem solved with metaheuristic algorithms. The majority of the techniques given in this monograph are based upon the work found in Van Velduizen (2000). The approaches given therein will be applied to evaluate the GRASP to be implemented in this work. The notation used here is similar to that found in Collete & Siarry (2004).

5.4.1. Error Ratio

This metric is a measurement of the nonconvergence of a heuristic solution approach toward the optimal Pareto set by examining both the optimal and heuristic trade off surfaces and the elements of each set.

$$E = \frac{\sum_{l=1}^{L} e_l}{|L|}$$

where $|L|$ is the cardinality of the optimal Pareto set, and

$$e_l = \begin{cases} 0 \text{ if } l \text{ in the Pareto efficient set is also in the heuristic tradeoff surface} \\ 1 \text{ otherwise} \end{cases}$$

The closer this evaluation metric is to 0, the more the heuristic solution set has converged toward the optimal tradeoff surface.

6. Computational Results

6.1. Data Inputs

A series of ten multi-level location problems will be solved, each with varying numbers of warehouses and plants to locate, capturing a variety of scenarios under differing budget restrictions. For the computational results presented in this work, the "Sortcap" data set (Daskin 1995) consisting of 49 nodes representing the 48 contiguous United States and Washington, D.C. was used. The location of each node is the capital of their respective state where demand is equal to a 1990 population census totaling 247,051,601. Great circle distances are used, and all 49 nodes are candidate facility locations for both warehouses and plants. The Julia programming language (Bezanson et al. 2012) was used to implement all solution methodologies used in this work, with the Gurobi solver being called to solve the elastic constraint models. Table 6.1 is a listing of all scenarios evaluated.

Table 6.1: Scenarios Considered

Scenario	*Flow (S or M)*	*Warehouses (P)*	*Plants (Q)*
1	M	1	1
2	S	2	1
3	S	2	2
4	S	3	1
5	M	2	1
6	S	3	2
7	S	4	1
8	S	5	1
9	M	2	2
10	M	3	2

As seen in Table 6.1, the multi-level location problem where *P* warehouses and *Q* plants are located (model 3.3) is considered in the computational exercises of this work. Both single flow and multi-flow variants of the problem are evaluated throughout the ten scenarios analyzed here.

The coverage distance is set at 500 miles for all scenarios, a relatively conservative estimate of the maximum travel distance of a freight vehicle in one day (considering the maximum driving time allowed in a day), slightly offset by the fact that big circle distances tend to understate road travel distances. At a coverage distance of 500 miles, all demands can be covered with five facilities. For this reason, the last scenarios considered are $P = 5$, $Q = 1$, single flow and $P = 3$, $Q = 2$, multi-flow.

6.2. Elastic Constraint Methodology Results

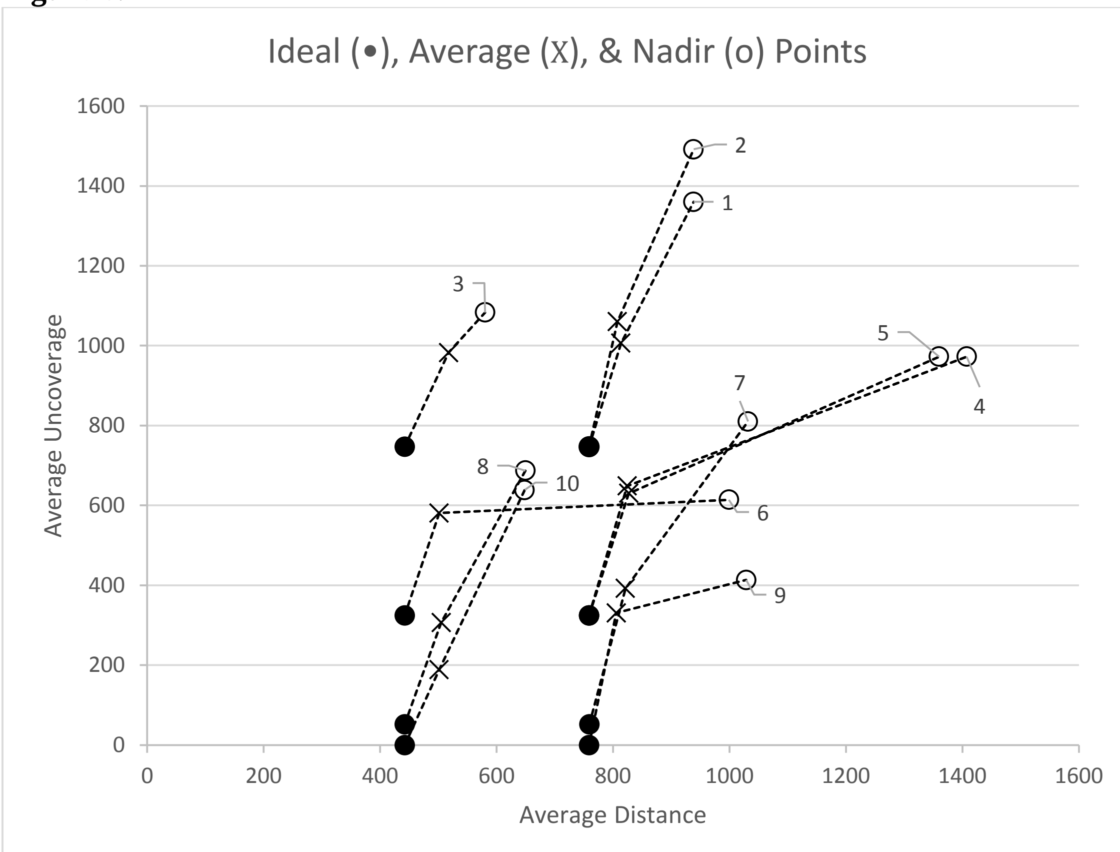
In order to find all non-dominated solutions in the trade-off of average distance and coverage, the elastic constraint methodology discussed in Chapter 5 was applied to all scenarios. Given below in Table 6.2 and are the summary results of this analysis.

Table 6.2: Summary Results

Scenario	*Average Distance*	*Average Uncoverage*
1	808	1060
2	814	1006
3	518	982
4	824	649
5	827	630
6	502	581
7	821	392
8	806	331
9	505	307
10	501	189

The results presented in Table 6.2 are given in descending order from worst to best average service level. Depicted in Figure 6.1 are the Ideal, Average, and nadir points for all scenarios.

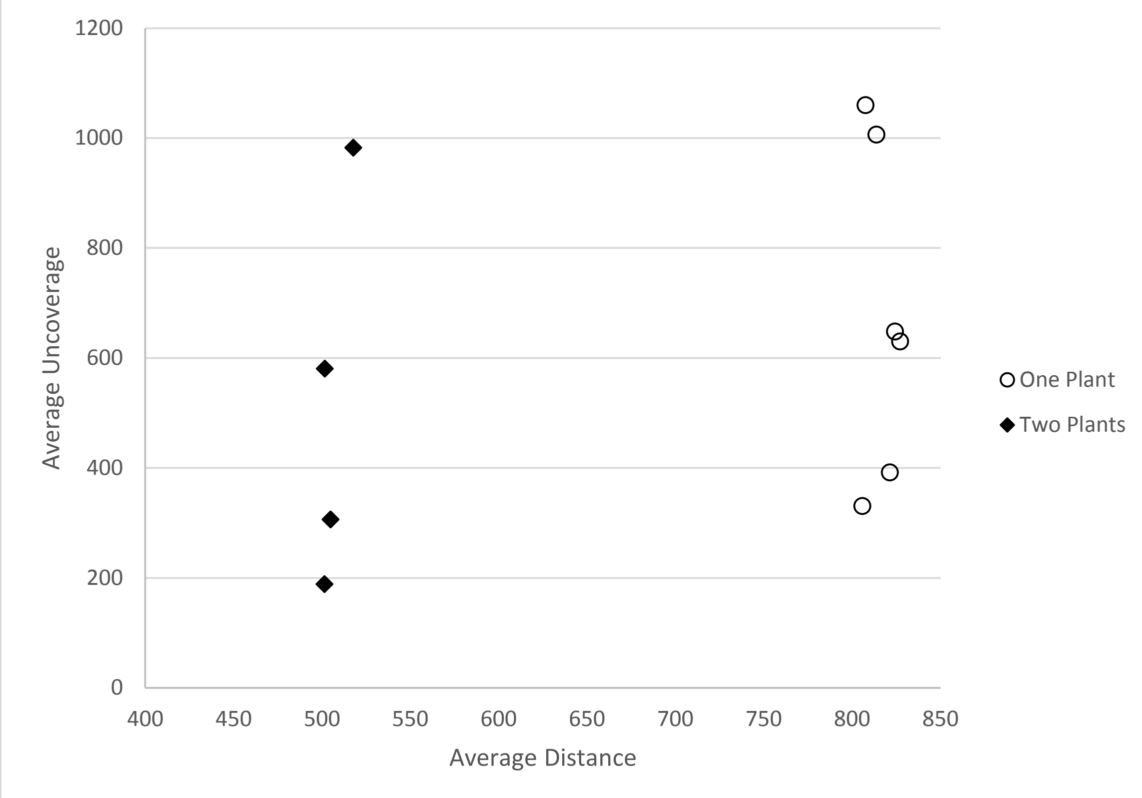
Figure 6.1



As seen in Figure 6.1, the average points utilized in comparing the scenarios in this analysis exhibit the same pattern as the ideal points. However, in each scenario, the average point is usually feasible and is always very close to or on the frontier.

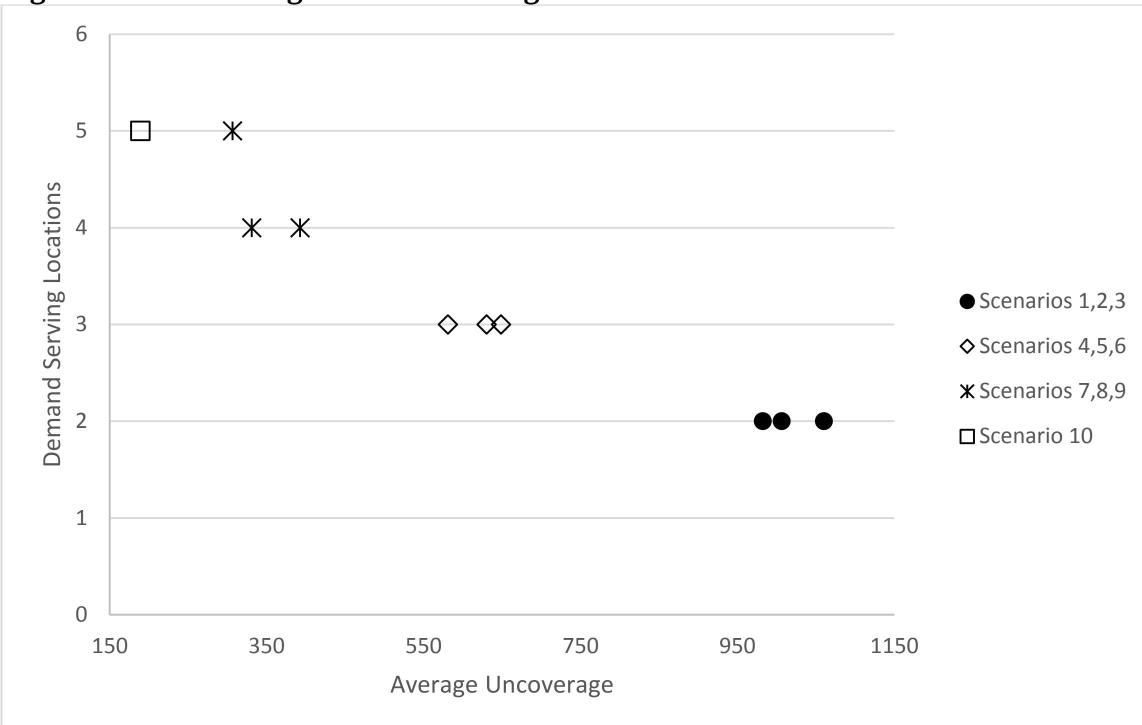
As Table 6.2 indicates, average distance doesn't always decrease with average uncoverage level across these scenarios. The cause of this is the existence of an extra plant. Figure 6.2 below illustrates this finding.

Figure 6.2: One Plant vs. Two Plant Solutions



Clearly, an extra plant significantly reduces the average distance in multi-level location problems. In fact, average distance across all efficient solutions varies little among scenarios with the same number of plants. Additionally, Table 6.2 indicates that comparable or superior service level is achievable through locating fewer facilities with the restriction that all of them be distribution capable. Figure 6.3 shows, in addition to the impact that varying the amount of demand serving locations has on the average uncoverage levels of the problem.

Figure 6.3: Increasing Demand Serving Locations



As seen in Figure 6.3, an expected increase in service levels occurs as the number of demand serving locations increase. However, an interesting finding can be seen by comparing scenarios 8 and 9. Specifically, a slightly better average service level is achievable by opening four locations (2/2/M) as opposed to six (5/1/S). Granted, opening a distribution capable plant is generally more costly than a warehouse, it may not be as costly as opening three warehouses in some industries.

The work in Shen & Daskin (2005) highlights the importance of a multi-criteria approach in facility location problems. They found that significant improvements in service levels can almost always be achieved at a minor increase in cost above the cost minimizing level. The results of this work support that finding. Table 6.3 presents these findings across all ten scenarios.

Table 6.3: Percentage Improvement in Service Level with Minor Increase in Distance

Scenario	*Increase in average Distance*	
	Less than %1	*Up to 5% increase*
1	25.7%	37.9%
2	18.2%	30.3%
3	-	1.0%
4	18.6%	37.2%
5	14.5%	33.1%
6	16.1%	16.9%
7	19.2%	39.0%
8	25.3%	45.7%
9	21.0%	30.3%
10	22.0%	33.0%
Mean	18.1%	30.4%

As Table 6.3 indicates, a significant improvement in service level is possible in all cases, with the exception of Scenario 2 (2/2/S). Another important finding of this work is that quite often, a weakly Pareto efficient solution will be found when solving the mono objective problem of distance minimization (or cost) relative to the conflicting criteria of service level. In this analysis, at a precision of one average mile, pure distance minimization resulted in the identification of a weakly Pareto optimal point in 9 of the 10 cases. In other words, alternate optimal solutions with better service levels are possible, yet could never be found in a mono-objective approach to the problem. These findings strongly support the use of multi-objective methodologies to solving multi-level facility location problems.

6.3. Multi-Objective Greedy Random Adaptive Search Results

As discussed in Chapter 5, the most influential parameter to the performance of a GRASP algorithm is the one which dictates the size of the restricted candidates list (α). In addition, a parameter controlling the stopping point of each run of the algorithm by controlling the maximum number of iterations contributes to the performance of the algorithm as well.

After extensive experimentation and parameter tuning, it was found that the best configuration is to set $\alpha = .1$ and β (max iterations) equal to 10 total iterations without improving the incumbent solution. This result supports the findings in Montoya-Torres et al. (2011), a recent work applying GRASP in a facility location paper. In that paper, a three echelon location problem is solved using a GRASP across a set of random instances of data of varying size from 8000 mixed integer programming (MIP) variables up to 500,000 MIP variables. Given below in Table 6.5 are the results of the MOG algorithm after 20 runs of each data set.

Table 6.5: Multi-Objective GRASP Results

Scenario	*Elastic Constraint*		*MOG*		
	Efficient Solutions	*Average Run Time (s)*	*Efficient Points Found*	*Average Run Time (s)*	*Error Ratio*
1	9	4.95	7	0.747	0..22
2	8	4.99	7	0.93	0.12
3	7	0.94	3	0.25	0.57
4	21	16.7	16	1.58	0.24
5	20	18.26	12	1.47	0.40
6	12	4.93	9	2.43	0.25
7	24	18.04	16	2.34	0.33
8	31	34.5	24	3.48	0.23
9	17	31.2	14	2.13	0.18
10	28	42.57	21	3.78	0.25
Mean	18	17.71	13	1.91	0.28

As seen in Table 6.5, the MOG algorithm identified 72% of the Pareto efficient solutions on average, as found with the elastic constraint methodology. Additionally, the mean solution time of the MOG is over ten times faster than using optimal search. Due to the fact that MOG is a stochastic algorithm, it is highly likely that more efficient points can be found if the algorithm is allowed to run more than 20 times. Further testing of the effectiveness of the key parameters (α and β), and a detailed analysis of the tuning of such parameters is needed. Lastly, an evaluation of a set of local search heuristics is need in a GRASP framework to compare the effectiveness of a variety of these techniques in location problems. Care should be taken in the interpretation of these findings, as these results are not generalizable across all multi-level location problems.

The MCDA solution selection techniques derived in Chapter 4 were applied to all scenarios. Given below in Table 6.4 is the results of this analysis.

Table 6.4: Selected Solutions/ARN_i

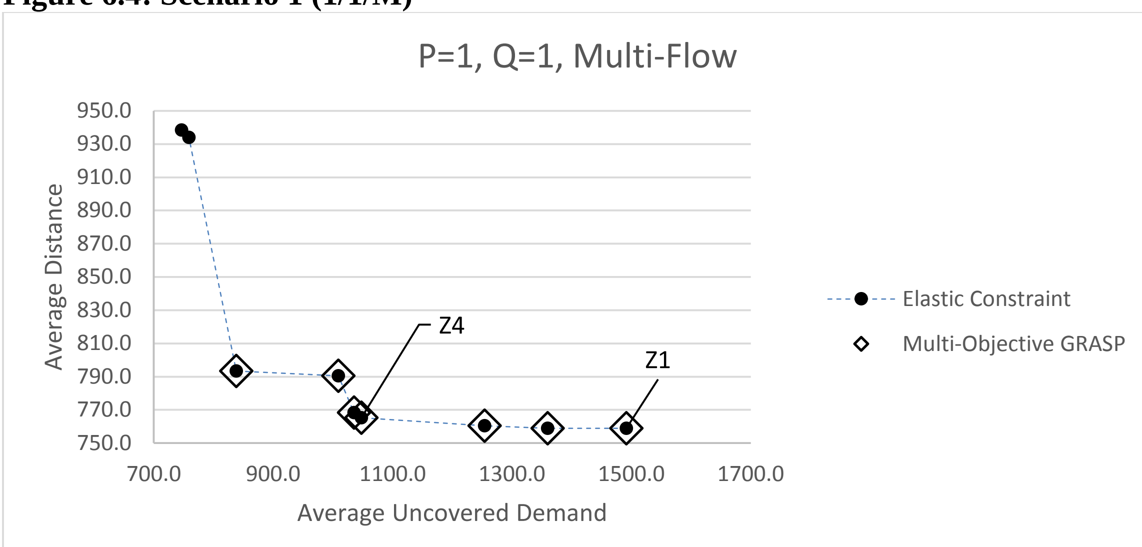
Scenarios	$p_k = .05$	$p_k = .25$
1	$\mathbf{z}_1/6$	$\mathbf{z}_4/8$
2	$\mathbf{z}_1/5$	$\mathbf{z}_3/7$
3	$\mathbf{z}_5/3$	$\mathbf{z}_4/6$
4	$\mathbf{z}_5/10$	$\mathbf{z}_{17}/18$
5	$\mathbf{z}_7/12$	$\mathbf{z}_{16}/19$
6	$\mathbf{z}_4/4$	$\mathbf{z}_{10}/10$
7	$\mathbf{z}_{10}/11$	$\mathbf{z}_{21}/20$
8	$\mathbf{z}_{20}/10$	$\mathbf{z}_{12}/26$
9	$\mathbf{z}_6/4$	$\mathbf{z}_{13}/16$
10	$\mathbf{z}_{10}/11$	$\mathbf{z}_{23}26$

Table 6.4 shows the selected solutions and the number of recourse networks available at the provided objective performance degradation level. As indicated, a most flexible solution, according to the metrics derived here, changed in every scenario when p_k was increased from 5% to 25%. Another interesting finding is that 45% of the

efficient solutions presented in Table 6.4 are “convex dominated,” or don’t lie on the convex hull of their respective solution spaces. This supports the conclusion that the use of more sophisticated search techniques is necessary when applying the MCDA techniques derived here in order to identify a more complete non-dominated set. Lastly, the incorporation of the mandatory service restrictions as defined in Chapter 3 have a negligible effect on total distance. This result coincides with another finding in Shen & Daskin (2005), where closest assignment restrictions resulted in negligible increases in total cost. What follows are illustrations of the efficient set of points, those found by the MOG algorithm, and the selected solutions based upon the criteria of flexibility across all ten scenarios considered in this work.

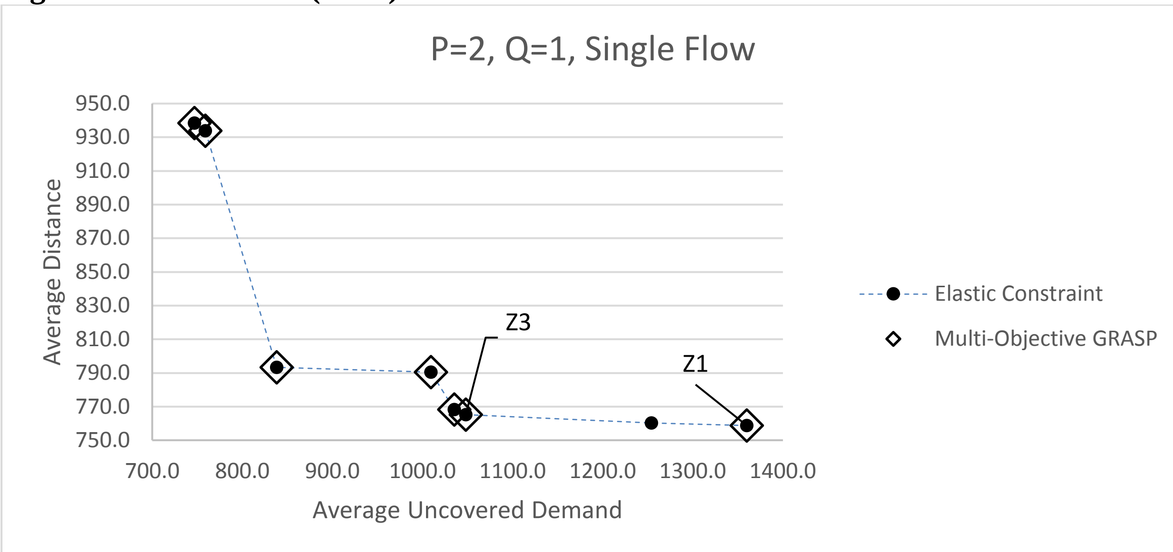
Depicted below in Figures 6.4 and 6.5 are the non-dominated frontiers from Scenarios 1 and 2.

Figure 6.4: Scenario 1 (1/1/M)



The two points highlighted in Figure 6.4 correspond to the selected solutions for Scenario 1 in Table 6.4. In this case, the most flexible solutions are near the cost (distance) minimizing extreme of the frontier.

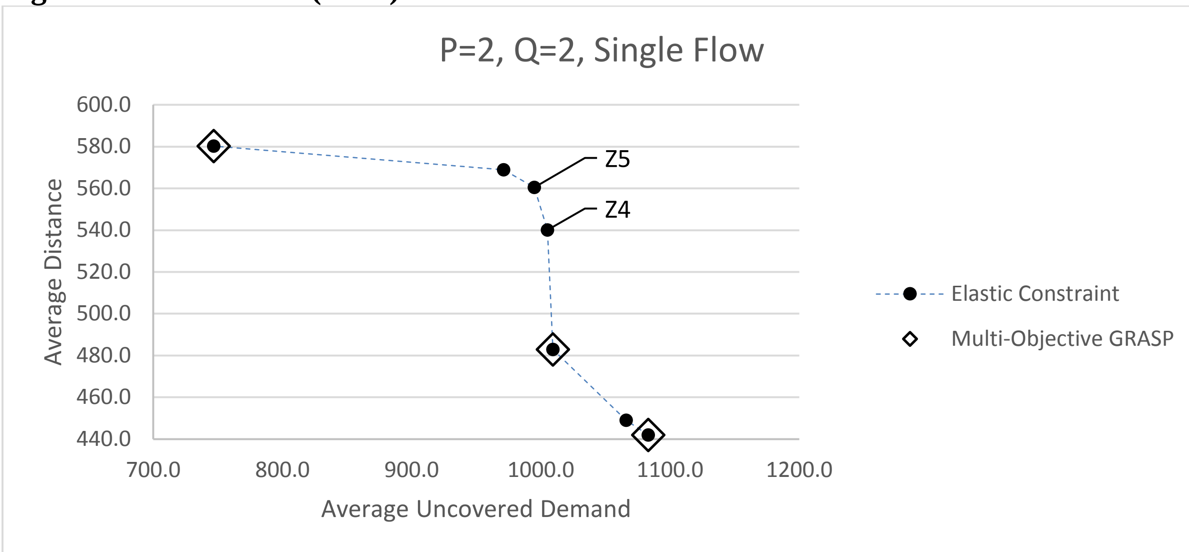
Figure 6.5: Scenario 2 (2/1/S)



Similar to Figure 6.4, the more flexible solutions perform well on the average distance objective. In fact, at an allowable objective degradation level of %5, the distance minimizing point is the most flexible solution for both scenarios 1 and 2. Also, when comparing the Figure 6.4 and Figure 6.5, it is apparent that the scenarios are similar and have nearly identical frontiers. The reason for this is because in the single flow problem, the best solution at each iteration usually involves co-locating plants with warehouses, and the chosen locations are the same as the multi-flow problem. This result captures the scenarios explaining efficient location in multi-level facility location problems in Figures 3.5 & 3.6 in chapter three.

Given below in Figure 6.6 is the frontier for Scenario 3. The MOG algorithm didn’t perform nearly as well here, only finding the two extreme points on the frontier and one compromise solution. Additionally, both selected solutions are convex dominated points in the recessed portion of the frontier. Unfortunately, neither were found with the MOG algorithm.

Figure 6.6: Scenario 3 (2/2/S)



Below are Figures 6.7 and 6.8, illustrations of two more Scenarios with similar frontiers.

Figure 6.7: Scenario 4 (3/1/S)

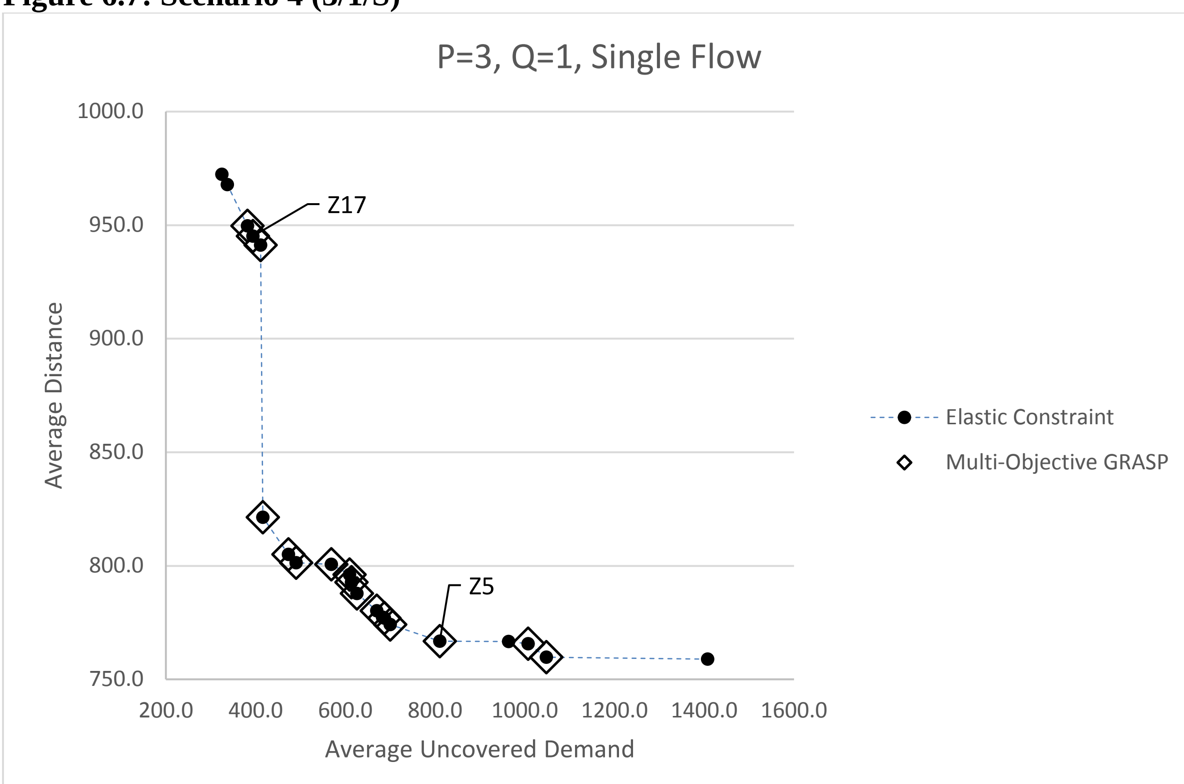
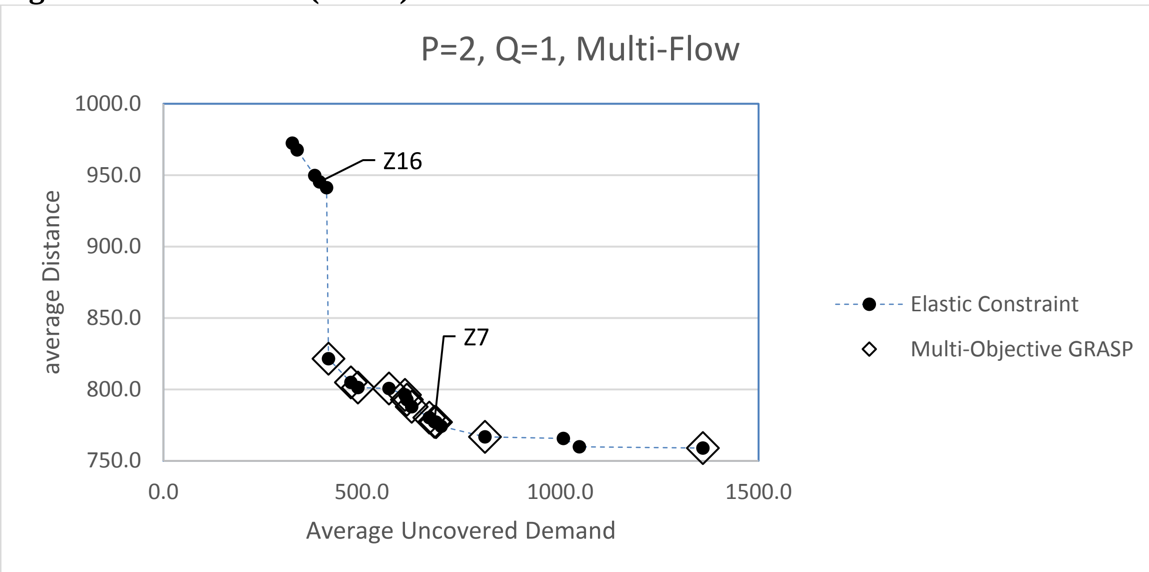


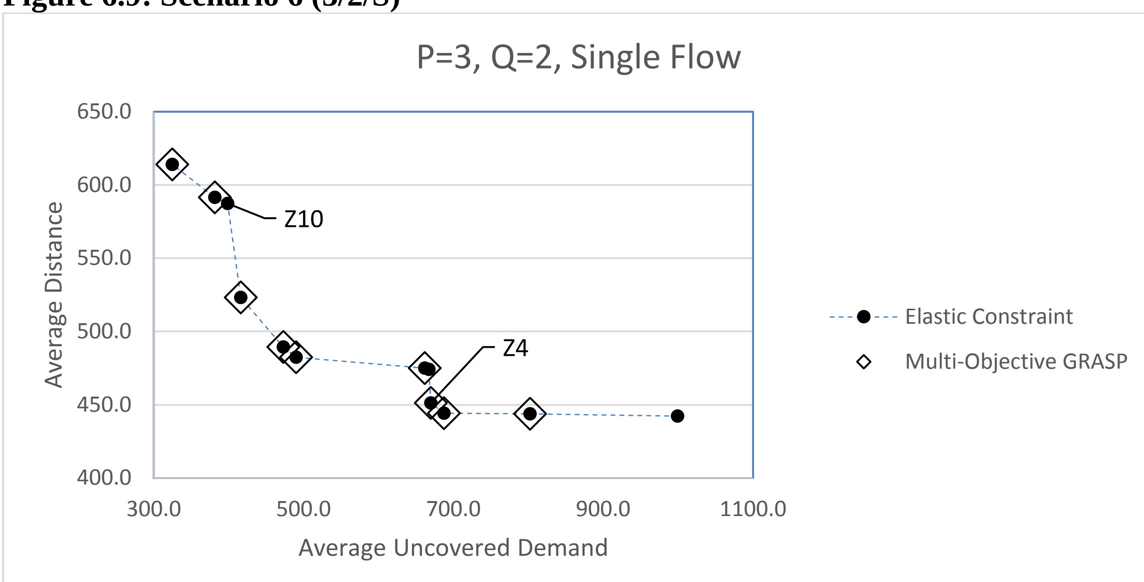
Figure 6.8: Scenario 5 (2/1/M)



As seen in Figures 6.7 and 6.8, the solutions identified as being the most flexible are in the same areas of the frontier. As the allowable objective function degradation increases, the selected solution is one near the service level maximizing point in both cases.

Presented below in Figure 6.9 is the frontier for Scenario 6.

Figure 6.9: Scenario 6 (3/2/S)



In this scenario, the selected solution jumps from one near the cost minimizing point to one near the service level maximizing point, similar to Scenarios 4 and 5. Given below are the frontiers for Scenarios 7 and 8.

Figure 6.10: Scenario 7 (4/1/S)

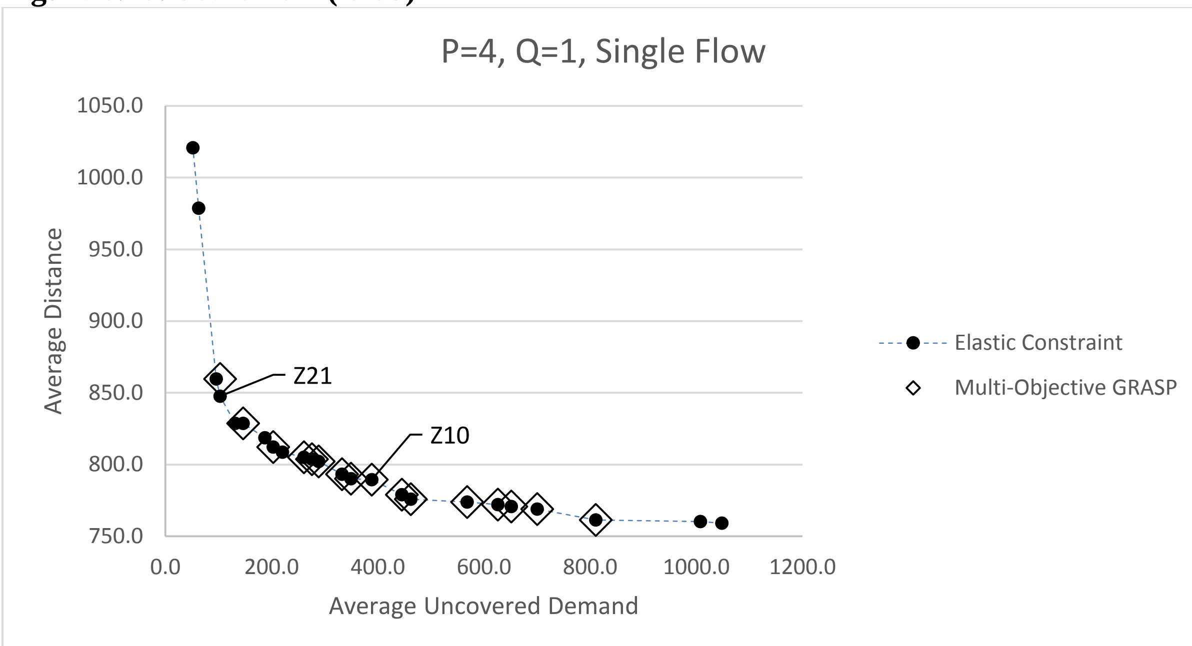
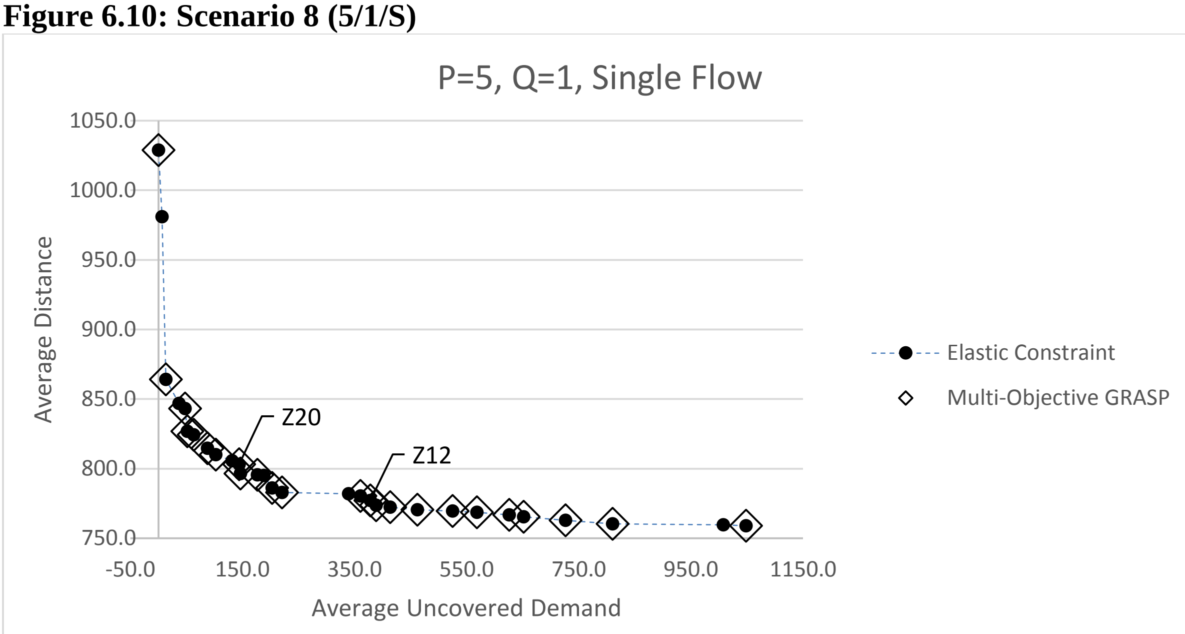


Figure 6.10: Scenario 8 (5/1/S)



In these scenarios, both selected solutions are near the center of the frontier among the compromise solutions. Additionally, in Scenario 8 the chosen solution moved toward the distance minimizing point as allowable objective performance degradation (p_k) was increased. The only other time this occurred was in Scenario 3. Given below in Figure 6.11 and Figure 6.12 are the frontiers for Scenarios 9 and 10.

Figure 6.11: Scenario 9 (2/2/M)

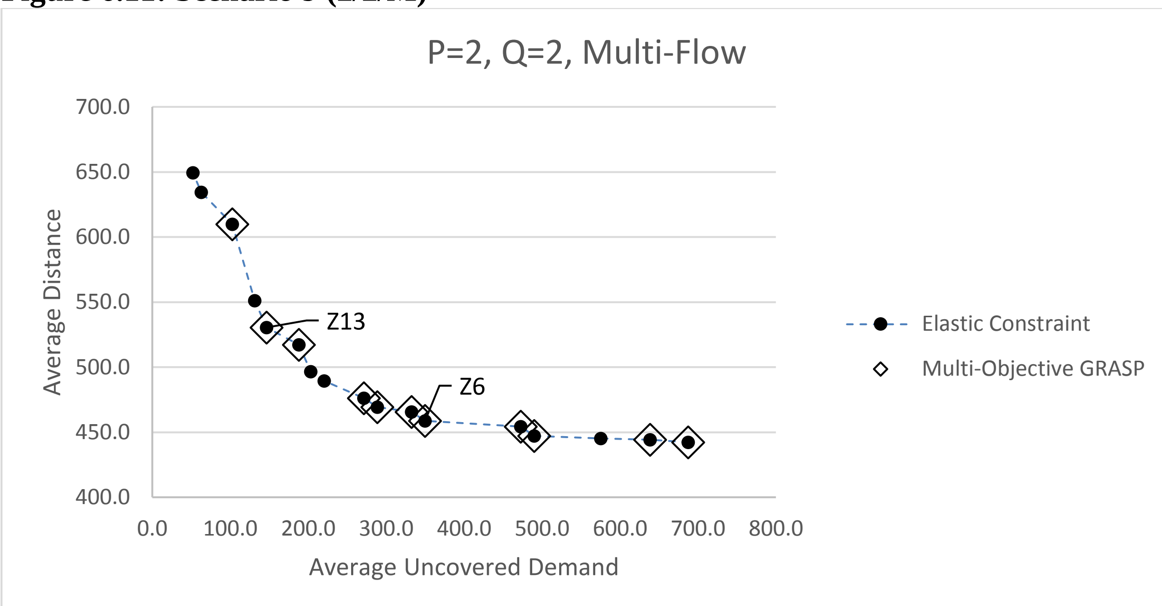
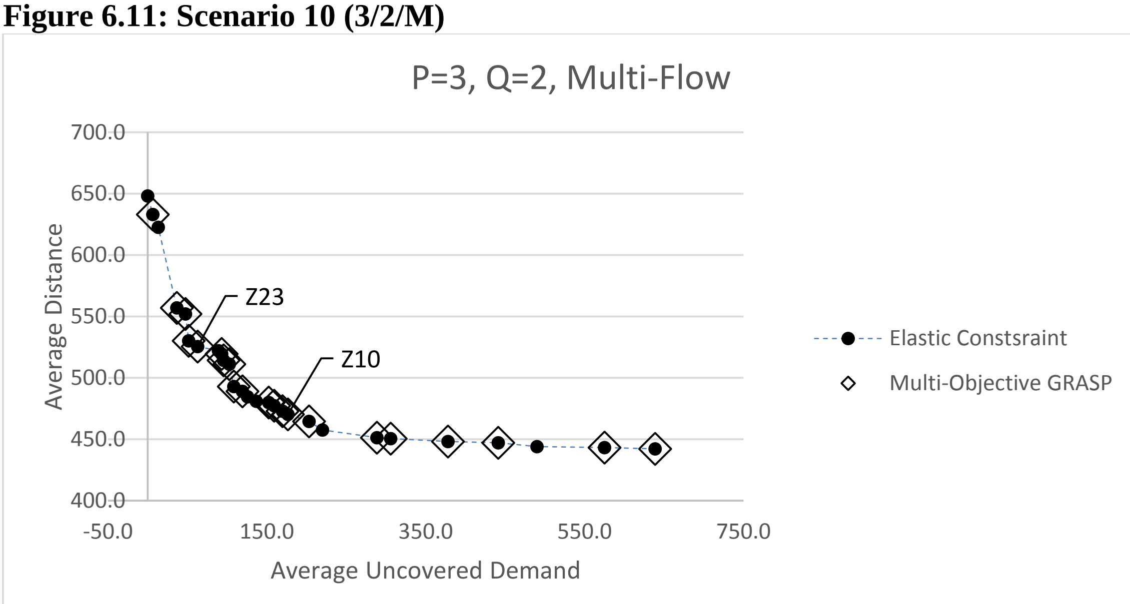


Figure 6.11: Scenario 10 (3/2/M)



In both Scenarios 9 and 10, the selected solutions are toward the center of the frontier among the compromise solutions. Additionally, it can be seen that the MOG algorithm did quite well discovering well over half of the points in both scenarios. However, the uncoverage minimization point was not found in both scenarios with the MOG algorithm.

The findings, contributions, and future research of this work is summarized in the following section. Additionally, managerial insights are provided based upon the results given here.

7. Conclusion

7.1 Findings, Contributions, and Future Research

There were three main goals of this research; each with a different purpose. The first purpose of this research was to contribute to location theory by developing new multi-level location models, which address the problem of efficient assignment of demands under a cost/service level trade-off, while enforcing mandatory service restrictions. A mandatory service policy is one in which any customer must be covered, if it is at all possible to do so, when given a set of selected locations. The task is then to choose the best subset of candidate locations at the ideal trade-off level of cost and customer service while enforcing mandatory service.

Historically, enforcing adequate service level, in the context given in this work, is usually addressed via closest assignment restrictions. However, closest assignment constraints can be overly restrictive, and may not be the most efficient means of addressing service level restrictions, especially in multi-level location problems. The models we developed here address this issue. Although the scenarios considered here are multi-objective location problems, the idea of mandatory service restrictions in distribution modeling can be applied in the mono-objective scenario as well, a simplification that will be explored in future research.

The classic approach of coverage-based modeling, a technique typically reserved for public sector location problems, has been successfully adapted to private sector distribution system design in this work. This is not to say that covering hasn't been used in private sector problems before, because it has. However, our approach is unique, and incorporates the distribution decisions which dictate the flow in the model, something which is all but ignored in most covering papers in the literature. This contribution is most significant, and will lead to a stream of research fully exploring this type of restriction in a variety of scenarios where existing service level modeling approaches in private sector location papers are compared and contrasted. Additionally, with the emergence of e-commerce and a general surge in focus on customer service in recent years, the idea of same day delivery is taking off in the business world. As such, modeling approaches to facility location in strategic network design should focus more on customer service.

The second purpose of this work is to contribute to the field of multi-criteria decision making in facility location. This was done by quantifying the similarity of distribution networks, an approach which is unique. In this work, the foundation for a multi-criteria decision aid in facility location was laid. The metrics designed capture the flexibility of any given set of location decision in regards to its neighboring efficient solutions. Quite simply, the solutions that are most flexible are the ones that can be altered efficiently to any other non-dominated solution via opening and closing locations. Because operating priorities and strategies can evolve, and a company's strategy can shift between cost and service. Selecting a distribution network that can be economically adapted to differing non-dominated solutions is ideal. This work begins to address this problem.

As alluded to periodically throughout this work, only using Hamming distances to quantify flexibility may not be the best approach. However, this work is merely a starting point in a new research stream. It remains to be seen how effective the metrics can be by themselves in practice. Regardless, the contribution of this thrust is in the initiation of a new research idea in modeling and capturing flexibility, and the starting point for measures meant to quantify the flexibility of a distribution network as a means of incorporating this facet in a decision methodology.

Future research in this area will focus on ways of including distance, assignment arcs, and product flows into the quantifying of network flexibility, as well as developing dynamic solution approaches to consider the criteria of flexibility during the search for non-dominated solutions. Additionally, real world applications are needed to test and refine these metrics. Although the analysis here was not cost based, the cost based approach was provided for the metrics given, and cost based applications seem to be especially promising areas of future research.

The third and final purpose of this research was to apply a multi-objective greedy random adaptive search metaheuristic to the multi-level, multi-criteria location problem, first such research to do so. The criteria of cost in the form of average distance, and customer service in the form of total demand serviced within a given service level was considered. Although GRASP has been applied in a handful of location papers, there have been no MOG algorithms applied in facility location.

The local search phase of the MOG is especially vital to the performance of the algorithm. The technique applied here is a simple facility exchange routine with a tabu search restriction. Implementing tabu search in this local search subroutine greatly improved the performance of the algorithm by limiting the amount of dominated solutions found at each iteration, thereby nearly eliminating the redundant discovery of previously found efficient solutions. This work contributes to the literature by presenting a unique MOG algorithm for the multi-level location problem and applying it with high success (72% of the non-dominated points found in just 20 runs).

7.2. Managerial Insights

What follows are some managerial insights from this research. Caution needs to be taken when interpreting these insights. Scenarios too dissimilar to the ones considered in this work may not be directly applicable.

1. Customer service level should not be disregarded in distribution system design. In almost every situation, a significant increase in customer service can be experienced at a nominal increase in cost.

2. Comparable performance can be achieved with fewer locations, as long as all locations can distribute to customers. This should be considered in real world distribution system design and redesign decisions. It may ultimately be more efficient to simply expand the plant locations to incorporate customer distribution operations than simply opening a new warehouse.

3. Opening additional plants can significantly decrease variable distribution cost in multi-level scenarios. In fact, the impact can be much more significant than opening additional warehouses.

4. Adding customer distribution capability to plants can lead to significant decreases in variable distribution cost and will always improve service level.

5. The cost minimization objective is usually highly inflexible. In other words, choosing to ignore customer service restrictions may lead to a distribution network that is very costly to reconfigure if an improvement in customer service is desired while maintaining Pareto efficiency.

6. The flexibility of a non-dominated solution can vary greatly across the frontier, even amongst points which are adjacent. This means that the average cost of a reconfiguration is not necessarily correlated in any way with the pursuit of cost minimization or customer service maximization.

7. Greedy Random Adaptive Search Procedures can be effectively applied in facility location problems. As problem sizes increase, optimal search techniques become impossible to use to solve location problems. Heuristics are used in these cases. The GRASP and the MOG are excellent alternative solution methods to optimal search in facility location.

Works Cited

Aardal, K. 1998. "Reformulation of capacitated facility location problems: how redundant information can help." *Annals of Operations Research* 82: 289-308.

Aardal, K., Chudak, F.A., Shmoys, D.B. 1999. "A 3-approximation algorithm for the k-level uncapacitated facility location problem." *Information Processing Letters* 72: 161-167.

Aardal, K., Labbe, M., Leung, J., Queyranne, M. 1996. "On the two-level uncapacitated facility location problem." *INFORMS Journal on Computing* 8: 289-301.

Addis, B., Carello, G., Ceselli, A. 2012. "Exactly solving a two-level location problem with modular node capacities." *Networks* 161-180.

Ageev, A. 2002. "Improved approximation algorithms for multilevel facility location problems." *Operations Research Letters* 30: 327-332.

Ageev, A., Ye, Y, Zhang, J. 2004. "Improved combinatorial spproximation algorithms for the k-level facility location problem." *SIAM Journal On Discrete Mathematics* 18: 207-217.

Alminyana, M., Borras, F., Pastor, J.T. 1998. "A new directed heuristic for the pq-median problem." *Location Science* 6: 1-23.

Altiparmak, F., Gen, M., Lin, L., Paksoy, T. 2006. "A genetic algiorithm approach for multi-objective optimization of supply chain networks." *Computers & Industrial Engineering.*

Amin, S., Zhang, G. 2013. "A multi-objective facility location model for closed-loop supply chain network under uncertian demand and return." *Applied Mathematical Modelling.*

Asadi, M., Niknafs, A., Ghodsi, M. 2008. "An approximation algorithm for the k-level uncapacitated facility location problem with penalties." In *CSICC 2008*, by H. Sarbazi-Azad, 41-49. Berlin Heidelberg: Springer-Verlag.

Azaron, A., Brown, K., Tarim, S., Modarres, M. 2008. "A multi-objective stochastic programming approach for supply chain design considering risk." *International Journal of Production Economics.*

Baray, J., Cliquet, Gerard. 2013. "Optimizing locations through a maximum covering/p-median hierarchical model: Maternity hospitals in France." *Journal of Business Research* 66: 127-132.

Barros, A.I., Dekker, R., Scholten, V. 1998. "A two-level network for recycling sand: a case study." *European Journal of Operational Research* 110: 199-214.

Barros, A.I., Labbe, M. 1994. "A general model for the uncapacitated facility and depot location problem." *Location Science* 2: 173-191.

Bezanson, J., Karpinski, S., Shah, V., Edelman, A. 2012. "Julia: a fast dynamic language for technical computing."

Bhattacharya, R., Bandyopadhyay, S. 2010. "Solving conflicting bi-objective facility location problem by NSGA II evolutionary algorithm." *International Journal of Advanced Manufacturing Technology.*

Bilgen, B., Ozkarahan, I. 2004. "Strategic tactical and operational production-distribution models: a review." *International Journal of Technology Management.*

Bottcher, H., Rembold, G.H. 1977. "Optimization model for a regional system of solid-waste and wastewater disposal which considers legal pollution standards." *Environment and Planning A* 9: 771-786.

Bumb, A. 2001. "An approximation algorithm for the maximization version of the two level uncapacitated facility location problem." *Operations Research Letters* 29: 155-161.

Bumb, A., Kern, W. 2001. "A simple dual ascent algorithm for the multilevel facility location problem." *Proceedings of the 4th international workshop on approximation algorithms* *for combinatorial optimization problems (APPROX'2001), Lecture notes in comoputer science.* Berlin: Springer. 55-62.

Calvo, A.B., Marks, D.H. 1973. "Location of health care facilities: an analytical approach." *Socio-Economic Planning Science* 7: 407-422.

Camm, J., Chorman, T., Dill, F., Evans, J., Sweeney, D. 1997. "Blending OR/MS, judgement, and GIS: Restructuring P&G's supply chain." *Interfaces* 128-142.

Caradona-Valdes, Y, Alvarez, A., Pacheco, J. 2014. "Metaheuristic procedure for a bi-objective supply chain design problem with uncertainty." *Transportation Research Part B.*

Cardona-Valdex, Y., Alvarez, A., Ozdemir, D. 2011. "A bi-objective supply chain design problem with uncertainty." *Transportation Research Part C.*

Chaabane, A., Ramudhin, A., Paquet, M. 2010. "Design of sustainable supply chains under the emission trading scheme." *International Journal of Production Economics.*

Chardaire, P. 1999. "Hierarchical two level location problems." In *Telecommunications network planning*, by B., Soriano, P. Sanso. Boston: Kluwer Academic Publishers.

Chardaire, P., Lutton, J.L., Sutter, A. 1999. "Upper and lower bounds for the two-level simple plant location problem." *Annals of Operations Research* 86: 117-140.

Charnes, A., Storbeck, J. 1980. "A goal programming model for the siting of multilevel EMS systems." *Socio-Economic Planning Science* 14: 155-161.

Chen, C., Yuan, T., Lee, W. 2007. "Multi-criteria fuzzy optimization for locating warehouses and distribution centers in a supply chain network." *Journal of Chinese Institute of Chemical Engineers.*

Cheshmehgaz, H. R., Desa, I. M., Wibowo, A. 2013. "A flexible three-level logistic network design considering cost and time criteria with a multi-objective evolutionary algorithm." *Journal of Intelligent Manufacturing.*

Chow, J.Y.J, Regan, A. 2011b. "Network-based real option models." *Transportation Research Part B* 682-695.

Chow, T.Y.J., Regan, A.C. 2011. "Real option pricing of network design investments." *Transportation Science* 50-63.

Chung, S., Myung, Y., Tcha, D. 1992. "Optimal design of a two-level hierarchical structure." *European Journal of Operational Research* 62: 105-115.

Church, R., Eaton, D.J. 1987. "Hierarchical location analysis using covering objectives." In *Spatial analysis and location-allocation models*, by A., Rushton, G., Gosh. New York.

Church, R., ReVelle, C. 1974. "The maximal covering location problem." *Papers in Regional Science.*

Collette, Y., Siarry, P. 2004. *Multiobjective Optimization: Principles and Case Studies.* Berling Heidelberg: Springer-Verlag.

Current, J., Min, H., Schilling, D. 1990. "Multiobjective analysis of facility location decisions." *European Journal of Operational Research* 295-307.

Daskin, M.S. 1995. *Network and Discrete Location: Models, Algorithms, and Applications.* New York: Wiley.

—. 2013. *Network and Discrete Location: Models, Algorithms, and Applications.* Edited by 2nd. Hoboken, New Jersey: Wiley.

Dehghanian, F., Mansour, S. 2009. "Designing sustainable recovery network of end-of-life products using genetic algorithm." *Resources, Conservation and Recycling.*

Desai, A., Sorbeck, J. 1988. "Characterization of the constraint in successively inclusive location hierarchies." *Environment and Planning B* 15: 131-141.

Dokmeci, V.F. 1979. "A multiobjective model for regional planning of health facilities." *Environment Planning A* 11: 517-525.

Dokmeci, V.F. 1977. "A quantitative model to plan regional health facility systems." *Management Science* 24: 411-419.

Dokmeci, V.F. 1973. "An optimization model for a hierarchical spatial system." *Journal of Regional Science* 13: 439-451.

Dokmeci, V.F. 1991. "Optimum location of hierarchical production units with respect to price-elastic demand." *Environment and Planning A* 23: 1671-1678.

Du, F., Evans, G. 2008. "A bi-objective reverse logistics network analysis for post-sale service." *Computers & Operations Research.*

Edgeworth, F.Y. 1987 (original edition in 1881). "Mathematical Physics." *University Microfilms International (Out of print).*

Efroymson, M.A., Ray, T.L. 1966. "A branch-bound algorithm for plant location." *Operations Research* 361-368.

Ehrgott, M. 2006. "A discussion of scalarization techniques for multiple objective integer programming." *Annals of Operations Research* 343-360.

Eitan, Y., Narula, S.C., Tien, J.M. 1991. "A generalized approach to modeling the hierarchical location-allocation problem." *IEEE Transactions on Systems, Management and Cybernetics* 21: 39-46.

Erkut, E. Daragiannidis, A., Perkoulidis, G., Tjandra, S. 2008. "A multicriteria facility location model for municipal solid waste management in North Greece." *European Journal of Operational Research.*

Erlenkotter, D. 1978. "A dual-based procedure for uncapacitated facility location." *Operations Research* 992-1009.

Espejo, L.G.A., Galvao, R.D., Boffey, B. 2002. "Dual-based heuristics for the hierarchical covering location problem." *Computers and Operations Research* 30: 165-180.

Farahani, R. Asgari, N. 2007. "Combination of MCDM and covering techniques in a hierarchical model for facility location: a case study." *European Journal of Operational Research.*

Farahani, R.Z., Hekmatfar, M., Fahimnia, B., Kazemzadeh, N. 2014. "Hierarchical facility location problem: models, classifications, techniques, and applications." *Computers & Industrial Engineering* 104-117.

Farahani, R.Z., SteadieSeifi, M., Asgari, N. 2012. "Multiple criteria facility location problems: A survey." *Applied Mathematical Modelling* 1689-1709.

Feo T.A., Resende M.G.C. 1989. "A probabilistic heuristic for a computationally difficult set covering problem." *Operations Research Letters* 67-71.

Feo, T.A., Resende, M.G.C. 1995. "Greedy randomized adaptive search procedures." *Journal of GLobal Optimization* 109-133.

Flynn, J., Ratick, S. 1988. "A multiobjective hierarchical covering model for the essential air service program." *Transportation Science* 22: 139-147.

Foneseca, M., Garcia-Sanchez, A., Ortega-Mier, M., Saldanha-da-Gama, F. 2010. "A stochastic bi-objective location model for strategic reverse logistics." *TOP.*

Galvao, R.D., Espejo, L.G.A., Boffey, B., Yates, D. 2006. "Load balancing and capacity constraints in a hierarchical location model." *European Journal of Operational Research* 172: 631-646.

Gao, L., Robinson, E.P. 1992. "A dual-based optimization procedure for the two-echelon uncapacitated facility location problem." *Naval Research Logistics* 39: 191-212.

Gao, L., Robinson, E.P. 1994. "Uncapacitated facility location: general solution procedure and computational experience." *European Journal of Operational Research* 76: 410-427.

Gass, S., Saaty, T. 1955. "The computational algorithm for the parametric objective function." *Naval Research Logistics Quarterly* 2: 39-45.

Geoffrion, A.M., Graves, G.W. 1974. "Multicommodity distribution system design by benders decomposition." *Management Science* 822-844.

Gerrad, R.A., Church, R.L. 1994. "A generalized approach to modeling the hierarchical maximal covering location problem with referral." *Papers in Regional Science* 73: 425-453.

Giarola, S., Zamboni, A., Bezzo, F. 2011. "Spatially explicit multi-objective optimization for design and planning of hybrid first and second generation biorefineries." *Computers & Chemical Engineering.*

Grossman, You &. 2008. "Design of responsive supply chains under demand uncertainty." *Computers & Chemical Engineering.*

Gu, W., Wang, X. 2012. "An interactive geospatial analysis platform for facility location decision-making." *International Journal of Numerical Analysis and Modeling* 9: 217-231.

Guha, S., Meyerson, A., Munagala, K. 2000. "Hierarchical placement and network design problems." *Proceedings of the 41st annual symposium on foundations of computer science* 603-612.

Guillen, G., Mele, F., Bagajewicz, M. Espuna, A., Puigjaner, L. 2005. "Multiobjective supply chain design under uncertainty." *Chemical Engineering Science.*

Gunawan, S., Azaram, S. 2004. "Multi-objective robust optimization using a sensitivity region concept." *Structural and Multidisciplinary Optimization* 50-60.

Gurobi. 2013. "Gurobi Optimizer Version 5.1." Houston, Texas: Gurobi Optimization Inc.

Haimes, Y.Y., Lasdon, L.S., Wismer, D.A. 1971. "On a bicriterion formulation of the problems of integrated system identification and system optimization." *IEEE Transactions on Systems, Man, and Cybernetics* 1: 296-297.

Hakimi, S.L. 1964. "Optimum location of sitchin centers and the absolute centers and medians of a graph." *Operations Research* 450-459.

Hakimi, S.L. 1965. "Optimum location of switching centers in a communication network and some related graph theoretic problems." *Operations Research* 462-475.

Halpern, J. 1980. "Duality in the cent-dian of a graph." *Operations Research* 722-735.

Halpern, J. 1978. "Finding minimal center-median convex combinaiton (cent-dian) of a graph." *Management Science* 534-544.

Halpern, J. 1976. "The location of a center-median convex combination on an undirected tree." *Journal of Regional Science* 237-245.

Hamming, Richard W. 1950. "Error detecting and error correcting codes." *Bell System Technical Journal* 29: 147-160.

Handler, G.Y. 1985. "Medi-centers of a tree." *Transportation Science* 246-260.

Harvey, M.E., Hung, M.S., Brown, J.R. 1974. "The application of a p-median algorithm to the identification of nodal hierarchies and growth centers." *Economic Geography* 50: 187-202.

Hertwin, M., Elias, O., Ruben, T., Luis, M. 2014. "Variations in the flow approach to CFCLP-TC for multiobjective supply chain design." *Mathematical Problems in Engineering.*

Hinojosa, Y., Puerto, J., Fernandez, F.R. 2000. "A multi-period two-echelon multi-commodity capacitated plant location problem." *European Journal of Operational Research* 123: 271-291.

Hiremath, N., Sahu, S., Tiwari, M. 2013. "Multi objective outbound logistics network design for a manufacturing supply chain." *Journal of Intelligent Manufacturing.*

Hodgson, M.J. 1986. "A hierarchical location-allocation model with allocations based on facility size." *Annals of Operations Research* 6: 273-289.

Hodgson, M.J. 1984. "Alternative approaches to hierarchical location-allocation systems." *Geographical Analysis* 16: 275-281.

Hodgson, M.J. 1988. "An hierarchical location-allocation model for primary health care delivery in a developing area." *Social Science and Medicine* 26: 153-161.

Holmes, J., Williams, F., Brown, L. 1972. "Facility location under a maximum travel restriction: and example using day care facilities." *Geographical analysis.*

Hugo, A., Pistikopoulos, E. 2005. "Environmentally conscious long-range planning and design of supply chain networks." *Journal of Cleaner Production.*

Jabal-Ameli, M., Mortezaei, M. 2011. "A hybrid model for multi-objective capacitated facility location network design problem." *International Journal of Industrial Engineering Computations.*

Jayaraman, V., Gupta, R., Pirkul, H. 2003. "Selecting hierarchical facilities in a ervice-operations environment." *European Journal of Operational Research* 147: 613-628.

Johnson, M.P, Gorr, W.L., Roehrig, S. 2005. "Location of service facilities for the elderly." *Annals of Operations Research* 136: 329349.

Kantor, E., Peleg, D. 2006. "Approximate Hierarchical Facility Location and Applications to the Shallow Steiner Tree and Range Assignment Problems." In *CAIC 2006*, by T., Finocchi, I., Italiano, G.F. Calamoneri, 211-222. Berlin Heidelberg: Springer-Verlag.

Khajavi, L., Seyed-Hosseini, S., Makui, A. 2011. "An integrated forward/reverse logistics network optimization model for multi-stage capacitated supply chain." *iBuisiness.*

Kim, J., Moon, I. 2008. "Strategic design of hydrogen infrastructure considering cost and safety using multiobjective optimization." *International Journal of Hydrogen Energy.*

Kim, J., Tcha, D. 1992. "Optimal design of a two-level hierarchical network with tree-star configuration." *Computers and Industrial Engineering* 22: 273-281.

Klose, A., Drexl, A. 2005. "Facility location models for distribution system design." *European Journal of Operational Research* 4-29.

Kohsaka, H. 1983. "A central-place model as a two-level location-allocation system." *Environment and Planning A* 15: 5-14.

Koutra, D., Parikh, A., Ramdas, A., Xiang, J. 2011. "Algorithms for graph similarity and subgraph matching."

Kuehn, A.A., Hamburger, M.J. 1963. "A heuristic program for locating warehouses." *Management Science.*

Lauri, J. 2011. "Subgraphs as a measure of similarity." In *Structural Analysis of Complex Netowks*. Boston.

Lee, C., Ro, H., Tcha, D. 1993. "Topological design of a two-level network with ring-star configuration." *Computers and Operations Research* 20: 625-637.

Lee, J.M., Lee, Y.H. 2010. "Tabu based heuristics for the generalized hierarchical covering location problem." *Computers & Industrial Engineering* 58: 638-645.

Liao, S., Hsieh, C., Lai, P. 2011. "An evolutionary approach for multi-objective optimization of the integrated location-inventory distribution network problem in vendor-managed inventory." *Expert Systems with Applications.*

Lin, L. Gen, M., Wang, X. 2007. "A hybrid genetic algorithm for logistics network design with flexible multistage model." *International Journal of Information Systems for Logistics and Management.*

Litvinchev, I., Lucero, E.L.O. 2012. "Solving the two-stage capacitated facility location problem by the lagrangian heuristic." In *ICCL 2012*, by H. Hu, 92-103. Berling Heidelberg: Springer-Verlag.

Loureiro, M., Pereira, P. 2012. "Portfolios of real options and capacity expansion in transmission network expansion planning."

Makui A., Haerian, L., Eftekhar, M. 2006. "Designing a multi-objective nonlinear cross-dicking location allocation model using genetic algorithm." *Journal of Industrial Engineering International.*

Mandell, M.B. 1998. "Covering models for two-tiered emergency medical services systems." *Location Science* 6: 355-368.

Marianov, V., Serra, D. 2001. "Hierarchical location-allocation models for congested systems." *European Journal of Operational Research* 135: 195-208.

Marin, A., Pelegrin, B. 1999. "Applying lagrangian relaxation to the resolution of two-stage location problems." *Annals of Operations Research* 86: 179-198.

Melachrinoudis, E., Messac, A., Mon, H. 2005. "Consolidating a warehouse network: a physical programming approach." *International Journal of Production Economics.*

Melo, M.T., Nickel, S., Saldanha-da-Gama, F. 2009. "Facility location and supply chain managment - A review." *European Journal of Operational Research* 401-412.

Meyerson, A., Munagala, K., Plotkin, S. 2000. "Cost distance: two metric network design." *Proceedings of the 41st annual symposium on foundations of computer science* 624-630.

Miettinen, K. M. 1999. *Nonlinear Multiobjective Optimization.* Boston: Kluwer Academic Publishers.

Mitropoulos, P., Giannikos, I., Mitropoulos, I. 2009. "Exact and heuristic approaches for the locational planning of an integrated solid waste management system." *Operational Research: An International Journal* 9: 329-347.

Mitropoulos, P., Mitropoulos I., Giannikos, I., Sissouras, A. 2006. "A biopjective model for the locational planning of hospitals and health centers." *Health Care and Management Science* 9: 171-179.

Mo, Y., Harrison, T., Barton, R. 2011. "Solving stochastic programming models in supply chain design using sampling heuristics." *IMA Journal of Management Mathematics* 65-77.

Montoya-Torres, J.R., Aponte, A., Rosas, P. 2011. "Applying GRASP to solve the multi-item three echelon uncapacitated facility location problem." *Journal of Operational Research Society* 62: 397-406.

Moon, I., Chaoudry, S. 1984. "An analysis of network location problems with distance constraints." *Management Science.*

Moore, G.C., ReVelle, C. 1982. "The heirarchical service location problem." *Management Science* 28: 775-780.

Muphy, P.R., Wood, D.F. 2011. *Contemporary Logistics.* Prentice Hall/Pearson.

Narasimhan, S., Pirkal, H. 1992. "Hierarchical concentrator location problem." *Computer Communications* 15: 185-191.

Narula, S. C. 1982. "Heirarchical location-allocation problems: A classification scheme." *European Journal of Operational Research* 15: 93-99.

Nickel, S., Puerto, J. 2005. *Location Theory: A Unified Approach.* New York: Springer.

Okumura, M., Tsukai, M. 2012. "A rank size rule in a firm, produced from a hierarchical branch office location model." *Review of Urban & Regional Development Studes* 22: 73-88.

Olivares-Benitez, E. Gonzalez-Velarde, J., Rios-Mercado, R. 2012. "A supply chain design problem with facility location and bi-objective transportation choices." *TOP.*

Olivares-Benitez, E., Rios-Mercado, R., Gonzalez-Velarde, J. 2013. "A metaheuristic algorithm to solve the selection of transportation channels in supply chain design." *International Journal of Production Economics.*

Pahlavani, A., Mehrabad, M.S. 2011. "A fuzzy multi-objective programming for optimization of hierarchical service centre locations." *Journal of Uncertain Systems* 5: 202-226.

Paksoy, T., Ozceylan, E., Weber, G. 2011. "A multi-objective model for optimization of a green supply chain network." *Global Journal of Technology & Optimization.*

Pareto, V. 1964 (original edition in 1896). *Cours d'Economie Politique.* Geneve.

—. 1971 (original edition in 1927). *Manual of Political Economy.* The MacMillan Press Ltd.

Pati, R., Vrat, P., Kumar, P. 2008. "A goal programming model for paper recycling system." *Omega.*

Pinto-Varela, T., Barbosa-Povoa, A., Novais, A. 2011. "Bi-objective optimization approach to the design and planning of supply chains: economic versus environmental performances." *Computers and Chemical Engineering.*

Pirkul, H., Jayaraman, V. 1998. "A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution." *Computers and Operations Research* 25: 869-878.

Pirkul, H., Jayaraman, V. 1996. "Production, transportation and distribution planning in multi-commodity tri-echelon system." *Transportation Science* 30: 291-302.

Pishvaee, M. Torabi, S. 2010. "A possibilistic programming approach for closed-loop supply chain network design under uncertainty." *Fuzzy Sets and Systems.*

Pishvaee, M., Farahani, R., Dullaeart, W. 2010. "A memetic algorithm for bi-objective integrated forward/reverse logistics network design." *Computers & Operations Research.*

Plaxton, C.G. 2006. "Approximation algorithms for hierarchical location problems." *Journal of Computer and System Sciences* 72: 425-443.

Rad, S., Desa, M., Azari, S. 2014. "Model and solve the bi-criteria multi source flexible multistaage logistics network." *International Journal of Advanced Computer Science and Information Technology.*

Ramezani, M. Bashiri, M., Ravakkoli-Moghaddam, R. 2013. "A new multi-objective stochastic model for a forward/reverse logistic network design with responsiveness and quality level." *Applied Mathematical Modeling.*

Ramudhin, A., Chaabane, A., Paquet, M. 2010. "Carbon market sensitive sustainable supply chain network design." *International Journal of Management Science.*

ReVelle, C., Cohon, J., Shobrys, D. 1981. "Multiple Objectives in Facility Location: A Review." In *Organization: Multiple Agents with Multiple Criteria*, by J.N. (ed.) Morse. Berlin Heidelberg: Springer-Verlag.

ReVelle, C., Swain, R. 1970. "Central facilities location." *Geographical Analysis.*

ReVelle, C.S., Eiselt, H.A. 2005. "Location Analysis: A synthesis and survey." *European Journal of Operational Research* 1-19.

Rezazadeh, M., Farahani, S. 2010. "A multi-objective approach for multi capacity warehouse location within distribution supply chain problem." *Journal of American Scicne.*

Ro, H., Tcha, D. 1984. "A branch and bound algorithm for the two-level uncapacitated facility location problem with some side constraints." *European Journal of Operational Research* 18: 349-358.

Roy, B. 1977. "A Conceptual Framework for a Prescriptive Theory of 'Decision Aid' ." *Management Science.*

Ruefli, T.W., Storbeck J. 1982. "Behaviorally linked location hierarchies." *Environment and Planning B* 9: 121-129.

Sahin, G., Sural, H. 2007. "A Review of Hierarchical Facility Location Models." *Computers & Operations Research.*

Sahin, G., Sural, H., Meral, S. 2007. "Locational Analysis for Regionalization of Turkish Red Crescent." *Computers & Operations Research* 692-704.

Schilling, D., Elzinga, D.J., Cohon, J., Church, R., ReVelle, C. 1979. "The team/fleet models for simultaneous facility and equipment siting." *Transportation Science* 13: 163-175.

Scott, A.J. 1971. "Operational analysis of nodal hierarchies in network systems." *Operations Research Quarterly* 22: 25-36.

Selim, H., Ozkarahan, I. 2008. "A supply chain distribution network design model: an interactive fuzzy goal programming-based solution approach." *International Journal of Advanced Manufacturing Technology.*

Serra, D. 1996. "The coherent covering location problem." *Papers in Regional Science* 79-101.

Serra, D. 1996. "The coherent covering location problem." *Papers in Regional Science* 75: 79-101.

Serra, D., Marianov, V., ReVelle, C. 1992. "The maximum capture hierarchical problem." *European Journal of Operational Research* 62: 363-371.

Serra, D., ReVelle, C. 1993. "The pq-median problem: location and districting of hierarchical facilities." *Location Science* 1: 299-312.

Serra, D., ReVelle, C. 1994. "The pq-median problem: location and districting of hierarchical facilities II. Heuristic solution methods." *Location Science* 2: 63-82.

Shavandi, H., Mahlooji, H. 2008. "Fuzzy hierarchical queueing models for the location set covering problem in congested systems." *Scientia Iranica* 15: 378-388.

Shavandi, H., Mahlooji, H., Eshghi, K., Khanmohammadi, S. 2006. "A fuzzy coherent hierarchical location-allocation model for congested systems." *Scientia Iranica* 13: 14-24.

Shen, Z. 2007. "Integrated supply chain desing models: a survey and future research directions." *Journal of Industrial and Management Optimization* 1-27.

Shen, Z.M., Daskin, M.S. 2005. "Trade-offs Between Customer Service and Cost in Integrated Supply Chain Design." *Manufacturing and Service Operations Management* 188-207.

Shimizu, Y., Matsuda, S., Wada, T. 2006. "A flexible design of logistic network against uncertain demands through hybrid meta-heuristic method." *Computer Aided Chemical Engineering.*

Shmoys, D.B., Tardos, E., Aardal, K. 1997. "Approximation algorithms for facility location problems." *Proceedings of the 29ths annual ACM symposium on theory of computing* 265-274.

Smith, H.K., Harper, P.R., Potts, C.N. 2013. "Bicriteria efficiency/equity hierarchical location models for public service application." *Journal of Operational Research Society* 64: 500-512.

Tcha, D., Lee, B. 1984. "A branch and bound algorithm for the multi-level uncapacitated location problem." *European Journal of Operational Research* 18: 25-43.

Teixeira, J.C., Antunes, A.P. 2008. "Hierarchical location model for public facility planning." *European Journal of Operational Research* 185: 92-104.

Toregas, C., Swain, R., ReVelle, C., Bergman, L. 1971. "The location of emergency service facilities." *Operations Research.*

Trigeorgis, L. 1996. *Real Options: Managerial flexibility and strategy in resource allocation.* Cambridge: The MIT Press.

Van Roy, T.J. 1989. "Multi-level production and distribution planning with transportation fleet optimization." *Management Science* 35: 1443-1453.

Van Veldhuizen, D.A. 2000. "On measuring multiobjective evolutionary algorithms performances." *Congress on Evolutionary Computation.* Piscataway, New Jersey. 205-211.

Vernekar, A., Anandalingam, G., Dorny, C.N. 1990. "Optimization of resource location in hierarchical computer networks." *Computers and Operations Research* 17: 375-388.

Wang, F., Lai, X., Shi, N. 2011. "A multi-objective optimization for green supply chain network design." *Decision Support Systems.*

Wang, T., Lu, T., Zhang, C. 2013. "Integrated logistics network design in hybrid manufacturing/remanufacturing system under low-carabon restriction." *Proceedings of 2nd International Conference on Logistics, Informatics and Service Science.* Berlin Heidelberg.

Watson, M., Lewis, S., Cacioppi, P., Jayaraman, J. 2013. *Supply Chain Network Design: Applying Optimization and Analytics to the Global Supply Chain.* Upper Saddle River, New Jersey: FT Press.

Wirashinghe, S.C., Waters, N.M. 1983. "An approximate procedure for determining the number, capacities and location of solid waste transfer-stations in an urban region." *European Journal of Operational Research* 12: 105-111.

Wu, Q., Wang, S., Jingwei, S. 2011. "The minimum cost of hierarchical location-allocation models for banks in Kunming, China." *IEEE.*

Xu, J. Liu, Q., Wang, R. 2008. "A class of multi-objective supply chain networks optimal model under random fuzzy environment and its application to the industry of Chinese liquor." *Information Sciences.*

Yasenovskiy, V., Hodgson, J. 2007. "Hierarchical Location-Allocation with Spatial Choice Interaction Modeling." *Annals of the Association of American Geographers* 97: 496-511.

Yazdian, S., Shahanaghi, K. 2011. "A multi-objective possibilistic programming approach for locating distribution centers and allocating customers demands in supply chains." *International Journal of Industrial Engineering Computations.*

You, F., Grossman, I. 2011a. "Balancing responsiveness and economics in process supply chain design with multi-echelon stochastic inventory." *American Institute of Chemical Engineers.*

You, F., Grossmand, I. 2008. "Design of responsive supply chains under demand uncertainty." *Computers & Chemical Engineering.*

You, F., Tao, L., Graziano, D., Snyder, S. 2011. "Optimal design of sustainable cellulosic biofuel supply chains: multiobjective optimization coupled with life cycle assessment and input-output analysis." *American Institute of Chemical Engineers.*

Yu, Y., Murphy, S., Murphy, L. 2010. "Planning base station and relay station locations for IEEE802.16j network with capacity constraints." *IEEE Communications Society.*

Zhang, J., Ye, Y. 2002. "A note on the maximization version of the multi-level facility location problem." *Operations Research Letters* 30: 333-335.

Appendix

1. Data File

```
# Updated: 10/2/2014
# Created by Jeremy North
########################################################################
# Daskin_Data.jl
# Dissertation Dataset 1
# Daskin 1995
########################################################################

# Define containers
points, dist = Dict{Int32, Array}(), Dict{Tuple, Float64}()
coverage, non_coverage = Dict{Int32, Array}(), Dict{Int32, Array}()

# points dictionary. [location name,demand(10000s),lat,lon]
points[1] = ["CA",297.60021,115800,38.56685,-121.46736]
points[2] = ["NY",179.90455,101800,42.66575,-73.799017]
points[3] = ["TX",169.86510,72600,30.30588,-97.750522]
points[4] = ["FL",129.37926,72400,30.457,-84.281399]
points[5] = ["PA",118.81643,38400,40.27605,-76.884503]
points[6] = ["IL",114.30602,59200,39.781433,-89.644654]
points[7] = ["OH",108.47115,66000,39.988933,-82.987381]
points[8] = ["MI",92.95297,48400,42.7091,-84.553996]
points[9] = ["NJ",77.30188,71300,40.2234,-74.764224]
points[10] = ["NC",66.28637,96600,35.82195,-78.658753]
points[11] = ["GA",64.78216,71200,33.7629,-84.422592]
points[12] = ["VA",61.87358,66600,37.53105,-77.474584]
points[13] = ["MA",60.16425,161400,42.336029,-71.017892]
points[14] = ["IN",55.44159,60800,39.7764,-86.146196]
points[15] = ["MO",51.17073,61500,38.571902,-92.190459]
points[16] = ["WI",48.91769,75200,43.0798,-89.387519]
points[17] = ["TN",48.77185,74400,36.17155,-86.784829]
points[18] = ["WA",48.66692,77800,47.041917,-122.893766]
points[19] = ["MD",47.81468,138500,38.97165,-76.503033]
points[20] = ["MN",43.75099,70900,44.947744,-93.103686]
points[21] = ["LA",42.19973,67900,30.448967,-91.126043]
points[22] = ["AL",40.40587,62200,32.3544,-86.284287]
points[23] = ["KY",36.85296,61500,38.19077,-84.865203]
points[24] = ["AZ",36.65228,77100,33.54255,-112.071399]
points[25] = ["SC",34.86703,72600,34.039236,-80.886341]
points[26] = ["CO",32.94394,79000,39.768035,-104.872655]
points[27] = ["CT",32.87116,133800,41.7657,-72.683866]
points[28] = ["OK",31.45585,54900,35.46705,-97.513491]
```



```
points[29] = ["OR",28.42321,60300,44.9245,-123.022057]
points[30] = ["IA",27.76755,49500,41.576738,-93.617405]
points[31] = ["MS",25.73216,54600,32.3205,-90.207591]
points[32] = ["KS",24.77574,48800,39.0379,-95.691999]
points[33] = ["AR",23.50725,64200,34.7224,-92.354076]
points[34] = ["WV",17.93477,66100,38.35055,-81.630439]
points[35] = ["UT",17.22850,67200,40.777267,-111.929921]
points[36] = ["NE",15.78385,61700,40.8164,-96.688171]
points[37] = ["NM",15.15069,99000,35.678502,-105.954149]
points[38] = ["ME",12.27928,79500,44.330647,-69.729714]
points[39] = ["NV",12.01833,99300,39.148328,-119.743243]
points[40] = ["NH",11.09252,112400,43.231594,-71.560077]
points[41] = ["ID",10.06749,67700,43.606651,-116.2261]
points[42] = ["RI",10.03464,113000,41.82195,-71.419732]
points[43] = ["MT",7.99065,63200,46.596522,-112.020381]
points[44] = ["SD",6.96004,59500,44.372982,-100.322483]
points[45] = ["DE",6.66168,88700,39.158691,-75.517441]
points[46] = ["ND",6.38800,67900,46.805467,-100.767298]
points[47] = ["DC",6.06900,123900,38.90505,-77.016167]
points[48] = ["VT",5.62758,94100,44.266482,-72.571854]
points[49] = ["WY",4.53588,68700,41.14545,-104.792349]

# Calculate big circle distance between two points
function haversine(point1, point2)
  point1[1], point2[1] = pi * point1[1]/180, pi * point2[1]/180
  point1[2], point2[2] = pi * point1[2]/180, pi * point2[2]/180
  dlon, dlat = point2[2] - point1[2], point2[1] - point1[1]
  a = (sin(dlat/2))^2 + cos(point1[1]) * cos(point2[1]) * (sin(dlon/2))^2
  c = 2 * atan2(sqrt(a), sqrt(1-a))
  return(3961*c)        # 3961 = radius of Earth in miles
end

# Create distance matrix
function distanceMatrix()
  for i in 1:49
    for j in i+1:49
      dist[(i,j)] = haversine(points[i][4:5], points[j][4:5])
      dist[(j,i)] = dist[(i,j)]
    end
    dist[(i,i)] = 0
  end
end


# Create coverage matrix (Aij)
function coverageMatrix()
```



```
  for i in 1:49
    coverage[i] = zeros(49)
    non_coverage[i] = zeros(49)
    for j in 1:49
      if dist[(i,j)] <= minimumCoverage
        coverage[i][j] = 1
      else non_coverage[i][j] = 1
      end
    end
  end
end

bigM, minimumCoverage, numNodes = 2470.51601, 500, length(points)
distanceMatrix(), coverageMatrix()
```


2. Solution Display File

```
# Updated: 10/19/2014
# Created by Jeremy North
########################################################################
# Reads solution, displays selected locations and solution to user
########################################################################

include("Daskin_Data.jl")
#include("DissertationModel1.jl")
#include("DissertationModel2.jl")

function getLocations(answerJ, answerK)
  ansLocsW = Dict()
  ansLocsP = Dict()
  numNodes = length(answerJ)
  for i in 1:numNodes
    ansLocsW[i] = String[]
    for j in 1:length(answerJ[i])
      if answerJ[i][j] == 1
        push!(ansLocsW[i], locations[j])
      end
    end
  end
  for i in 1:numNodes
    ansLocsP[i] = String[]
    for j in 1:length(answerK[i])
      if answerK[i][j] == 1
        push!(ansLocsP[i], locations[j])
      end
    end
  end
  for i in 1:numNodes
    if ansLocsW[i] != ""
      println(i, " warehouses ", ansLocsW[i])
    end
    if ansLocsP[i] != ""
      println(i, " plants ", ansLocsP[i])
    end
  end
end

function getMOGLocations(answerJ, answerK)
  for i in 1:length(answerJ)
    if answerJ[i] == 1
      println(i, " warehouses ", points[i][1])
```



```
    end
  end
  for i in 1:length(answerK)
    if answerK[i] == 1
      println(i, " plants ", points[i][1])
    end
  end
end
```


3. Elastic Constraint Model: Single Flow

```
# Updated: 10/6/2014
# Created by Jeremy North
#######################################################################
# Single-Flow, P-Warehouse, Q-Plant location problem
# Multi-Criteria: Min Distance, Max Service (coverage)
# Application of Ehrgott (2006) elastic constraint methodology
#######################################################################

## MODULES & PACKAGES REQUIRED ##
using JuMP
using Gurobi
include("Daskin_Data.jl")
include("DissertationSolutionDataCollection.jl")

## DATA ##
customerI = length(points)            # set cardinality
warehouseJ, plantK = customerI, customerI
unCovered = bigM                      # initialize for cost min
P, Q = 1, 1                           # p warehouses q plants
penalty = 100000000000000             # slack & surplus penalty (very big number)

# solution containers and bookkeeping parameters
solutionNum = 0
runTimes, total_cost, total_coverage, keys = Float64[], Float64[], Float64[], Int[]
answerJ, answerK, answerX1, answerX2 = Dict(), Dict(), Dict(), Dict()
answerC, solutions = Dict(), Dict()

## MODEL ##
function hierarchicalMCTC(P, Q, requiredCover)

  tic()
  # Create Model #
  m = Model(solver=GurobiSolver())

  # Decision Variables #
  @defVar(m, Y1[1:customerI,1:warehouseJ] >= 0)       # flow w to c
  @defVar(m, Y2[1:warehouseJ,1:plantK] >= 0)          # flow p to w
  @defVar(m, X1[1:warehouseJ], Bin)                   # warehouse selection
  @defVar(m, X2[1:plantK], Bin)                       # plant selection
  @defVar(m, C[1:customerI], Bin)                     # customer coverage

  # Elastic Constraint Variables #
  @defVar(m, L >= 0)                                  # slack
  @defVar(m, S >= 0)                                  # surplus
```

```
# Solution Tracking Variables #
@defVar(m, totalCost >= 0)
@defVar(m, totalCoverage >= 0)

# Objective: min total cost + penalty*slack #
@setObjective(m, Min, sum{dist[(i,j)]*Y1[i,j], i = 1:customerI, j = 1:warehouseJ} +
                      sum{dist[(j,k)]*Y2[j,k], j = 1:warehouseJ, k = 1:plantK} +
                      penalty*S + penalty*L)

# Constraints #
# coverage
for i in 1:customerI
  for j in 1:warehouseJ
    @addConstraint(m, coverage[i][j]*X1[j] <= C[i])
  end
end

for i in 1:customerI
    @addConstraint(m, points[i][2]*C[i] <= sum{coverage[i][j]*Y1[i,j],
                                                                j=1:warehouseJ})
end

# linking X1 with Y
for j in 1:warehouseJ
  for i in 1:customerI
    @addConstraint(m, Y1[i,j] <= points[i][2]*X1[j])
  end
end

# linking X2 with Z
for k in 1:plantK
    @addConstraint(m, sum{Y2[j,k], j=1:warehouseJ} <= bigM*X2[k])
end

# demand
for i in 1:customerI
  @addConstraint(m, sum{Y1[i,j], j=1:warehouseJ} == points[i][2])
end

# flow balance
for j in 1:warehouseJ
  @addConstraint(m, sum{Y1[i,j], i=1:customerI} == sum{Y2[j,k], k=1:plantK})
end

# P Warehouses, Q Plants
```



```
  @addConstraint(m, sum{X1[j], j in 1:warehouseJ} == P)
  @addConstraint(m, sum{X2[k], k in 1:plantK} == Q)

  # elastic constraint (Covering Objective)
  @addConstraint(m, sum{points[i][2]*C[i], i=1:customerI} + L - S >=
                                                      requiredCover)

  # book keeping
  @addConstraint(m, totalCost == sum{dist[(i,j)]*Y1[i,j], i = 1:customerI, j =
        1:warehouseJ} + sum{dist[(j,k)]*Y2[j,k], j = 1:warehouseJ, k = 1:plantK})
  @addConstraint(m, totalCoverage == sum{points[i][2]*C[i], i = 1:customerI})

  # Solve (Gurobi default) #
  status = solve(m)
  # error checking
  if status == :Infeasible
    error("Model is infeasible!")
  end

  # Print results #
  println("Best objective: $(round(getObjectiveValue(m)))")

return toc(), int(getValue(totalCost)), getValue(totalCoverage), getValue(X1),
                  getValue(X2), getValue(Y1), getValue(Y2), getValue(C)
end

# Pareto Frontier Generation #
function generateSolutions(P, Q)
  requiredCover = 0

  # find max coverage
  runTime, runCost, maxCover, selectedJ, selectedK, Y1, Y2, C =
                                          hierarchicalMCTC(P, Q, bigM)

  # find rest of PO solutions, starting with cost min
  while requiredCover <= maxCover
    requiredCover += 1

   runTime, runCost, runCover, selectedJ, selectedK, Y1, Y2, C =
   hierarchicalMCTC(P, Q,     requiredCover)

    if [runCost, runCover] in values(solutions) == false    #
      global solutionNum += 1
      push!(keys, solutionNum)
      push!(runTimes, runTime)
      push!(total_coverage, runCover)
```

```
    push!(total_cost, runCost)
    answerJ[solutionNum] = selectedJ
    answerK[solutionNum] = selectedK
    answerX1[solutionNum] = Y1
    answerX2[solutionNum] = Y2
    answerC[solutionNum] = C
    solutions[solutionNum] = [runCost, runCover]
    global unCovered = bigM - runCover
    requiredCover = bigM - unCovered
   end
  end
end

generateSolutions(5,1)
```


4. Elastic Constraint Model: Multi-Flow

```
# Updated: 10/6/2014
# Created by Jeremy North
##############################################################################
# Multi-Flow, P-Warehouse, Q-Plant location problem
# Multi-Criteria: Min Distance, Max Service (coverage)
# Application of Ehrgott (2006) elastic constraint methodology
##############################################################################

## MODULES & PACKAGES REQUIRED ##
using JuMP, Gurobi
include("Daskin_Data.jl")

## DATA ##
customerI = length(points)             # set cardinality
warehouseJ, plantK = customerI, customerI
unCovered = bigM                       # initialize for cost min
P, Q = 1, 1                            # p warehouses q plants
penalty = 100000000000000              # slack & surplus penalty

# solution containers and bookkeeping parameters
solutionNum = 0
runTimes, total_cost, total_coverage, keys = Float64[], Float64[], Float64[], Int[]
answerJ, answerK, answerX1, answerX2 = Dict(), Dict(), Dict(), Dict()
answerX3, answerC, solutions = Dict(), Dict(), Dict()

## MODEL ##
function hierarchicalMCTC(P, Q, requiredCover)

  tic()
  # Create Model #
  m = Model(solver=GurobiSolver())

  # Decision Variables #
  @defVar(m, Y1[1:customerI,1:warehouseJ] >= 0)      # flow from w to c
  @defVar(m, Y2[1:warehouseJ,1:plantK] >= 0)         # flow from p to w
  @defVar(m, Y3[1:customerI,1:plantK] >= 0)          # flow from p to c
  @defVar(m, X1[1:warehouseJ], Bin)                  # warehouse selection
  @defVar(m, X2[1:plantK], Bin)                      # plant selection
  @defVar(m, C[1:customerI], Bin)                    # customer coverage

  # Elastic Constraint Variables #
  @defVar(m, L >= 0)                                 # slack
  @defVar(m, S >= 0)                                 # surplus
```

```
# Solution Tracking Variables #
@defVar(m, totalCost >= 0)
@defVar(m, totalCoverage >= 0)


# Objective: min total cost + penalty*slack #
@setObjective(m, Min, sum{dist[(i,j)]*Y1[i,j], i = 1:customerI, j = 1:warehouseJ} +
                sum{dist[(j,k)]*Y2[j,k], j = 1:warehouseJ, k = 1:plantK} +
                sum{dist[(i,k)]*Y3[i,k], i = 1:customerI, k = 1:plantK} +
                penalty*S + penalty*L)

# Constraints #
# coverage
for i in 1:customerI
  for j in 1:warehouseJ
    @addConstraint(m, coverage[i][j]*X1[j] <= C[i])
  end
  for k in 1:plantK
    @addConstraint(m, coverage[i][k]*X2[k] <= C[i])
  end
end

for i in 1:customerI
    @addConstraint(m, points[i][2]*C[i] <= sum{coverage[i][j]*Y1[i,j],
                j=1:warehouseJ} + sum{coverage[i][k]*Y3[i,k], k=1:plantK})
end

# linking X1 with Y
for i in 1:customerI
  for j in 1:warehouseJ
    @addConstraint(m, Y1[i,j] <= points[i][2]*X1[j])
  end
  for k in 1:plantK
    @addConstraint(m, Y3[i,k] <= points[i][2]*X2[k])
  end
end

# linking X2 with Z
for k in 1:plantK
    @addConstraint(m, sum{Y2[j,k], j=1:warehouseJ} <= bigM*X2[k])
end

# demand
for i in 1:customerI
  @addConstraint(m, sum{Y1[i,j], j=1:warehouseJ} + sum{Y3[i,k], k=1:plantK} ==
                                                                    points[i][2])
```



```
  end

  # flow balance
  for j in 1:warehouseJ
    @addConstraint(m, sum{Y1[i,j], i=1:customerI} == sum{Y2[j,k], k=1:plantK})
  end

  # P Warehouses, Q Plants
  @addConstraint(m, sum{X1[j], j in 1:warehouseJ} == P)
  @addConstraint(m, sum{X2[k], k in 1:plantK} == Q)

  # elastic constraint (Covering Objective)
  @addConstraint(m, sum{points[i][2]*C[i], i=1:customerI} + L - S >=
                                                        requiredCover)

  # book keeping
  @addConstraint(m, totalCost == sum{dist[(i,j)]*Y1[i,j], i = 1:customerI, j =
        1:warehouseJ} + sum{dist[(j,k)]*Y2[j,k], j = 1:warehouseJ, k = 1:plantK} +
                        sum{dist[(i,k)]*Y3[i,k], i = 1:customerI, k = 1:plantK})
  @addConstraint(m, totalCoverage == sum{points[i][2]*C[i], i = 1:customerI})

  # Solve (Gurobi default) #
  status = solve(m)
  # error checking
  if status == :Infeasible
    error("Model is infeasible!")
  end

  # Print results #
  println("Best objective: $(round(getObjectiveValue(m)))")

  return toc(), int(getValue(totalCost)), getValue(totalCoverage), getValue(X1),
          getValue(X2), getValue(Y1), getValue(Y2), getValue(Y3), getValue(C)
end

# Pareto Frontier Generation #
function generateSolutions(P, Q)
  requiredCover = 0

  # find worst coverage
  runTime, runCost, maxCover, selectedJ, selectedK, Y1, Y2, Y3, C =
                                          hierarchicalMCTC(P, Q, bigM)

  # find rest of PO solutions, starting with cost min
  while requiredCover < maxCover
    requiredCover += 1
```

```
    runTime, runCost, runCover, selectedJ, selectedK, Y1, Y2, Y3, C =
                                         hierarchicalMCTC(P, Q,    requiredCover)

    if [runCost, runCover] in values(solutions) == false    #
      global solutionNum += 1
      push!(keys, solutionNum)
      push!(runTimes, runTime)
      push!(total_coverage, runCover)
      push!(total_cost, runCost)
      answerJ[solutionNum] = selectedJ
      answerK[solutionNum] = selectedK
      answerX1[solutionNum] = Y1
      answerX2[solutionNum] = Y2
      answerX3[solutionNum] = Y3
      answerC[solutionNum] = C
      solutions[solutionNum] = [runCost, runCover]
      global unCovered = bigM - runCover
      requiredCover = bigM - unCovered
    end
  end
end

generateSolutions(3,2)
```


5. Assign Demands and Evaluate Performance for Heuristic

```
# Updated: 10/19/2014
# Created by Jeremy North
########################################################################
# Assigns demands to selected facilities with mandatory service restrictions
# Multi-Criteria: Min Distance, Max Service (coverage)
# Evaluate performance of heuristics (distance, coverage)
########################################################################

## MODULES & PACKAGES REQUIRED ##
include("Daskin_Data.jl")

# finds demand assignment with manditory service restrictions, single flow
function sfWeightedDistance(selectedWarehouses, selectedPlants)
  x, y, covered = selectedWarehouses, selectedPlants, zeros(49)
  z, shorts, assignments, numNodes = zeros(49), zeros(49), zeros(49), length(points)
  if sum(x) == 0
    x = y
  end
  if sum(y) == 0
    y = x
  end
  for i in 1:numNodes
    short = 100000000000
    for j in 1:numNodes
      if x[j] == 1
        if dist[(i,j)] <= 500
          covered[i] = points[i][2]
          for k in 1:numNodes
            if y[k] == 1
              temp = (dist[(i,j)] + dist[(j,k)]) * points[i][2]
              if short > temp
                short = temp
                assignments[i] = j
                shorts[i] = short
              end
            end
          end
        end
      end
    end
    if assignments[i] == 0
      tempCheck = 100000000000
      for k in 1:numNodes
        if y[k] == 1
```

```
      temp = dist[(i,k)] * points[i][2]
       if tempCheck > temp
         short = temp
         tempCheck = temp
         assignments[i] = k
         shorts[i] = short
       end
     end
    end
  end
  z[assignments[i]] += points[i][2]
 end
 return(shorts, z, assignments, covered)
end

# finds demand assignment with manditory service restrictions, multi-flow
function mfWeightedDistance(selectedWarehouses, selectedPlants)
 z, shorts, assignments, covered = zeros(49), zeros(49), zeros(49), zeros(49)
 x, y, combined, numNodes = selectedWarehouses, selectedPlants,
selectedWarehouses, length(points)
 for i in 1:numNodes
  if selectedPlants[i] == 1
   combined[i] = 1
  end
 end
 for i in 1:numNodes
  short = 100000000000
  for j in 1:numNodes
   if combined[j] == 1
    if dist[(i,j)] <= 500
     covered[i] = points[i][2]
     for k in 1:numNodes
      if y[k] == 1
       temp = (dist[(i,j)] + dist[(j,k)]) * points[i][2]
       if short > temp
        short = temp
        assignments[i] = j
        shorts[i] = short
       end
      end
     end
    end
   end
  end
  if assignments[i] == 0
   tempCheck = 100000000000
```



```
    for k in 1:numNodes
      if y[k] == 1
      temp = dist[(i,k)] * points[i][2]
        if tempCheck > temp
          short = temp
          tempCheck = temp
          assignments[i] = k
          shorts[i] = short
        end
      end
     end
   end
   z[assignments[i]] += points[i][2]
  end
  return(shorts, z, assignments, covered)
end

function solutionPerformance(selectedWarehouses, selectedPlants, flowType)
  assignments = zeros(49)
  if flowType == 1
    totalToCust, demandAtWarehouses, assignments, totalCovered  =
sfWeightedDistance(copy(selectedWarehouses), copy(selectedPlants))
  else
    totalToCust, demandAtWarehouses, assignments, totalCovered  =
mfWeightedDistance(copy(selectedWarehouses), copy(selectedPlants))
  end
  totalDistance, totalUncoverage = sum(totalToCust)/bigM, bigM -
sum(totalCovered)
  return(totalDistance, totalUncoverage, assignments)
end
```

6. Multi-Objective GRASP

```
# Updated: 10/17/2014
# Created by Jeremy North
########################################################################
# Single-Flow & Multi-Flow, P-Warehouse, Q-Plant location problem
# Multi-Criteria: Min Distance, Max Service (coverage)
# Multi-Objective Greedy Random Adaptive Search Procedures (MOG)
########################################################################

## MODULES & PACKAGES REQUIRED ##
include("Daskin_Data.jl")
include("Assign_Demands.jl")
include("DissertationSolutionDataCollection.jl")
include("Local_Search_Heuristic.jl")
#using Assign_Demands

## DATA ##
customerI = length(points)              # set cardinality
warehouseJ, plantK = customerI, customerI
unCovered = bigM                        # initialize for cost min
P, Q = 1, 1                             # p warehouses q plants
penalty = 100000000000000               # slack & surplus penalty

# solution containers and bookeeping parameters
solutionNum, numNodes, flowType = 0, length(points), 1
runTimes, total_cost, total_coverage, keys = Float64[], Float64[], Float64[], Int[]
MOGanswerJ, MOGanswerK, answerX1, answerX2 = Dict(), Dict(), Dict(), Dict()
answerC, MOGsolutions = Dict(), Dict()

function constructionPhaseSF(P, Q, alpha, requiredUncoverage, flowType, weight)
 x, y, z, oneFacBestCover = zeros(49), zeros(49), [], 1427.1979399999998
 selectedJ, selectedK, chosenJ, chosenK = zeros(49), zeros(49), zeros(49), zeros(49)
 randomSelection, RCLnum, zMaxD, zMinD, zMaxC, zMinC = [], 0, 0, 0, 0, 0

 for i in 1:numNodes
  temp = zeros(49)
  temp[i] = 1
  x[i], y[i], z = solutionPerformance(zeros(49), temp, flowType)
 end
 RCL = Dict()
 zMaxD, zMinD, zMaxC, zMinC = maximum(x), minimum(x), maximum(y),
minimum(y)
 zMaxD2, zMaxC2 = copy(zMaxD), copy(zMaxC)

 function normalizePerformance(x,y,zMaxD,zMinD,zMaxC,zMinC)
```

```
    x1, y1, z1 = copy(x), copy(y), zeros(49)
    for i in 1:numNodes
      x1[i] = (zMaxD - x[i]) / (zMaxD - zMinD)
    end
    for i in 1:numNodes
      y1[i] = (zMaxC - y[i]) / (zMaxC - zMinC)
    end
    for i in 1:numNodes
      z1[i] = weight * x1[i] + (1-weight) * y1[i]
    end
    return(x1,y1,z1)
  end
  zMaxD, zMinD, zMaxC, zMinC = maximum(x), minimum(x), maximum(y),
minimum(y)
  x,y,weightedXYNorm = normalizePerformance(x,y,zMaxD,zMinD,zMaxC,zMinC)
  zMaxD, zMinD, zMaxC, zMinC = maximum(x), minimum(x), maximum(y),
minimum(y)
  zMaxWeighted, zMinWeighted = maximum(weightedXYNorm),
minimum(weightedXYNorm)
  minPerformanceD, minPerformanceC = zMaxD - alpha * (zMaxD - zMinD),
zMaxC - alpha * (zMaxC - zMinC)
  minPerformanceWeighted = zMaxWeighted - alpha * (zMaxWeighted -
zMinWeighted)

  # first RCL list creation & solution selection, plants
  for i in 1:numNodes
    if weightedXYNorm[i] >= minPerformanceWeighted
      RCLnum += 1
      selectedK = zeros(49)
      selectedK[i] = 1
      #print(sum(selectedK))
      RCL[RCLnum] = selectedK
    end
  end
  chosenK = RCL[rand(1:RCLnum)]
  # given first plant selection, keep going till all chosen
  Q -= 1
  while Q > 0
    Q -= 1
    RCL, RCLnum, x, y = Dict(), 0, zeros(49), zeros(49)
    for i in 1:numNodes
      temp = copy(chosenK)
      if temp[i] == 1
        x[i], y[i] = zMaxD2, zMaxC2
      end
      if temp[i] == 0
```



```
      temp[i] = 1
      x[i], y[i], z = solutionPerformance(zeros(49), temp, flowType)
    end
  end
  zMaxD, zMinD, zMaxC, zMinC = maximum(x), minimum(x), maximum(y),
minimum(y)
  x,y,weightedXYNorm =
normalizePerformance(x,y,zMaxD,zMinD,zMaxC,zMinC)
  zMaxD, zMinD, zMaxC, zMinC = maximum(x), minimum(x), maximum(y),
minimum(y)
  zMaxWeighted, zMinWeighted = maximum(weightedXYNorm),
minimum(weightedXYNorm)
  #minPerformanceD, minPerformanceC = zMaxD - alpha * (zMaxD - zMinD),
zMaxC - alpha *   (zMaxC - zMinC)
  minPerformanceWeighted = zMaxWeighted - alpha * (zMaxWeighted -
zMinWeighted)

  # first RCL list creation & solution selection, warehouses
  for i in 1:numNodes
    if weightedXYNorm[i] >= minPerformanceWeighted
      RCLnum += 1
      selectedK = copy(chosenK)
      selectedK[i] = 1
      RCL[RCLnum] = selectedK
    end
  end
  chosenK = RCL[rand(1:RCLnum)]     # random selction
 end
 bestK = chosenK                    # outcome for chosen plants
  # choose P warehouses
 while P > 0
  P -= 1
  RCL, RCLnum, x, y = Dict(), 0, zeros(49), zeros(49)
  for i in 1:numNodes
    temp = copy(chosenJ)
    if temp[i] == 1
      x[i], y[i] = zMaxD2, zMaxC2
    end
    if temp[i] == 0
      temp[i] = 1
      x[i], y[i], z = solutionPerformance(temp, chosenK, flowType)
    end
  end
  zMaxD, zMinD, zMaxC, zMinC = maximum(x), minimum(x), maximum(y),
minimum(y)
```



```
    x,y,weightedXYNorm =
normalizePerformance(x,y,zMaxD,zMinD,zMaxC,zMinC)
    zMaxD, zMinD, zMaxC, zMinC = maximum(x), minimum(x), maximum(y),
minimum(y)
    zMaxWeighted, zMinWeighted = maximum(weightedXYNorm),
minimum(weightedXYNorm)
    #minPerformanceD, minPerformanceC = zMaxD - alpha * (zMaxD - zMinD),
zMaxC - alpha * (zMaxC - zMinC)
    minPerformanceWeighted = zMaxWeighted - alpha * (zMaxWeighted -
zMinWeighted)

    # first RCL list creation & solution selection
    for i in 1:numNodes
      if weightedXYNorm[i] >= minPerformanceWeighted
        RCLnum += 1
        selectedJ = copy(chosenJ)
        selectedJ[i] = 1
        RCL[RCLnum] = selectedJ
      end
    end
    chosenJ = RCL[rand(1:RCLnum)]     # random selction
  end
  bestJ = chosenJ                   # outcome for chosen warehouses
  return bestJ, bestK
end

# Construct frontier using GRASP
function buildFrontier(P,Q,alpha,requiredCover,flowType,maxIterations, weight)
  MOGTimeStart = tic()

  finalJ, finalK, finalSolution = [], [], [1000000000000000,bigM]
  while maxIterations > 0

    answerDistance, answerCoverage, assigns = 0, 0, []
    selectedWarehouses, selectedPlants = [], []
    weight = weight

    selectedWarehouses, selectedPlants =
constructionPhaseSF(P,Q,alpha,requiredCover,flowType, weight)

    tempJ, tempK = copy(selectedWarehouses), copy(selectedPlants)
    answerDistance, answerCoverage, assigns = solutionPerformance(tempJ, tempK,
flowType)
    if answerDistance <= finalSolution[1] && answerCoverage < finalSolution[2] ||
       answerCoverage <= finalSolution[2] && answerDistance < finalSolution[1]
```



```
    finalJ, finalK, finalSolution = selectedWarehouses, selectedPlants,
[answerDistance, answerCoverage]
    else
      maxIterations -= 1
    end
  end
  MOGTimeEnd = toc()
  println(finalSolution)
  return finalJ, finalK, finalSolution, (MOGTimeEnd-MOGTimeStart) /
maxIterations
end

weight = 1
solutionNum = 1
while weight >= 0
  finalJ, finalK, finalSolution = buildFrontier(3, 2, .2, bigM, 0, 10, weight)
  MOGanswerJ[solutionNum] = finalJ
  MOGanswerK[solutionNum] = finalK
  MOGsolutions[solutionNum] = finalSolution
  solutionNum += 1
  weight -= .01
end

MOGSolutionSet = Set()
for i in 1:length(MOGsolutions)
  push!(MOGSolutionSet, MOGsolutions[i])
end

for i in 1:length(MOGsolutions)
  println(MOGsolutions[i])
  getMOGLocations(MOGanswerJ[i], MOGanswerK[i])
end

println(MOGSolutionSet)

function Pareto_Filter()
  MOGsolutionsTemp = copy(MOGsolutions)
  for i in 1:length(MOGsolutions)-1
    for j in i+1:length(MOGsolutions)
      MOGsolutions[i][1] <= MOGsolutions[j][1] && MOGsolutions[i][2] <
MOGsolutions[j][2] ||
      MOGsolutions[i][2] <= MOGsolutions[j][2] && MOGsolutions[i][1] <
MOGsolutions[j][1]
      delete!(MOGanswerJ, j)
      delete!(MOGanswerK, j)
      delete!(MOGsolutionsTemp, j)
```



```
    end
  end
  MOGsolutions = MOGsolutionsTemp
end
```

7. Local Search: Facility Switch with Tabu Mechanism

```
# Updated: 10/26/2014
# Created by Jeremy North
##########################################################################
# Single-Flow & Multi-Flow, P-Warehouse, Q-Plant location problem
# Multi-Criteria: Min Distance, Max Service (coverage)
# Local Search: Facility switch with tabu
##########################################################################

function GRASP_LocalSearch(selectedWarehouses, selectedPlants, callTracker,
flowType, weight)
  tempJ, tempK, = copy(selectedWarehouses), copy(selectedPlants)
  numNodes, numLocsJ, numLocsK = 49, sum(tempJ), sum(tempK)
  solutionD, solutionC = solutionPerformance(selectedWarehouses, selectedPlants,
flowType)
  finalJ, finalK = copy(selectedWarehouses), copy(selectedPlants)

  function updateTabu(i)
    for j in 1:numNodes
      if haskey(tabuDict, j) == true
        if tabuDict[j] > 10
          delete!(tabuDict, j)
        else
          tabuDict[j] += 1
        end
      end
    end
    if haskey(tabuDict, i) == false
     tabuDict[i] = 1
    end
  end

  function updateWarehouses()
    x, y, z, selectedLocs = zeros(49), zeros(49), zeros(49), Int32[]
    for i in 1:numNodes
      if tempJ[i] == 1
          push!(selectedLocs, i)
      end
    end
    while selectedLocs != []
      x, y = zeros(49), zeros(49)
      tempJ = copy(selectedWarehouses)
      node = pop!(selectedLocs)
      for i in 1:49
        tempJ = copy(selectedWarehouses)
```



```
    x[node], y[node] = bigM, bigM
    tempJ[node] = 0

    tempJ[i] = 1
    if tempJ in values(MOGanswerJ) && finalK in values(MOGanswerJ)
      x[i], y[i] = bigM, bigM
    else
      x[i], y[i] = solutionPerformance(tempJ, finalK, flowType)
    end
    if weight*x[i] + (1-weight)*y[i] < weight*solutionD + (1-weight)*solutionC &&
        haskey(tabuDict, i) == false

      if [x[i], y[i]] in values(MOGsolutions) == false
        solutionD, solutionC = x[i], y[i]
        finalJ = copy(tempJ)
      end
    end
    updateTabu(i)
  end
 end
end

function updatePlants()
 x, y, z, selectedLocs = zeros(49), zeros(49), zeros(49), Int32[]

 for i in 1:numNodes
  if tempK[i] == 1
     push!(selectedLocs, i)
  end
 end

 while selectedLocs != []
  x, y = zeros(49), zeros(49)
  tempK = copy(selectedPlants)
  node = pop!(selectedLocs)
  for i in 1:49
    tempK = copy(selectedPlants)
    x[node], y[node] = bigM, bigM
    tempK[node] = 0

    tempK[i] = 1
    if tempK in values(MOGanswerK) && finalJ in values(MOGanswerJ)
      x[i], y[i] = bigM, bigM
    else
      x[i], y[i] = solutionPerformance(finalJ, tempK, flowType)
    end
```

```
    if weight*x[i] + (1-weight)*y[i] < weight*solutionD + (1-weight)*solutionC &&
     haskey(tabuDict, i) == false
     if [x[i], y[i]] in values(MOGsolutions) == false
      solutionD, solutionC = x[i], y[i]
      finalK = copy(tempK)
     end
    end
    updateTabu(i)
   end
  end
 end

 if callTracker == 0
  updatePlants()
  updateWarehouses()
 else
  updateWarehouses()
  updatePlants()
 end
 return finalJ, finalK
end

tabuDict = Dict()
function Call_LocalSearch(selectedWarehouses, selectedPlants, callTracker, weight)
 loopTemp1, loopTemp2 = [],[]
 while loopTemp1 != selectedWarehouses || loopTemp2 != selectedPlants
  loopTemp1, loopTemp2 = copy(selectedWarehouses), copy(selectedPlants)
  answerDistance, answerCoverage = solutionPerformance(selectedWarehouses,
selectedPlants, flowType)

   if callTracker == 0
    selectedWarehouses, selectedPlants = GRASP_LocalSearch(selectedWarehouses,
selectedPlants, callTracker, flowType, weight)
    callTracker = 1
    selectedWarehouses, selectedPlants = GRASP_LocalSearch(selectedWarehouses,
selectedPlants, callTracker, flowType, weight)
   else
    selectedWarehouses, selectedPlants = GRASP_LocalSearch(selectedWarehouses,
selectedPlants, callTracker, flowType, weight)
    callTracker = 0
    selectedWarehouses, selectedPlants = GRASP_LocalSearch(selectedWarehouses,
selectedPlants, callTracker, flowType, weight)
   end
 end
 return selectedWarehouses, selectedPlants
end
```