2004

Scenario-based reliability facility location models

Lori Shuler
Lehigh University

Follow this and additional works at: http://preserve.lehigh.edu/etd

Recommended Citation

This Thesis is brought to you for free and open access by Lehigh Preserve. It has been accepted for inclusion in Theses and Dissertations by an authorized administrator of Lehigh Preserve. For more information, please contact preserve@lehigh.edu.
Shuler, Lori
Scenario-Based Reliability Facility Location Models

May 2004
Scenario-Based Reliability Facility Location Models

by

Lori Shuler

A Thesis
Presented to the Graduate and Research Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science

in
Information and Systems Engineering

Lehigh University

May 2004
This thesis is accepted and approved in partial fulfillment of the requirements for the Master of Science.

April 29, 2004
Date

Thesis Advisor

Chairperson of the Department
Acknowledgements

I would like to thank my advisor, Larry Snyder, for all of his support as I completed this thesis. I am very grateful for his guidance and patience during the many hours that we spent working on this project.

Also, thank you to Rosemary Berger for being a wonderful mentor and role model to me. I have greatly appreciated her advice and encouragement over the last two years.

I would also like to thank my undergraduate advisor, William Pottenger, for his guidance and for first giving me the opportunity to participate in academic research.

I am also very grateful to all of the people in the Dean’s office for their continued motivation and encouragement during my time at Lehigh.

My parents, Dave and Mary Jo Shuler, deserve my thanks for supporting me throughout my education and teaching me the value of hard work and dedication.

Finally, I would like to thank my boyfriend, Chris Cunningham, for his encouragement, love, and understanding as I have worked to complete my degrees at Lehigh.
Contents

List of Tables .................................................. vi
List of Figures .................................................. vii

Abstract ..................................................................... 1

1 Introduction .......................................................... 2
  1.1 Motivation from the Business World ..................... 3
  1.2 Classification of Uncertainty .................. 4
  1.3 Research Contributions .......................... 5
  1.4 Outline .................................................... 7

2 Literature Review .................................................. 8
  2.1 Introduction ................................................ 8
  2.2 The P-Median Problem .............................. 9
  2.3 Solution Techniques .................................. 11
  2.4 Uncertainty in Facility Location Models .......... 12
  2.5 Scenario-Planning .................................. 13
  2.6 Stochastic Location Models ..................... 14
  2.7 Robust Location Models ...................... 16
  2.8 Reliable Location Models .................... 18
  2.9 Conclusion ............................................ 20

3 Scenario-Based Reliability Models ..................... 21
  3.1 Introduction ............................................. 21
    3.1.1 Existing Reliability Models ............... 22
List of Tables

3.1 SB-RPMP-EFC data .......................... 32
3.2 Probabilities with which scenarios occur .................. 35
3.3 SB-RPMP-MFC data, original data set ...................... 43
3.4 Comparison of EFC and MFC Heuristic Performance .......... 44
3.5 SB-p-S-RPMP data ................................ 50
List of Figures

1.1 Classification of Uncertainty Problems. ......................... 4

3.1 Number of Customers vs. Expected Failure Cost. ............ 33
3.2 Number of Facilities vs. Expected Failure Cost. ............ 34
3.3 Number of Scenarios vs. Expected Failure Cost. ............ 35
3.4 Number of Failures per Scenario vs. Expected Failure Cost. 36
3.5 Number of Customers vs. Run Time. .......................... 37
3.6 Number of Facilities vs. Run Time. ............................ 37
3.7 Number of Scenarios vs. Run Time. ............................ 38
3.8 Comparison of Run Times for the EFC and MFC cases. ...... 42
3.9 EFC vs. MFC as the Number of Customers Increases. ...... 45
3.10 EFC vs. MFC as the Number of Facility Failures per Scenario Increases. ........................................ 45
Abstract

Many of the facility location models discussed in the literature are deterministic and static, meaning that all parameters are known with certainty and are assumed not to change over time. Clearly, such models have limited use in today's global business environment in which demands, costs, travel distances, and the availability of suppliers can change on a moment's notice. In this thesis we develop "reliability" models, based on the $P$-median problem, to address the issue of "supply-side" uncertainty, in which we minimize various "failure cost" objectives. The terms reliability and supply-side uncertainty describe the class of problems that optimize some objective when various components of a system fail. Facilities may fail for a number of reasons including labor strikes, severe weather, or changes in ownership. We define the failure cost of a system as the sum of the transportation costs from customers to their assigned facilities, given that one or more facilities in the system have failed.

We combine elements of stochastic, robust, and reliable optimization methods and develop three new reliability performance measures that incorporate scenario planning techniques. Stochastic methods typically minimize expected costs while robust methods generate solutions that perform well over all possible scenarios by minimizing the worst-case scenario cost. We look at both classes of models in the context of minimizing various objectives when facilities in the system fail. We also develop a reliability version of Snyder and Daskin's (2003) stochastic $p$-robust location problem. We solve each measure with AMPL/CPLEX and problem-specific heuristics, which are based on the greedy algorithm and exchange heuristic, and provide computational results.
Chapter 1

Introduction

Nearly all public and private organizations in the planning and developmental stages face issues concerning facility location. For example, owners of restaurants, stores, and other small businesses must determine locations that will serve a maximum number of potential customers. The decision is more complex for large corporations whose goals are to maximize profit throughout a complete supply chain of factories, distribution centers, and retailers. Cities of all sizes require certain public facilities, such as hospitals and schools, to comply with regulations governing acceptable travel times to the facilities and the percentage of the population served by them.

Beginning in the 1960s, models were developed to help decision-makers in these various organizations optimize objectives such as locating the minimum number of facilities required to meet customer demand, minimizing average and maximum demand-weighted distances between facilities, and minimizing fixed costs and transportation costs associated with various facility location configurations (Current, Daskin, and Schilling 2001). These original models are deterministic and static, meaning that all parameters are known with certainty and are assumed not to change over time.
1.1 Motivation from the Business World

Clearly, such models have limited use in today's global business environment in which demands, costs, travel distances, and the availability of suppliers can change on a moment's notice. Estimates of parameters, such as demand, may be inaccurate due to forecasting or measurement errors, changing market trends, or other factors.

Flexibility in the supply chain is crucial to avoid disaster when facilities fail. For instance, a fire caused by a lightning strike at a major semiconductor supplier paralyzed single-sourced Ericsson and cost the company $400 million in potential revenue, while its major competitor Nokia emerged relatively unscathed because of its network of backup suppliers (Latour 2001). In a similar incident, Toyota experienced losses of $195 million due to the company's just-in-time philosophy, which single-sourced many parts and unknowingly created vulnerabilities in the supply chain. General Motors also endured significant decreases in sales due to a labor strike that caused vehicle shortages and sent potential customers to its competition (White 1998). Martha and Vratimos (2002) provide further motivation and strategies for making demand-based supply chains better able to handle unexpected events.

Models that incorporate uncertainty are especially important in facility location problems because of the strategic (long-term) nature of the decisions and significant capital outlay that is often required. A company simply cannot afford to relocate warehouses and distribution centers as costs and customer demands shift over time, so taking these uncertainties into account in the planning phase is crucial. The term uncertainty broadly categorizes a number of modeling approaches, and for clarity we next outline the classification scheme that will be used throughout the remainder of this thesis.
1.2 Classification of Uncertainty

In the broadest sense, optimization problems contain \textit{uncertainty} when one or more parameters are random, in contrast to the deterministic \textit{certainty} problems discussed above. Within uncertainty problems, there are multiple approaches to optimization, such as \textit{robust optimization}, \textit{stochastic optimization}, and \textit{reliable optimization}. Please refer to Figure 1.1 for a visual representation of our taxonomy.

We incorporate Snyder's (2003) distinction between "demand-side" uncertainty (uncertainty in input parameters such as costs and demands) and "supply-side" uncertainty (uncertainty in the availability of facilities in the solution itself). In this classification, demand-side uncertainty encompasses \textit{robust optimization} and \textit{stochastic optimization}, while supply-side uncertainty refers to \textit{reliable optimization}.

Robust optimization finds solutions that perform well in all possible scenarios of future conditions, though not necessarily optimally in any specific situation (Snyder 2003). Random parameters may fall into known continuous value ranges or they may be described using "scenario planning." The scenario planning approach generates a number of potential scenarios, in which each

![Figure 1.1: Classification of Uncertainty Problems.](image_url)
scenario is a possible realization of the input parameters. In either case, specific probability distributions are unknown for the random parameters. Typical objectives of robust optimization problems are to minimize the maximum system wide cost or regret. The term regret refers to the difference in cost between a particular solution and the optimal solution, if the parameters had been known in advance.

In contrast, stochastic optimization typically minimizes the expected cost of the system. Stochastic models produce solutions that perform well in the long run, but may perform badly in some worst case or unlikely situations. Random parameters usually have a known probability distribution, and for our purposes we will assume this is always the case. The probability distributions may be discrete or continuous, although continuous distributions are very difficult to solve and have not received much attention in the literature. We will use discrete distributions in several of our scenario-based models.

Reliable optimization models, which address uncertainty on the supply-side, differ significantly from the previous two models. These models produce solutions that perform well when part of the system fails. In the context of facility location, a failure can result from events such as labor strikes, severe weather, or changes in ownership. This novel approach to supply chain management was developed by Snyder (2003) and Snyder and Daskin (2004). The objectives are to minimize either the expected failure cost or the maximum failure cost, where “failure cost” is defined as the transportation cost when a facility fails. The models produce trade-off curves between daily operating costs and the reliability of the system.

1.3 Research Contributions

This thesis combines elements of robust, stochastic, and reliable optimization models. Three new reliability performance measures are developed that incorporate scenario planning methods, which to date have been restricted to
stochastic and robust models. The measures are classified as reliable because our approach utilizes scenarios of various combinations of facility failures. Each scenario may be assigned a probability of occurring, depending on the performance measure. These formulations allow greater flexibility because multiple simultaneous facility failures and dependencies in the failures can be modeled. The models are viewed as extensions of the P-median problem, which is discussed in Section 2.2.

The first measure minimizes the expected failure cost of a system using a scenario planning approach. It combines stochastic and reliable optimization techniques because the scenarios of facility failure follow a discrete distribution and we minimize expected system wide failure cost. This model is concerned with achieving optimal costs in the long-run.

The second measure minimizes the maximum failure cost across scenarios. Because of the minimax failure cost structure, this model is classified as robust and reliable. There are no specific probabilities associated with the scenarios. The model minimizes the worst-case failure cost across scenarios. Both of these measures are based on the work of Snyder (2003).

The third measure is a blend of all three types of optimization: robust, stochastic, and reliable. Snyder and Daskin (2003) introduce the concept of stochastic p-robust optimization, and we extend this model to cover reliability by incorporating scenarios of facility failure. The objective is stochastic because the expected failure cost of the system is minimized and the scenarios follow a discrete distribution. The concept of p-robustness introduces a regret constraint in which the failure cost in every scenario must be within 100p% of optimal. This measure is similar to the first, with the addition of the regret constraint.

Each of the models are formulated and discussed in greater detail. We explain specific heuristics that were developed to solve the problems, provide computational results, and discuss insights that can be gained from these new reliability performance measures.
1.4 Outline

The remainder of this thesis is organized as follows. In Chapter 2 we review some of the related facility location literature. We discuss the $P$-median problem, which forms the basis of our models, and several solution techniques including the greedy algorithm and exchange heuristic. We then survey the literature on relevant stochastic, robust, and reliable optimization models, and briefly examine scenario-planning literature from the business and decision sciences perspectives. In Chapter 3 we formulate and solve the Scenario-Based Reliability $P$-Median Problems for the Expected and Maximum Failure Cost cases, and also the Scenario-Based $p$-Robust Stochastic Reliability Problem. We explain our problem-specific heuristics and include computational results comparing the performance of the AMPL/CPLEX models with our heuristics. Finally, Chapter 4 summarizes our work and offers suggestions for future research.
Chapter 2

Literature Review

2.1 Introduction

Current, Daskin, and Schilling (2001) identify eight basic deterministic facility location models: set covering, maximal covering, $P$-center, $P$-dispersion, $P$-median, fixed charge, hub, and maxisum. The first two problems locate facilities based on the assumption that demand is "covered" if it is within a certain distance from the supplier. The $P$-center problem locates $P$ facilities such that the maximum distance from any customer to its assigned supplier is minimized. The $P$-dispersion problem also locates $P$ facilities, but it maximizes the distance between the two closest facilities. The $P$-median and fixed charge problems are the focus of this research and are further explained below. The goal of the hub problem is to minimize the transportation cost of moving goods between hub and non-hub facilities in "hub and spoke" systems. Finally, the maxisum problem is used to locate "undesirable" facilities, such as prisons, and seeks to maximize the demand-weighted distance between customers and their assigned facilities. The interested reader is encouraged to view Daskin (1995) and Drezner (1995) for detailed information on model formulations and solution methods. For general information on supply chain management see Simchi-Levi, Kaminsky, and Simchi-Levi (2000).
2.2 The $P$-Median Problem

This research most closely relates to the $P$-median problem (PMP), which locates $P$ facilities such that the transportation cost between customers and their assigned facilities is minimized. The PMP was first introduced by Hakimi (1964; 1965) in the context of optimally placing “switching centers” on a network. The following notation can be used to formulate the model:

Sets

$I$ = set of customers, indexed by $i$

$J$ = set of potential facility locations, indexed by $j$

Parameters

$h_i$ = annual demand at customer $i \in I$

d$_{ij}$ = cost per unit to ship from facility location $j \in J$ to customer $i \in I$

$P$ = number of facilities to locate

Decision Variables

\[ X_j = \begin{cases} 
1, & \text{if a facility is opened at location } j \\
0, & \text{otherwise} 
\end{cases} \]

\[ Y_{ij} = \begin{cases} 
1, & \text{if a facility at location } j \in J \text{ serves customer } i \in I \\
0, & \text{otherwise} 
\end{cases} \]
The PMP is formulated as follows:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in I} \sum_{j \in J} h_{ij}d_{ij}Y_{ij} & \quad (2.1) \\
\text{subject to} & \quad \sum_{j \in J} Y_{ij} = 1 & \forall i \in I \\
& \quad Y_{ij} \leq X_j & \forall i \in I, \forall j \in J \\
& \quad \sum_{j \in J} X_j = P & \quad (2.2) \\
& \quad X_j \in \{0, 1\} & \forall j \in J \\
& \quad Y_{ij} \in \{0, 1\} & \forall i \in I, \forall j \in J & \quad (2.5)
\end{align*}
\]

The objective function (2.1) minimizes the total transportation cost between customers and their assigned facilities. Constraints (2.2) require each customer to be assigned to exactly one facility. Constraints (2.3) are called "linking constraints": a customer may only be assigned to a facility if it is opened. Constraint (2.4) requires exactly \( P \) facilities to be opened. Finally, Constraints (2.5) and (2.6) require the decision variables to be binary. It is sufficient for Constraints (2.6) to be written as \( Y_{ij} \geq 0 \ \forall i \in I, \forall j \in J \) without loss of integrality in the optimal solution. If \( Y_{ij} \) is interpreted as the percent of customer \( i \)'s demand that is satisfied by a particular facility \( j \), then in the optimal solution, all of a customer's demand will be supplied by a single facility \( (Y_{ij} = 1) \) since customers are assigned to facilities with the minimum transportation cost, or the closest open facility.

The uncapacitated fixed charge location model (UFLP: Balinski, 1965) is closely related to the PMP. It requires the following additional parameter:

\[ f_j = \text{fixed cost of opening facility } j \in J \]

and can be formulated by substituting the following for the objective function (2.1):

\[
\begin{align*}
\text{minimize} & \quad \sum_{j \in J} f_j X_j + \sum_{i \in I} \sum_{j \in J} h_{ij}d_{ij}Y_{ij} & \quad (2.7)
\end{align*}
\]
The constraints are identical to the PMP, except that Constraint (2.4) is no longer needed because the model opens the proper number of facilities to minimize the total fixed and transportation costs of satisfying customer demand. The models presented in this work are based on the PMP, but can easily be transformed into the UFLP depending on the modeler’s preferences. It should also be noted that there is a capacitated version of this problem (Daskin 1995) in which each facility has a limit on the amount of demand it can serve.

2.3 Solution Techniques

Most facility location problems are classified as NP-hard, and therefore traditional solution techniques utilizing polynomial-time algorithms cannot be used. Instead, there are two common ways to approach these problems: exact algorithms and heuristics. Exact algorithms are guaranteed to find an exact solution, while heuristics find good solutions to the problem relatively quickly, but there is no guarantee of optimality.

Lagrangian relaxation is among the best-known exact algorithms for solving facility location problems. This method creates a dual problem by omitting the constraints that make the problem difficult to solve. In exchange for removing those constraints, a penalty term is added to the objective function. A technique called “subgradient optimization” is used to find the best penalty terms that give the tightest lower bound on the optimal solution. An upper bound is calculated by converting the Lagrangian solution into a feasible solution. Fisher (1981; 1985) provides two excellent tutorials on this subject for the interested reader.

There are two main types of heuristics: construction heuristics, which create an initial feasible solution, and improvement heuristics, which, as the name suggests, attempt to improve an existing feasible solution. There are many types of construction and improvement heuristics; we use the well-known
“greedy” algorithm and “exchange heuristic” (Teitz and Bart 1968) in this thesis.

The greedy algorithm iteratively opens facilities that will provide the maximum benefit to the objective function. In the context of the PMP, the algorithm runs for $P$ cycles, and at the end of each cycle the facility that provides the largest reduction in the system wide transportation cost is fixed open. The fixed open facilities are treated as givens at the start of the next cycle. The algorithm is “greedy” because it is only concerned with opening the facility that provides the greatest benefit at the current moment, regardless of future decisions.

In our case, the exchange heuristic begins with the output of the greedy algorithm. It considers potential swaps of all pairs of open and closed facilities and implements the exchange if a pair is found that reduces the objective. There are several variations of the exchange heuristic: “first improve” and “best improve.” First improve implements the first swap that decreases the objective. This process continues until there is an iteration when all open and closed facility pairs are considered and no reduction in the objective is found.

The best improve algorithm selects the pair of facilities that has the largest decrease in the objective, after all possible combinations of open and closed facilities are analyzed. The change is implemented and the process continues. The algorithm stops when all possible swaps are compared and the “best” possible swap has either a zero or positive change in the objective. We utilize a blend of the first and best improve algorithms, which is explained in greater detail in Section 3.2.4.

2.4 Uncertainty in Facility Location Models

We now turn our focus to examine a wide variety of literature that addresses the uncertainty inherent in any type of supply chain. Simchi-Levi, Snyder, and Watsen (2002) further motivate the need for uncertainty models and discuss
the importance of flexibility in the supply chain. Owen and Daskin (1998) and
Daskin, Snyder, and Berger (2004) provide excellent surveys of the research to
date.

Averbakh and Berman (2000) state that there are two main approaches to
addressing the uncertainties of real world data: reactively, through sensitivity
analysis, or proactively, with the use of stochastic programming. They use a
similar definition of stochastic to mean that the probability distributions of
random parameters are known, and distinguish this with "true uncertainty"
situations in which probability distributions are inappropriate or unknown.

Our models examine the "true uncertainty" situations utilizing a scenario
planning approach. We first provide a background on scenario planning and
then survey the literature on stochastic, robust, and reliable optimization mod­
els. These models serve as a foundation for the reliability models that we
develop in later sections, which combine various elements of the stochastic and
robust approaches to uncertainty with reliability models.

2.5 Scenario-Planning

Beginning in the 1980s, scenario planning received considerable attention in
the business world as a means of strategic planning and as an alternative
to traditional forecasting. While forecasting generally involves extrapolat­
ing business trends to predict future outcomes, scenario planning challenges
decision-makers to identify internal forces, such as company goals, and exter­
nal forces, such as the environment and economy, and ask "what if" questions
about various possible futures. The company develops answers to these ques­
tions and in doing so prepares strategies for reacting to a number of different
scenarios, instead of relying on the future to fall in line with current trends.
In this context, the use of the term scenario planning is qualitative, referring
to a tool utilized by managers to improve the productivity and effectiveness of
their businesses. For further information on qualitative scenario-planning the
reader is referred to Amara and Lipinski (1983) and Georgantzas and Acar (1995).

Van der Heijden (1994) argues that scenario planning is more intuitive to decision-makers than other planning methods because it is based on causality. The exercise of developing scenarios helps form a solid understanding of the problem and provides a base for probabilistic planning models that rely on well-defined alternative options.

When scenarios are used to describe specific parameter values for specific applications, scenario planning can be thought of as a qualitative strategic planning method. Mobashter, Orren, and Sioshansi (1989) present a case study in which scenario-planning techniques helped electric utilities company Southern California Edison regain control after the strikes and oil crisis of the 1970s. Mulvey (1996) discusses the merits of a scenario-planning system that was installed at Towers Perrin, a pension management company. Vanston, Frisbie, Lopreato, and Poston (1977) develop a qualitative 12-step scenario generation technique.

Sheppard (1974) was the first to suggest the use of scenarios in facility location, though he does not formally include scenario-based uncertainty in his models. Owen (1999) provides an overview of relevant literature and develops scenario planning-based facility location models that incorporate uncertainty.

2.6 Stochastic Location Models

Louveauix (1986) was among the first to transform the original deterministic facility location models into 2-stage stochastic programs in which the first stage determines the location and size of the facilities and the second stage allocates production capacity based on uncertain demands. Specifically, he focused on the UFLP and the PMP. As illustrated below, many authors have developed extensions of the PMP by incorporating various elements of uncertainty, and our research, which also uses this popular model as a base, is no exception.
Mirchandani, Oudjit, and Wong (1985) develop a multidimensional (multiple objective or multiple scenario) version of the PMP with stochastic travel costs and demands. They define a method of transforming the multidimensional PMP into a deterministic PMP with a larger node set and solve the problem using Lagrangian relaxation and the subgradient method (Fisher 1981; 1985).

Weaver and Church (1983) develop solution procedures for Mirchandani and Odoni's (1979) stochastic network P-median model. They solve the specific application of locating ambulance stations considering three scenarios of quiet, normal, and rush hours in which the demand and travel distances are uncertain.

While not a PMP, Carson and Batta (1990) solve a similar problem utilizing scenario planning to locate a single ambulance on a specific college campus to minimize system wide average response time. The authors construct a network of campus locations, weight the nodes relative to the number of calls received by each node, and solve the scenario-based stochastic model heuristically. The scenarios describe the locations of the campus population throughout a 24-hour time period.

The next two models are not based on the PMP, but are related to this research and provide relevant background. Eppen, Martin, and Schrage (1989) develop a “multiproduct, multiplant, multiperiod capacity planning problem” to assist General Motors in making strategic manufacturing facility location and configuration decisions. Their model is a stochastic program with recourse that maximizes the present value of the expected discounted cash flows over optimistic, standard, and pessimistic scenarios that incorporate various economic and demand factors.

Ghosh and McLafferty (1982) develop a model to locate new supermarkets in uncertain future marketing environments, such that the solution is near optimal in every scenario. They solve the stochastic model with an exchange heuristic and provide analytical results.
Snyder and Daskin (2003) form a new optimization measure called stochastic $p$-robust optimization that combines the typical objectives of stochastic and robust models. This work was previously introduced in Section 1.3. They minimize the expected cost (or regret) while bounding the relative regret in each scenario, and the scenarios have assigned probabilities of occurring. The model is formulated both as a PMP and UFLP and is solved using variable-splitting (also known as Lagrangian decomposition).

2.7 Robust Location Models

Gupta and Rosenhead (1968) and Rosenhead, Elton and Gupta (1972) were among the first to formally define robustness measures in optimization models. However, they focus on decisions made over time, whereas we are concerned with modeling decisions made in the present that will perform well over all possible realizations of random data parameters. We explore other definitions of robustness below, and provide examples in the areas of manufacturing systems layout, international sourcing, and network design. The concept of $p$-robustness is also discussed.

Mulvey, Vanderbei, and Zenios (1995) develop a robust optimization formulation that combines goal programming with scenario-based planning. Their model generates a trade-off curve between solution robustness ("close" to optimal for all scenarios) and model robustness ("almost" feasible for all scenarios) and is illustrated with several examples, including the well-known diet problem. The distinguishing characteristic of this model is a penalty function that controls model robustness for variables dependent on random data. They discuss the benefits of robust optimization over both stochastic linear programming, another proactive approach to handling data uncertainty, and sensitivity analysis, a reactive approach.

Kouvelis, Karawarwala, and Gutiérrez (1992) construct algorithms that generate robust layout designs for manufacturing systems. Their definition of
robustness, that the solution is close to optimal over every scenario, though not necessarily optimal in any specific one, is consistent with ours. To bound the set of possible solutions, they develop an important performance measure that Snyder (2003) later coins as p-robustness: any solution must be within p% of optimal for any possible scenario, where p is some constant specified by the modeler. The solution method solves the problem for all scenarios and then uses branch and bound to generate a portfolio of solutions that are within the acceptable p% of optimal.

Gutiérrez and Kouvelis (1995) develop a scenario-based robustness approach to the complex problem of international sourcing. Exchange rates and inflation are critical and unique components to this problem and are handled in scenarios. This approach allows the decision-maker to investigate relationships between variables and specify correlation where appropriate. The model is a p-robust minimax regret UFLP that calculates the N best solutions that hedge performance against the realized exchange rates. Gutiérrez, Kouvelis, and Kurawarwala (1996) formulate a similar p-robust uncapacitated network design model to minimize routing and fixed costs. They use scenarios to represent the random routing costs and node flow volumes. Note that this work forms the basis of Snyder and Daskin’s (2003) stochastic p-robust optimization model, which was discussed in Section 2.6.

Averbakh and Berman (2000) formulate a minimax regret location model for a 1-median network problem with uncertain demands (node weights). The weights fall in some known interval, but the actual distribution is uncertain. In our taxonomy, this model is classified as a continuous value approach to a robust problem. The authors use similar definitions and refer to it as a robust problem for the case of “true uncertainty.” They develop the first polynomial-time algorithm for the robust 1-median problem on a general network.

Daskin, Hesse, and ReVelle (1997) discuss various objectives of scenario-planning models, such as optimizing the expected (stochastic) or worst-case (robust) system performance and minimizing the expected or worst-case regret.
over a number of possible scenarios that incorporate randomness in demands and transportation costs. They state weaknesses of these models and develop the $\alpha$-reliable minimax regret PMP that optimizes the worst-case regret over an endogenously determined set of scenarios whose total probability is at least some pre-specified level $\alpha$. Their model presents a "portfolio of solutions" to the decision-maker corresponding to the tradeoff between cost and reliability level. The authors refer to $\alpha$ as the reliability level, but in light of our definitions given in Section 1.2, $\alpha$ is actually considered a measure of robustness.

Lastly, not all authors concur on the definition of robustness. Bundschuh, Klabjan, and Thurston (2003) define robustness as the "extent to which a system is able to perform its intended function relatively well in the presence of failures of components or subsystems" (p. 2). They formulate a network flow model that adds redundancies to the system and limits how much a single supplier can source a particular component to hedge against supplier failures. This definition is very similar to Snyder's (2004) definition of reliability, which is covered in the next section. Bundschuh, et al. define reliability as "the probability that a system or a component performs its specified function as intended within a given time horizon and environment" (p. 2).

### 2.8 Reliable Location Models

The literature on reliability models is considerably sparser than that of the robust models, as noted by Snyder (2003). Schilling (1982) made an early contribution to this area with his Set Covering Options Analysis and Maximal Covering Options Analysis models that present various scenario-dependent facility location configurations. His models delay differentiation between the configurations as the number of facilities is increased, allowing more time before the decision-maker must commit to a specific option, which is in effect, dynamic programming. He develops a trade-off curve between the percentage of common facilities and the number of facilities. However, Daskin, Hopp, and
Medina (1992) later find that Schilling’s model produces sub-optimal results under certain scenarios.

Schilling’s models have elements of robustness and reliability. Delaying the facility location decisions and choosing facilities that are optimal in the greatest number of scenarios exhibits robust behavior. However, Schilling hints at the need for reliable models in the sense that each facility is chosen to hedge against “disruptive events.” He does not explicitly mention facility failure, but his reasoning coincides with a reliability approach to facility location. He also argues that scenarios are useful in developing contingency plans to deal with negative future situations.

Vidal and Goetschalckx (1997) underscore the lack of research on global logistics systems with random parameters. They offer a review of “strategic production-distribution models,” discuss some of the same added complexities of international models as Gutierrez and Kouvelis (1995), and formulate a model that includes supplier reliability. Vidal and Goetschalckx (2000) develop a complete model for the global supply chain, which is quite complex at the international level. The model determines all types of facility locations and their capacities, the set of suppliers and transportation channels, and the amount of raw materials and finished goods at every location of the supply chain. The added uncertainty greatly complicates the model, but is necessary to model reality more accurately. The authors use probabilities for random parameters such as demand, lead time, exchange rates, and supplier reliability. Supplier reliability is modeled by the combination of random safety stock and lead time parameters.

Snyder and Daskin (2001) incorporate reliability into the PMP and UFLP. Their models locate facilities to minimize the system cost while taking into consideration the expected transportation costs incurred when facilities fail. They solve “expected failure cost” models using Lagrangian relaxation and develop a trade-off curve between the system operating cost when all facilities are available and the expected costs of failures. Snyder (2003) also formulates
and solves reliability models for the "maximum failure cost" case.

2.9 Conclusion

The research presented in this section has illustrated a variety of facility location models that address uncertainty from the demand side (stochastic and robust models) and from the supply side (reliable models). Our goal is to combine these optimization techniques and develop new reliability performance measures that incorporate scenario planning. These models will provide decision-makers with a richer understanding of the effects of uncertainty in the supply chain.
Chapter 3

Scenario-Based Reliability Models

3.1 Introduction

The scenario-based reliability performance measures presented in the following chapters combine elements of stochastic, robust, and reliable optimization techniques. The models are based on the PMP, and have objectives that are typically thought of as stochastic and/or robust, such as minimizing expected or maximum costs. We transform these objectives into reliability measures by optimizing over various scenarios of facility failures. What were previously thought of as “operating costs” (normal transportation costs) are now considered “failure costs” because we calculate the transportation costs from customers to their assigned facilities when components of the system fail.

Before we discuss the models in detail, we return briefly to the taxonomy introduced in Section 1.2 to make an important clarification. Our models are classified as reliability models, even though they include stochastic and robust characteristics, because they deal with system performance given supply-side uncertainties. Our models do not explicitly consider demand-side uncertainties in parameters such as transportation costs, demands, and distances, but they could be extended in future research to include both types of uncertainty. For
instance, scenarios could be used to capture increased transportation costs incurred when facilities fail. For now we assume that demand-side parameters remain constant throughout the optimization.

3.1.1 Existing Reliability Models

We now revisit the “maximum failure cost” (MFC) and “expected failure cost” (EFC) reliability models developed by Snyder (2003) and Snyder and Daskin (2004) because our models directly build on their research. Both of their models hedge against failures in the system and minimize the multiple objectives of operating costs and failure costs. (As an aside, the models developed in this thesis all have single objectives: to minimize some form of failure cost in a specific manner.)

The MFC model bounds the greatest failure cost that can result from a facility failure. Each customer is assigned to one primary and one backup facility, which is sufficient because only one facility may fail at a time. Because of the model’s minimax structure, a common criticism is that the solution may be overly conservative because it optimizes based on highly unlikely worst case scenarios.

Snyder (2003) argues that the EFC model is more realistic because probability information is included. The model seeks to minimize the weighted sum of the day-to-day operating cost and the expected failure cost. Each facility is classified as either “failable” or “non-failable,” and all failable facilities have the same probability of failure. All customers are assigned to a primary facility and a set of backup facilities that can satisfy demand if the primary facility fails. This assignment strategy permits multiple simultaneous failures because as their assigned facilities fail, customers are repeatedly assigned to backup facilities. This occurs until customers are assigned to a non-failable facility, which requires no further backups.
3.1.2 Advantages of Scenario-based Reliability Models

The scenario-based models presented in this thesis have several advantages over the existing reliability models. First, there is no need for explicit customer assignments to primary and backup facilities and facilities are not classified as failable or non-failable. However, multiple simultaneous failures can still be modeled, and in addition, dependence among facility failures can be captured through the choice of scenarios. The ability to model facility dependence gives decision-makers a more accurate depiction of reality. For instance, it is likely that if one facility fails because of a weather-related incident, other facilities in the immediate geographical area will also experience problems and need to close. Another possible situation is one in which the set of potential facilities is composed of multiple companies under different ownership. For example, say that companies A, B and C represent the potential suppliers and for some reason, such as a labor strike or bankruptcy, company B becomes unavailable. We can model the scenario in which all facilities owned by company B fail simultaneously, and in doing so depict facility interdependence.

A second advantage of the scenario-based approach is that each scenario (and thus each facility failure or set of facility failures) is allowed a unique probability of occurring, in contrast with the global facility failure parameter in Snyder’s (2003) EFC model. In both cases, the probability is interpreted as the long-run percentage of time that facilities are unavailable. This pertains only to the stochastic models, since the minimax structure of the robust models does not require probability information.

We will now delve into a detailed explanation of each of the three new scenario-based reliability performance measures. In each case, we give an overview of the model, discuss the problem formulation, and explain solution techniques, including the problem-specific heuristics that were developed. To conclude each discussion, we provide computational results and insights gained from testing the model.
3.2 Expected Failure Cost Problem

3.2.1 Introduction

The Scenario-Based Reliability Expected Failure Cost P-Median Problem (SB-RPMP-EFC) is based on the PMP and minimizes the expected failure cost of a system over the long term. As in the PMP, we assume that our costs are the per-unit transportation costs from customers to their assigned facilities. Unlike the PMP, customers may not always be assigned to their closest open facilities because of failures in the system. Recall that we define “failure cost” as the overall system cost, given facility failures, and not the increase in transportation cost from a normally functioning system. The objective is stochastic because we define specific scenarios of facility failure that are assigned discrete probabilities of occurring and minimize the expected cost.

3.2.2 Notation

Much of the notation used in the PMP is reused in this model. We let $I$ be the set of customers, indexed by $i$, and $J$ be the set of potential facility locations, indexed by $j$. Set $S$, indexed by $s$, is the set of all possible scenarios, identified by the modeler, in which zero, one, or many facility failures occur. The only logical requirement for $S$ is that there cannot exist scenarios in which all facilities fail simultaneously because this would render the optimization problem infeasible.

As in the PMP, $P$ is the number of facilities to locate, $h_i$ is the annual demand at customer $i$, and $d_{ij}$ is the per-unit cost to ship from facility location $j$ to customer $i$. Additionally, we define $a_{js}$ as a binary parameter that is 1 if facility $j$ fails in scenario $s$ and 0 otherwise.

The final parameter, $q_s$, is the long-term probability with which each scenario occurs. Snyder (2003) and Owen (1999) have commented that such discrete probability distributions are difficult to define, which may be viewed
as a weakness of this model. However, the scenario-based approach provides
different insights and advantages, as discussed above, not addressed by the
continuous value range models. It simply comes down to the modeler’s needs
and preferences.

The SB-RPMP-EFC makes two decisions, each represented by a binary
variable: which facilities to open and how to optimally assign customer demand
to the open facilities across scenarios. \( X_j \) is 1 if a facility is opened at location
\( j \) and 0 otherwise. \( Y_{ijs} \) is 1 if customer \( i \) is served by facility location \( j \) in
scenario \( s \) and 0 otherwise.

### 3.2.3 Formulation

The SB-RPMP-EFC is formulated as:

\[
\text{minimize} \quad \sum_{i \in I} \sum_{j \in J} \sum_{s \in S} q_{ijs} d_{ij} Y_{ijs} \\
\text{subject to} \quad \sum_{j \in J} Y_{ijs} = 1 \quad \forall i \in I, \forall s \in S \quad (3.2)
\]

\[
Y_{ijs} \leq X_j (1 - a_{js}) \quad \forall i \in I, \forall j \in J, \forall s \in S \quad (3.3)
\]

\[
\sum_{j \in J} X_j = P \quad (3.4)
\]

\[
X_j \in \{0, 1\} \quad \forall j \in J \quad (3.5)
\]

\[
Y_{ijs} \in \{0, 1\} \quad \forall i \in I, \forall j \in J, \forall s \in S \quad (3.6)
\]

The objective function (3.1) minimizes the expected transportation cost
(failure cost) for the system across all scenarios. Constraints (3.2) guarantee
that all customers are assigned to exactly one facility in every scenario. Note
that customer assignments may change in various scenarios because of the
facility failures.

Constraints (3.3) allow a customer \( i \) to be assigned to a facility \( j \) if and
only if that facility is open and has not failed in the particular scenario \( s \). This
constraint is an extension of the PMP linking constraint. In the PMP, it is
sufficient to have $Y_{ij} \leq X_j \forall i \in I, \forall j \in J$, meaning that a customer cannot be assigned to a closed facility, but we have the additional problem of preventing customers from being assigned to facilities that have failed in the particular scenario $s$. Constraint (3.4) opens exactly $P$ facilities.

Constraints (3.5) and (3.6) are standard integrality constraints. Like in the PMP, it is sufficient for constraint (3.6) to be relaxed to $Y_{ij,s} \geq 0 \forall i \in I, \forall j \in J, \forall s \in S$ without loss of integrality in the optimal solution. This is possible because if $Y_{ij,s}$ is interpreted as the percent of customer $i$'s demand that is satisfied by a particular facility $j$, then in the optimal solution, all of a customer's demand will be supplied by a single facility ($Y_{ij,s} = 1$) since customers are assigned to facilities with the minimum transportation cost, or the closest open facility that has not failed.

3.2.4 Solution Methods

Introduction

We solve the SB-RPMP-EFC using AMPL/CPLEX and a problem-specific heuristic, implemented in C++, that is based on the greedy algorithm and exchange heuristic (Teitz and Bart 1968). The AMPL/CPLEX implementation is straightforward and can be found in Appendix A, so we use this section to discuss our C++ algorithm.

Greedy Algorithm

Our version of the greedy algorithm selects $P$ facilities for the initial solution from the set $J$. On the first pass, the facility with the minimum total transportation cost for all customers is fixed open. On the second through the $P^{th}$ passes, the candidate facility that minimizes the system transportation cost, given the previously opened facilities, is chosen. To make this decision, every customer is assigned either to their current assignment or the candidate facility, which ever one yields a lower transportation cost, and the objective is
calculated. This is repeated for all candidates and the one that minimizes the system cost is fixed open and customers are reassigned accordingly.

It is important to note that the objective calculated is not the expected failure cost because we are not considering scenarios at this point. This is necessary to prevent infeasibility issues while constructing the initial configuration. For instance, if we have opened facility $j$ and are considering candidate facility $k$, and in a particular scenario both of those facilities fail, then it appears that we have an infeasible solution, when in fact as long as $P > 2$ the solution may become feasible. (Other facilities will absorb all of the demand.)

Once we have determined an initial set of $P$ facilities we must verify that the solution is feasible across all scenarios before the expected failure cost can be calculated and passed to the exchange heuristic for optimization. To pass this check, a scenario must not exist in which all opened facilities have failed. This is computed by counting the number of facilities that are opened and failed in a particular scenario. If this counter equals $P$ then the solution is infeasible in one, and possibly more, scenarios. To correct this, we swap an opened and closed facility and recalculate the counter until we have found a feasible solution or we have exhausted all possible swaps. If the latter is true the program ends with a message stating that there is an error in the data file. Note that we do not search for the best swap, just one that makes the solution feasible, since the optimization is handled by the exchange heuristic.

It is also important to note the difference in assigning customers to facilities when scenarios are taken into account: the facility must be open and not failed in the scenario. Customers may be assigned to different facilities in different scenarios because of reassignments due to facility failures. This is a normal part of the algorithm and is incorporated into our definition of failure cost.
Our greedy algorithm can be summarized as follows:

1. bestCost ← MAX

2. loop for pass (1 to P)

   (a) loop for all \( j \in J \)

      i. find candidate facility (facility that is currently closed)
      ii. if pass = 1 assign all customers \( (i \in I) \) to candidate facility \( j \)
      iii. else assign all customers \( (i \in I) \) to the closer of candidate
           facility \( j \) or current assignment
      iv. calculate failureCost
      v. if failureCost < bestCost then
          A. bestFacility ← \( j \)
          B. bestCost ← failureCost

   (b) open bestFacility

   (c) assign all customers \( (i \in I) \) to closest open facility

**Exchange Heuristic**

As discussed in Section 2.3 there are two types of exchange heuristics: “first improve” and “best improve.” First improve swaps the first pair of open and closed facilities that reduces the objective. Best improve considers all possible exchanges before a swap is implemented and chooses the best one. First improve tends to favor facilities in the beginning of the sets of opened and closed facilities and best improve is less efficient because of the number of computations required before a swap is made. We develop a hybrid of these two methods in which the “best” closed facility is swapped with the current open facility in consideration. A more detailed explanation of our approach follows.

For a given open facility we consider all possible exchanges with closed facilities by making temporary swaps and calculating the expected failure costs. If an improvement in the objective is found (objective value decreased) we store this configuration, and implement the best swap for the open facility, provided that it reduces the objective and passes the feasibility test as described above. This process is repeated for all other open facilities.
If we cannot improve the objective by swapping any of the closed facilities with a particular open facility, we set a marker at that facility. The heuristic continues until we consider every other open facility and there is still no improvement in the objective. (The marker is unset when an improvement in the objective is found.) The purpose of the marker is to verify that all possible combinations of swaps have been considered before the algorithm ends with the optimized solution.

The exchange heuristic can be summarized as follows:

1. bestCost ← initialGreedyCost
2. improved ← false
3. marker ← -1
4. loop for all open facilities $j \in J$
   (a) loop for all closed facilities $k \in J$
      i. swap open facility $j$ and closed facility $k$
      ii. if swap passes feasibility test assign customers to closest open facility in all scenarios and calculate failureCost
      iii. if failureCost < bestCost then
         A. save swapped facility IDs
         B. bestCost ← failureCost
         C. improved ← true
         D. marker ← -1
   (b) if improved = true then implement best swap
   (c) else marker ← $j$
   (d) assign customers to closest open facility in all scenarios
   (e) improved ← false
   (f) if $j = \text{marker}$ STOP

3.2.5 Computational Results

Data Set Generation

We tested all of our models with randomly generated data sets, beginning with a “base case” of 50 facilities, 100 customers, and 5 scenarios, in which 5
facilities fail in each of the first four scenarios and there are no failures in the fifth scenario. We let the most likely scenario (zero facility failures) occur with probability 0.80, and then equally divided the remaining 0.20 among the other scenarios, giving each of them a probability of occurrence of 0.05. We wrote a small C++ program to randomly generate the indices of the facilities that fail in a particular scenario, given the number of facilities in the problem instance and the desired number of failures per scenario. The customer and potential facility location coordinates were chosen uniformly from U[0,100] and demand was randomly assigned to each customer from U[50,500].

We consider the “costs” in this model to be the demand-weighted Euclidean distances \( \sum_{i \in I} \sum_{j \in J} h_i d_{ij} Y_{ij} \) between customers and their assigned facilities. In other words, one unit of distance is equal to one dollar, which is then weighted by demand, and the expectation over all scenarios is taken to get the expected failure cost objective. The distances are interpreted as normal transportation costs if there are no facility failures, and failure costs if there are customers assigned to non-closest facilities because of failures in the system. We will loosely use the term “cost” in this way throughout the rest of this thesis.

Metrics

We designed our test plan around several major goals. First, we wanted to determine how large a problem could be solved by AMPL/Cplex in what we considered to be a reasonable amount of time, 600 seconds (10 minutes). We were interested in detecting trends such as the effect of increasing various parameters on the run times and failure costs. Secondly, we wanted to evaluate the correctness and efficiency our C++ heuristics in comparison to the AMPL models. We ran identical data sets through both so that we could objectively evaluate the results.

For each test, we recorded the expected failure cost (objective), the CPU run time in seconds, and the facilities that were opened. All testing was done
on a Dell Inspiron 5100 notebook computer with an Intel Pentium 4 Processor at 2.8 GHz. and 384 MB of memory. We used CPLEX version 8.1.

Results

After solving the base case, we were interested in finding the parameters (|I|, |J|, |S|, P, or number of facility failures per scenario) that had the most significant effect on the objective value and run time, so we increased a single parameter at a time until it exceeded our predefined maximum allowable run time. It is interesting to note the high performance of the C++ heuristic in relation to the optimal AMPL solutions. Our results are summarized in Table 3.1, and the columns are as follows.

I The number of customers in the system.
J The number of facilities in the system.
P The number of facilities to locate.
S The number of scenarios.
Fail/Scen The number of facility failures per scenario.

CPLEX Cost The expected failure cost given by AMPL/CPLEX.

CPLEX RT (s) The CPLEX solver run time in seconds.

Greedy Cost The expected failure cost after the greedy algorithm was run.

Exchange Cost The expected failure cost after the exchange heuristic was run.

C++ RT (s) The C++ heuristic run time in seconds, excluding input time.

% Error Percent error between the expected failure costs given by CPLEX and C++.
<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>P</th>
<th>S</th>
<th>Scen</th>
<th>Fail/</th>
<th>CPLEX Cost</th>
<th>CPLEX RT (s)</th>
<th>Greedy Cost</th>
<th>Exchange Cost</th>
<th>C++ Cost</th>
<th>RT (s)</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>85222</td>
<td>22</td>
<td>92349</td>
<td>85225</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>200</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>555849</td>
<td>124</td>
<td>576490</td>
<td>561431</td>
<td></td>
<td>7</td>
<td>1.00%</td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>1018498</td>
<td>279</td>
<td>1034754</td>
<td>1026482</td>
<td></td>
<td>9</td>
<td>0.78%</td>
</tr>
<tr>
<td>400</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>1527688</td>
<td>774</td>
<td>1651197</td>
<td>1527688</td>
<td></td>
<td>15</td>
<td>0.00%</td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>2000093</td>
<td>487</td>
<td>2144740</td>
<td>2022777</td>
<td></td>
<td>25</td>
<td>1.13%</td>
</tr>
<tr>
<td>600</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>&gt; 600</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>85222</td>
<td>22</td>
<td>92349</td>
<td>85225</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>84364</td>
<td>52</td>
<td>85934</td>
<td>84364</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>200</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>80366</td>
<td>60</td>
<td>87373</td>
<td>80366</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>300</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>79963</td>
<td>95</td>
<td>85635</td>
<td>79963</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>400</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>79963</td>
<td>176</td>
<td>85635</td>
<td>79963</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>500</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>79963</td>
<td>289</td>
<td>85835</td>
<td>79963</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>600</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>79963</td>
<td>292</td>
<td>85835</td>
<td>79963</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>700</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>79958</td>
<td>333</td>
<td>85835</td>
<td>79963</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>800</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>&gt; 600</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td></td>
<td>193609</td>
<td>15</td>
<td>312244</td>
<td>193609</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td>85222</td>
<td>22</td>
<td>92349</td>
<td>85225</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>20</td>
<td>5</td>
<td>5</td>
<td></td>
<td>43401</td>
<td>5</td>
<td>43968</td>
<td>43447</td>
<td></td>
<td>13</td>
<td>0.11%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td></td>
<td>90392</td>
<td>89</td>
<td>97637</td>
<td>92121</td>
<td></td>
<td>13</td>
<td>1.91%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>15</td>
<td>5</td>
<td></td>
<td>86599</td>
<td>174</td>
<td>94868</td>
<td>89572</td>
<td></td>
<td>13</td>
<td>3.43%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>20</td>
<td>5</td>
<td></td>
<td>87925</td>
<td>563</td>
<td>93779</td>
<td>87308</td>
<td></td>
<td>21</td>
<td>0.01%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>25</td>
<td>5</td>
<td></td>
<td>&gt; 600</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td></td>
<td>85222</td>
<td>22</td>
<td>92349</td>
<td>85225</td>
<td></td>
<td>5</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>15</td>
<td></td>
<td>80051</td>
<td>44</td>
<td>95700</td>
<td>89740</td>
<td></td>
<td>5</td>
<td>1.92%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>20</td>
<td></td>
<td>89122</td>
<td>33</td>
<td>96884</td>
<td>89651</td>
<td></td>
<td>4</td>
<td>0.59%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>25</td>
<td></td>
<td>92861</td>
<td>32</td>
<td>102286</td>
<td>95779</td>
<td></td>
<td>5</td>
<td>3.14%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>25</td>
<td></td>
<td>94078</td>
<td>30</td>
<td>104877</td>
<td>94194</td>
<td></td>
<td>8</td>
<td>0.12%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td></td>
<td>90392</td>
<td>87</td>
<td>97637</td>
<td>92121</td>
<td></td>
<td>13</td>
<td>1.91%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>10</td>
<td>10</td>
<td></td>
<td>93841</td>
<td>293</td>
<td>102363</td>
<td>93841</td>
<td></td>
<td>8</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>10</td>
<td>15</td>
<td></td>
<td>96752</td>
<td>143</td>
<td>105094</td>
<td>96752</td>
<td></td>
<td>6</td>
<td>0.00%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>10</td>
<td>20</td>
<td></td>
<td>102470</td>
<td>248</td>
<td>112672</td>
<td>106807</td>
<td></td>
<td>5</td>
<td>4.23%</td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>10</td>
<td>25</td>
<td></td>
<td>105160</td>
<td>115</td>
<td>124161</td>
<td>107305</td>
<td></td>
<td>5</td>
<td>1.75%</td>
</tr>
</tbody>
</table>

Table 3.1: SB-RPMP-EFC data.
The percent error of the heuristic was generally less than 1%, with a mean percent error of 0.74%. Our heuristic found the optimal solution in 50% of the test cases, including all cases when we varied the number of facilities in the system. Our results suggest that heuristics can be designed that have the ability to deliver high-performance solutions at a fraction of the run time.

As shown in Figure 3.1, the expected failure cost increased roughly linearly with the number of customers in the system. This result is somewhat intuitive because as each customer was added to the system, another transportation cost term was added to the objective. During these runs, the number of facilities opened was constant, but the actual facility locations differed depending on the number of customers. This also makes sense because the demand dispersion changed from one run to the next as more customer were added to the system. The location decisions tried to accommodate customers with greater demands since they contributed larger transportation costs to the objective.

As the number of facilities was increased, the objective approached a limit of $80,000, as seen in Figure 3.2. The actual dollar value has no significance because we were working with random data, but it is interesting that the
objective does not continue to improve after the number of facilities reaches a certain threshold value. The reasoning is that we saturated our 100x100 plane with potential facility locations and the locations became so closely packed together that one location did not offer a significant advantage over another in terms of decreasing the transportation cost. The actual facilities opened tended to vary slightly between runs because of alternate optimal solutions. It is also interesting to note that the C++ heuristic found the optimal solution in all of the test cases in which the number of customers and scenarios was held constant and we varied the number of facilities. Thus, in Figure 3.2 the C++ failure cost (after the exchange heuristic) and the AMPL failure cost lines blend together making it appear as though there are only two lines on the graph when in fact there are three.

We could not detect any meaningful relationship between the number of scenarios and the expected failure cost, as illustrated in Figure 3.3. As we increased the number of scenarios, we adjusted the probabilities with which the scenarios occurred so that the failure scenarios would not become too unlikely and therefore insignificant in the analysis. The probability of the

![Figure 3.2: Number of Facilities vs. Expected Failure Cost.](image-url)
<table>
<thead>
<tr>
<th>Number of Scenarios</th>
<th>P(fail)</th>
<th>P(not fail)</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>0.0375</td>
<td>0.700</td>
</tr>
<tr>
<td>15</td>
<td>0.0200</td>
<td>0.720</td>
</tr>
<tr>
<td>20</td>
<td>0.0150</td>
<td>0.715</td>
</tr>
<tr>
<td>25</td>
<td>0.0125</td>
<td>0.700</td>
</tr>
</tbody>
</table>

Table 3.2: Probabilities with which scenarios occur.

normal transportation cost scenario (no failures) was held around 0.70 and the failure scenarios varied between approximately 0.01 and 0.04, as shown in Table 3.2. We also had too few data points to make any valid conclusions about the effect of the number of failure scenarios on the objective because the lengthy run times very quickly ended our testing; this will be discussed in the next section.

Finally, we examined the relationship between the number of facility failures per scenario and the objective. Figure 3.4 shows that the expected failure cost increased roughly linearly (fairly slowly) as the number of failures per scenario increased for both instances of five and ten scenarios. We found that the objective differed by an additive constant when the number of scenarios

Figure 3.3: Number of Scenarios vs. Expected Failure Cost.
was doubled from five to ten.

Algorithm Performance

In general, the model run time increased as the magnitude of the parameters increased, as was first illustrated in Table 3.1. Figures 3.5, 3.6 and 3.7 also show that the CPLEX run times quickly exceeded the maximum allowable run time, while the C++ run times usually remained less than one minute. (The spike in Figure 3.5 is an outlier and can be attributed to particular difficulty in solving that problem instance.) The CPLEX run times increased at a faster rate than the C++ run times in all test runs. For instance, if the C++ run time increased linearly, the CPLEX run time increased faster than linearly, or if C++ remained relatively constant, CPLEX increased roughly linearly.

Somewhat surprisingly, increasing the number of customers was more difficult for CPLEX to solve than increasing the number of facilities, which was the opposite of the C++ heuristic. The number of constraints in the problem grew twice as fast when the number of customers was increased than when

Figure 3.4: Number of Failures per Scenario vs. Expected Failure Cost.
the number of facilities was increased. This occurred because of the assign-
ment constraint $\sum_{j \in J} Y_{ij} = 1 \forall i \in I, s \in S$, and consequently the problem
was more difficult to solve. On the other hand, the most labor-intensive com-
ponent of the C++ heuristic was considering all possible facility exchanges,
whereas assigning customers to their closest open facilities required much simpler computations.

To further investigate, we define the “size” of the problem to be $|I| \cdot |J| \cdot |S|$ and consider two problems of size 50,000. In the first problem $|I| = 200$, $|J| = 50$, and $|S| = 5$ and in the second $|I| = 100$, $|J| = 100$, and $|S| = 5$. Note that the number of constraints due to (3.4) and (3.3) are equal in both problems, but in the first problem (3.2) contributes $|I| \cdot |S| = 1000$ constraints while the second problem only has 500 assignment constraints. The first problem is from the case in which we varied the number of customers, which explains CPLEX’s difficulty in solving such problems in a reasonable amount of time.

3.2.6 Conclusion

In this section we introduced and formulated the SB-RPMP-EFC, and solved the problem using AMPL/CPLEX and a C++ heuristic. We found that CPLEX’s usefulness was limited rather quickly by what we considered to be reasonably sized problems because of excessive run times. Problems with large

![Graph showing run time vs. number of scenarios](image)

Figure 3.7: Number of Scenarios vs. Run Time.
numbers of customers or scenarios were the most difficult for CPLEX to solve because the assignment constraints greatly increased the size of the problem. These difficulties illustrate the importance of heuristics like the one we discussed in this section, which in most cases found a solution within 1% of optimality at a fraction of CPLEX's run time.

Our computational results showed that the expected failure cost increased roughly linearly with the number of customers in the system and that there is a limit to the decrease in expected failure cost that can be obtained by increasing the number of potential facility sites because the area becomes "saturated." While the number of scenarios did not have any direct impact on the objective, the number of failures per scenario was positively correlated to the expected failure cost.

3.3 Maximum Failure Cost Problem

3.3.1 Introduction

The Scenario-based Reliability Maximum Failure Cost problem (SB-RPMP-MFC) is similar in concept and formulation to the SB-RPMP-EFC. Instead of minimizing the stochastic expected failure cost objective, we minimize the robust maximum failure cost objective across scenarios. One advantage of the robust formulation is that probability information for the scenarios is not required. Recall that this discrete distribution is often difficult to identify in practice and its inaccuracy may skew the results of the stochastic model. In this robust model we instead focus on controlling the worst-case failure cost, which results from realizations of the specified scenarios. A disadvantage of this method is that the solution may place unnecessary weight on a particular worst-case and highly unlikely scenario that gives the excessive failure cost because all scenarios are considered equally. This tends to make the solution more conservative than the expected failure cost case, as shown by our
computational results in Section 3.3.5.

### 3.3.2 Notation

The only new notation introduced in the SB-RPMP-MFC is $V$, the maximum failure cost (transportation cost) across scenarios. Because of the minimax problem structure, $V$ appears in both the objective function and constraints, which are explained in the next section. Also note that the parameter $q_s$, which represented the discrete probability distribution of scenario occurrence, is not utilized in this formulation. All other parameters and decision variables remain the same.

### 3.3.3 Formulation

The SB-RPMP-MFC is formulated as:

$$\begin{align*}
\text{minimize} & \quad V \\
\text{subject to} & \quad \sum_{j \in J} Y_{ij,s} = 1 \quad \forall i \in I, \forall s \in S \tag{3.7} \\
& \quad Y_{ij,s} \leq X_j(1 - a_{js}) \quad \forall i \in I, \forall j \in J, \forall s \in S \tag{3.8} \\
& \quad \sum_{j \in J} X_j = P \tag{3.9} \\
& \quad \sum_{i \in I} \sum_{j \in J} h_{ij} Y_{ij,s} \leq V \quad \forall s \in S \tag{3.10} \\
& \quad X_j \in \{0,1\} \quad \forall j \in J \tag{3.11} \\
& \quad Y_{ij,s} \in \{0,1\} \quad \forall i \in I, \forall j \in J, \forall s \in S \tag{3.12}
\end{align*}$$

Constraints (3.11) force $V$ to be at least as great as the failure cost for every scenario and the objective (3.7) seeks to minimize $V$ as much as possible. Note that we interpret $V$ as the maximum failure cost, but by adding or multiplying Constraint (3.11) by a constant the problem can transformed to minimize the maximum regret. If we think of an $x$-$y$ coordinate plane with the failure costs on the $x$-axis and scenarios on the $y$-axis, constraints (3.11) define $s$ failure
costs and the vertical line at \( x = V \) marks the largest of these costs. The objective is to pull this line as close to the y-axis as possible. The remaining constraints are identical to the SB-RPMP-EFC and are interpreted in the same manner in this model.

### 3.3.4 Solution Methods

We again took two approaches to solve the SB-RPMP-MFC: modeling the problem in AMPL/Cplex and developing a heuristic based on the greedy algorithm and exchange heuristic. The AMPL model can be found in Appendix B. The greedy heuristic is identical to the one discussed in Section 3.2.4 for the EFC case, since both models compute transportation costs from customers to facilities in the same manner, and the specific objectives and scenarios are not yet considered.

The exchange heuristic is modified slightly from Section 3.2.4 in the way that the objective function is calculated and improvements are evaluated. In the MFC case, an improvement in the objective means that the maximum failure cost across all scenarios has decreased. When swaps are being considered, the failure cost in every scenario is computed and the largest cost is compared to the current objective value, as opposed to the weighted sum of failure costs across scenarios as in the EFC case.

### 3.3.5 Computational Results

#### Data Set Modification

We tested the SB-RPMP-MFC with the same random data sets discussed in Section 3.2.5, but with one minor modification. The parameter \( q_s \) (the probability that scenario \( s \) occurs) was deleted from the data files because we computed the worst case, instead of expected, failure costs. CPLEX had much greater difficulty solving this problem because of its minimax structure, so our maximum allowable run time quickly became a limiting factor during
the preliminary testing as illustrated in Figure 3.8 and Table 3.3.

Our initial data set with five scenarios (four failure scenarios and one normal scenario), in which five facilities failed per failure scenario, proved to be insufficient to test this model because there were numerous facilities that did not fail in any scenario. CPLEX simply opened those facilities because they had zero failure costs, which made the maximum failure cost across scenarios identical to the expected failure cost. We experimented with several options that forced CPLEX to open facilities that failed in at least one scenario to determine a base case that would produce more interesting results. Our main concern in choosing an appropriate base case was the increase in run time required for solving the more complex problems. We decided to increase the number of facility failures per scenario from five to ten. The facilities that failed were still assigned randomly, but the data set was reviewed and adjusted to guarantee that every facility failed in at least one scenario. All other parameters in the base case remained the same.

Figure 3.8: Comparison of Run Times for the EFC and MFC cases.
<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>P</th>
<th>S</th>
<th>Scen</th>
<th>Fail/ CPLEX</th>
<th>CPLEX Cost</th>
<th>RT (s)</th>
<th>Greedy Exchange</th>
<th>Exchange Cost</th>
<th>RT (s)</th>
<th>C++</th>
<th>Cost</th>
<th>RT (s)</th>
<th>% Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>85222</td>
<td>14</td>
<td></td>
<td>113591</td>
<td>85225</td>
<td></td>
<td>5</td>
<td>0.00%</td>
<td></td>
<td></td>
</tr>
<tr>
<td>200</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>555849</td>
<td>272</td>
<td></td>
<td>749807</td>
<td>561431</td>
<td>4</td>
<td>1.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1018498</td>
<td>382</td>
<td></td>
<td>1258050</td>
<td>1018705</td>
<td>7</td>
<td>0.02%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>400</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>1527688</td>
<td>403</td>
<td></td>
<td>2360511</td>
<td>1527688</td>
<td>11</td>
<td>0.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>500</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>2000093</td>
<td>388</td>
<td></td>
<td>2551097</td>
<td>2027777</td>
<td>17</td>
<td>1.13%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>600</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>&gt; 600</td>
<td>3192153</td>
<td>2528063</td>
<td>13</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>85222</td>
<td>14</td>
<td></td>
<td>113591</td>
<td>85225</td>
<td>5</td>
<td>0.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>100</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>84364</td>
<td>68</td>
<td></td>
<td>88593</td>
<td>84364</td>
<td>9</td>
<td>0.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>200</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>80366</td>
<td>274</td>
<td></td>
<td>126601</td>
<td>80366</td>
<td>24</td>
<td>0.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>300</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>&gt; 600</td>
<td>85035</td>
<td>79963</td>
<td>36</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>1</td>
<td>5</td>
<td>5</td>
<td>193609</td>
<td>12</td>
<td></td>
<td>312244</td>
<td>193609</td>
<td>0</td>
<td>0.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>85222</td>
<td>14</td>
<td></td>
<td>113591</td>
<td>85225</td>
<td>5</td>
<td>0.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>10</td>
<td>5</td>
<td>&gt; 600</td>
<td>113591</td>
<td>85225</td>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>10</td>
<td>85222</td>
<td>14</td>
<td></td>
<td>113591</td>
<td>85225</td>
<td>5</td>
<td>0.00%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>15</td>
<td>&gt; 600</td>
<td>141536</td>
<td>97907</td>
<td>4</td>
<td></td>
<td>3.41%</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Table 3.3: SB-RPMP-MFC data, original data set.

Algorithm Performance

The revised data set proved to be very difficult for CPLEX to solve; even the base case required over five minutes of computation time. Because our maximum allowable run time limited the number of CPLEX data points too severely to adequately analyze the results, we decided to test with our C++ heuristic since it performed near optimally during initial test runs at a fraction of the run time. The failure costs (objective values) and run times of the EFC and MFC heuristics for the case in which the number of customers in the system was increased are shown in Table 3.4. Note that the MFC problem was significantly more difficult to solve than the EFC problem, as indicated by the longer run times.
<table>
<thead>
<tr>
<th>I</th>
<th>Greedy Cost</th>
<th>Exchange Cost</th>
<th>C++ RT (s)</th>
<th>Greedy Cost</th>
<th>Exchange Cost</th>
<th>C++ RT (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>100756</td>
<td>97448</td>
<td>1</td>
<td>141536</td>
<td>119218</td>
<td>3</td>
</tr>
<tr>
<td>200</td>
<td>702175</td>
<td>625442</td>
<td>2</td>
<td>1483165</td>
<td>791117</td>
<td>8</td>
</tr>
<tr>
<td>300</td>
<td>1328479</td>
<td>1138086</td>
<td>3</td>
<td>3222552</td>
<td>1457750</td>
<td>7</td>
</tr>
<tr>
<td>400</td>
<td>1879156</td>
<td>1718002</td>
<td>4</td>
<td>2862235</td>
<td>2047298</td>
<td>13</td>
</tr>
<tr>
<td>500</td>
<td>2573117</td>
<td>2300492</td>
<td>4</td>
<td>4376443</td>
<td>2723991</td>
<td>16</td>
</tr>
<tr>
<td>600</td>
<td>3197868</td>
<td>2787138</td>
<td>7</td>
<td>5387908</td>
<td>3682564</td>
<td>18</td>
</tr>
</tbody>
</table>

Table 3.4: Comparison of EFC and MFC Heuristic Performance.

Results

Figure 3.9 plots the results from our testing with the larger data set for the case where the number of customers in the system was increased. These curves more closely matched our intuition about the behavior of the expected and worst-case failure cost models, namely, that the costs are greater in the MFC case. Note that the maximum failure cost deviated from the expected failure cost when facilities that failed in one or more scenarios were forced to be included in the solution. As an aside, we noticed that the number of facility failures per scenario was positively correlated with the gap between the expected and maximum failure costs. This is because the worst-case scenario cost was tempered by the smaller failure costs in the expected case, whereas the maximum failure cost objective was based only on the worst-case cost, which increased with the number of failures per scenario. These results are shown in Figure 3.10.

3.3.6 Conclusion

In this section we introduced and formulated the SB-RPMP-MFC, and solved the problem using AMPL/CPLEX and a C++ heuristic. Because the data set that was used to solve the EFC models had too few facility failures per scenario to illustrate the properties of this model, we increased the number of failures so that every facility failed in at least one scenario. As anticipated, the MFC
models showed more conservative solutions (higher failure costs) than the EFC models. We also found that CPLEX had a great deal of difficulty dealing with the minimax problem structure, and the excessive run times necessitated the

Figure 3.9: EFC vs. MFC as the Number of Customers Increases.

Figure 3.10: EFC vs. MFC as the Number of Facility Failures per Scenario Increases.
use of heuristics. As with the EFC case, our heuristics found solutions within 1% of optimality the majority of the time, with an average percent error of 0.34%. The average percent error is lower than in the EFC case, but we also have fewer data points for comparison.

### 3.4 Scenario-based $p$-Robust Stochastic Reliability Problem

#### 3.4.1 Introduction

The Scenario-based $p$-Robust Stochastic Reliability problem (SB-$p$-S-RPMP) is a reliability version of the Stochastic $p$-Robust problem ($p$-SPMP) developed by Snyder and Daskin (2003). This performance measure emphasizes the advantages of stochastic and robust optimization methods by combining them into a model that has a stochastic objective, but maximin constraints, similar to those found in the SB-RPMP-MFC, that bound the scenario (not expected) failure costs. This problem is also very similar to the SB-RPMP-EFC because we again minimize expected failure cost across scenarios. The parameter $p$ bounds the relative regret in each scenario. Regret in this context refers to the difference between the model’s solution, based on uncertainty, and the optimal solution, had the facility failures been known 	extit{a priori}. In the same manner as Snyder and Daskin (2003), we consider a solution $p$-robust if:

$$\frac{z_s(X) - z^*_s}{z^*_s} \leq p$$

or equivalently,

$$z_s(X) \leq (1 + p)z^*_s$$

in which $X$ is a feasible solution with objective $z_s(X)$ and $z^*_s$ is the optimal objective for scenario $s \in S$. In our case $z_s(X) = \sum_{j \in J} \sum_{i \in I} h_{ij}d_{ij}Y_{ij}, \forall s \in S$. The modeler can vary $p$ in the range $[0, \infty)$: if $p = 0$ the solution must be optimal in every scenario $s \in S$. If $p \to \infty$ and $|S| = 1$, the problem reduces...
to the PMP, and if \( p \to \infty \) and \(|S| \neq 1\), the problem reduces to the SB-RPMP-EFC. In general, as \( p \) decreases, the cost in each scenario approaches the optimal scenario cost \((z_s(X))\), but the cost of the system and the difficulty in solving the problem increases. Note that Snyder and Daskin (2003) also utilize a scenario-based approach, but their scenarios depict uncertainty in customer demands and transportation costs, not facility failures.

### 3.4.2 Notation

The notation for the SB-\( p \)-S-RPMP builds on that presented in Section 3.2.2 for the SB-RPMP-EFC. We add two additional parameters \((p\) and \(z^*_s\)), as discussed above, to create the "\( p \)-robustness" constraint. It is important to note that \(z^*_s\) must be calculated for every scenario \( s \in S \) prior to solving this problem. This can be done by solving the PMP \(|S|\) times, in which the data set is adjusted to account for the facility failures in the given scenario \( s \). For instance, if \( J = \{1,2,3,4,5\} \) and in a particular scenario facilities 2 and 4 fail, we would solve the PMP as if \( J = \{1,3,5\} \).
3.4.3 Formulation

The SB-p-S-RPMP is formulated as follows:

\[
\begin{align*}
\text{minimize} & \quad \sum_{i \in I} \sum_{j \in J} \sum_{s \in S} q_{ij} h_{ij} Y_{ij}, \\
\text{subject to} & \quad \sum_{j \in J} Y_{ij} = 1 \quad \forall i \in I, \forall s \in S \quad (3.16) \\
& \quad Y_{ij} \leq X_{j}(1 - a_{js}) \quad \forall i \in I, \forall j \in J, \forall s \in S \quad (3.17) \\
& \quad \sum_{j \in J} X_{j} = P \quad (3.18) \\
& \quad \sum_{i \in I} \sum_{j \in J} h_{ij} Y_{ij} \leq (1 + p)z^{*} \quad \forall s \in S \quad (3.19) \\
& \quad X_{j} \in \{0, 1\} \quad \forall j \in J \quad (3.20) \\
& \quad Y_{ij} \in \{0, 1\} \quad \forall i \in I, \forall j \in J, \forall s \in S \quad (3.21)
\end{align*}
\]

The objective (3.16) calculates the expected failure cost across scenarios. Constraints (3.20) bound the failure cost of the uncertainty solution in every scenario to be no more than 100p% from optimal. Also note the distinction between p, the desired robustness level, and P, the number of facilities to locate. All other constraints remain the same as in the SB-RPMP-EFC.

3.4.4 Solution Methods

As with the EFC and MFC problems, we solved the SB-p-S-RPMP in AMPL/Cplex and developed a heuristic based on the greedy algorithm and exchange heuristic. The AMPL model can be found in Appendix C. The greedy heuristic is again identical to the one used in the EFC case (Section 3.2.4), since both models compute transportation costs from customers to facilities in the same manner, and the specific objectives and scenarios are not yet considered.

The exchange heuristic was modified slightly from the EFC case because of the p-robustness constraint. The objective function is calculated and improvements are evaluated as described in Section 3.2.4, but the potential swap must
first pass a stricter feasibility check before it is tested to see whether it reduces the objective. In addition to verifying that all open facilities do not fail in any scenarios, the candidate objective must also be within the maximum allowable regret (100p%) of the optimal solution in all scenarios. If these conditions are not met, the swap is discarded and the heuristic continues to the next pair of swapped open and closed facilities.

3.4.5 Computational Results

Data Set Modification

We began testing the SB-p-S-RPMP with the random 50 facility, 100 customer, 5 scenario data set that was described in Section 3.2.5, but encountered a problem similar to the one that occurred with the MFC problem, described in Section 3.3.5, because there were too many facilities in the data set that did not fail in any scenario. As a result, the benefit of allowing the expected failure cost to deviate 100p% from optimality was not illustrated by our initial results because CPLEX was not forced to open facilities that had a positive failure cost. We again adjusted the data set as described in Section 3.3.5.

Results

We started with an initial p-robustness level of 0.01, and found that CPLEX reported integer infeasibility. We then varied p over the range (0,1) and found that the problem was generally infeasible for small values of p, but once the p-robustness level was increased high enough, an integer solution was found. The expected failure cost remained constant even as p was increased toward 1, the maximum value. Our results are illustrated in Table 3.5.

We concluded that our scenarios were still not diverse enough to capture the desired behavior of the SB-p-S-RPMP, namely, that the expected failure cost increases as p increases. We were not able to prove this behavior with our model for several reasons. The first is that the majority of the facilities only
Table 3.5: SB-p-S-RPMP data.

<table>
<thead>
<tr>
<th>I</th>
<th>J</th>
<th>P</th>
<th>S</th>
<th>p</th>
<th>Fail/Scen</th>
<th>AMPL Cost</th>
<th>Greedy Cost</th>
<th>Exchange Cost</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>0.01</td>
<td>infeasible</td>
<td>100756</td>
<td>98516</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>0.05</td>
<td>infeasible</td>
<td>100756</td>
<td>97448</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>0.10</td>
<td>infeasible</td>
<td>100756</td>
<td>97448</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>0.20</td>
<td>infeasible</td>
<td>100756</td>
<td>97448</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>0.25</td>
<td>infeasible</td>
<td>100756</td>
<td>97448</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>0.50</td>
<td>95599</td>
<td>100756</td>
<td>97448</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>0.75</td>
<td>95599</td>
<td>100756</td>
<td>97448</td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>50</td>
<td>5</td>
<td>5</td>
<td>1.00</td>
<td>95599</td>
<td>100756</td>
<td>97448</td>
<td></td>
</tr>
</tbody>
</table>

failed in one scenario, which kept the failure costs relatively minimal. We did not observe a difference in the objective from the EFC model because the EFC solution had already minimized regret. Small $p$ values only made the problem infeasible because a better solution did not exist, and increasing the $p$ values beyond the point of feasibility had no effect because the objective was already optimized.

### 3.4.6 Conclusion

In this section we introduced and formulated the SB-p-S-RPMP, and solved the problem using AMPL/CPLEX and a C++ heuristic. As in the MFC problem, our initial data set did not illustrate the desired properties of the model, so we expanded the data set so that facilities that failed in one or more scenarios were forced to be part of the solution. We concluded that our data set still did not encompass enough facility failures because for small values of $p$, the problems became infeasible, and once a $p$ value was reached that made the problem feasible, the objective failed to improve.
Chapter 4

Conclusion

This thesis combined elements of robust, stochastic, and reliable optimization models and developed three new reliability performance measures. Our models incorporate scenario planning methods, which to date have been reserved for stochastic and robust optimization models. The scenarios capture supply-side uncertainty since they represent various combinations of facility failures. Our formulations allow decision-makers greater flexibility because multiple simultaneous facility failures and dependencies in the failures can be modeled.

The first measure combined stochastic and reliable optimization techniques to minimize the expected failure cost of a system. The second minimized the maximum failure cost across scenarios, and is classified as robust and reliable because of the problem's minimax structure. The third measure, a reliability version of Snyder and Daskin's (2003) stochastic $p$-robust model, blended all three types of optimization. For each performance measure we formulated and discussed the model, developed AMPL/CPLEX and C++ heuristics to solve the problem, and provided computational results.

Much work can be done to enrich and develop new facility location models that encompass stochastic, robust, and reliable optimization techniques. For instance, it may be interesting to take these models one step further and incorporate demand-side uncertainty into the scenarios. A possible application might be situations in which demands change based on the availability of
various facilities.

We also considered developing reliability versions of other existing stochastic and robust problems. One measure of particular interest is Daskin, Hesse, and ReVelle's (1997) $\alpha$-Reliable $P$-Minimax Regret model. This model would blend robust and reliable optimization techniques by minimizing the worst-case failure cost over some subset of all scenarios called the "reliability set." The model endogenously places scenarios whose total probability is at least $\alpha$ from the exogenously specified set $S$ into the reliability set. The parameter $\alpha$ is specified by the modeler and can be thought of as the percentage of reality considered by the model, or the desired "$\alpha$-reliability" level. Recall from Section 2.7 that in this case reliability actually refers to a measure of robustness because we are searching for a solution that will perform well in at least $100\alpha\%$ of the originally specified scenarios.

This performance measure is important because it addresses the main weakness of the SB-RPMP-MFC, namely, that the solutions are overly conservative because unnecessary emphasis is placed on minimizing the excessive costs of highly unlikely scenarios. It is also preferable to the SB-RPMP-EFC because that problem calculates a long-run average cost, and in practice "average" conditions rarely occur.

It would also be interesting to further test the models presented here with actual data sets, such as those described in Daskin (1995) to gain insights on the effects on failure costs if entire regions of facilities failed simultaneously. Perhaps our AMPL models could also be altered to produce more reasonable run times. The list continues, and our hope is that this thesis has illustrated the importance and potential applications of scenario-based reliability models and sparked the reader's interest in continuing research in this area.
Bibliography


Appendix A

AMPL Implementation of the SB-RPMP-EFC

set I;
set J;
set S;

param q{S} >= 0;
param h{I} >= 0;
param d{I, J} >= 0;
param a{J,S} >= 0;
param P;

var X{J} binary;
var Y{S,I,J} >= 0;

minimize exp_failure_cost:
    sum{s in S, i in I, j in J} q[s]*h[i]*d[i,j]*Y[s,i,j];

subject to OpenPFac:
    sum{j in J} X[j] = P;

subject to AssignIfOpenAndNotFail{s in S, i in I, j in J}:
    Y[s,i,j] <= X[j]*(1 - a[j,s]);

subject to AssignAllCust{i in I, s in S}:
    sum{j in J} Y[s,i,j] = 1;
Appendix B

AMPL Implementation of the SB-RPMP-MFC

set I;
set J;
set S;

param h{I} >= 0;
param d{I, J} >= 0;
param a{J,S} >= 0;
param P;

var X{J} binary;
var Y{S,I,J} binary;
var V;

minimize Max_Cost: V;

subject to OpenPFac:
  sum{j in J} X[j] = P;

subject to AssignIfOpenAndNotFail{s in S, i in I, j in J}:
  Y[s,i,j] <= X[j]*(1 - a[j,s]);

subject to AssignAllCust{i in I, s in S}:
  sum{j in J} Y[s,i,j] = 1;

subject to BoundCost{s in S}:
  V >= sum{i in I, j in J} h[i]*d[i,j]*Y[s,i,j];
Appendix C

AMPL Implementation of the SB-\(p\)-S-RPMP

set I;
set J;
set S;

param q{S} >= 0;
param h{I} >= 0;
param d{I, J} >= 0;
param a{J,S} >= 0;
param P;
param p;
param z_star{S} >= 0;

var X{J} binary;
var Y{S,I,J} >= 0;

minimize total_cost:
sum{s in S, i in I, j in J} q[s]*h[i]*d[i,j]*Y[s,i,j];

subject to OpenPFac:
sum{j in J} X[j] = P;

subject to AssignIfOpenAndNotFail{s in S, i in I, j in J}:
Y[s,i,j] <= X[j]*(1 - a[j,s]);

subject to AssignAllCust{i in I, s in S}:
sum{j in J} Y[s,i,j] = 1;

subject to pRobust{s in S}:
sum{i in I, j in J} h[i]*d[i,j]*Y[s,i,j] <= (1 + p)*z_star[s];
Vita

Lori Shuler was born in Rochester, NY to David and Mary Jo Shuler on May 12, 1981. She graduated with high honors from Lehigh University in May 2003 with a Bachelor of Science in computer engineering. She was inducted into the Tau Beta Pi and Phi Eta Sigma national honor societies and received the Engineering Ingenuity Award for Undergraduate Leadership. Lori participated in the Cooperative Education program and worked for Lutron Electronics Co., Inc. in Coopersburg, PA. She was also an active member of the Society of Women Engineers.

Her publications include:


As a President’s Scholar, she is currently completing a Master of Science in Information and Systems Engineering. After graduation, Lori will begin working as an analyst for Accenture, a global consulting company.