PH.D. THESIS

Robust And Stochastic Approaches to Network Capacity Design Under Demand Uncertainty

Author:

Francis Izuagbe Garuba

Supervisors:

Peter Jacko, Marc Goerigk and Trivikram Dokka

A Thesis Submitted in Partial Fulfillment for the Degree of Doctor of Philosophy

DEPARTMENT OF MANAGEMENT SCIENCE

Lancaster, United Kingdom

Dedication

To The LORD God,

Be all the glory, the source of all grace, strength and wisdom

Declaration

I declare that the work in this thesis has been done by myself and has not been submitted elsewhere for the award of any other degree

Chapter 1 - Chapter 3 and Chapter 7 of this thesis are an original work done by myself while guided by the supervisors. Chapter 3 - Chapter 6 are included verbatim from published working papers, to writing of which the supervisors contributed with minor text amendments. Chapter 4 - Chapter 6 can be considered joint work, which are fully an original work done by myself while guided by the supervisors, except as detailed next. Marc Goerigk contributed extensively to the derivation of models presented in subsection 4.3.3. Chapter 5 includes computational results in Figure 5.3 and Section 5.4 from a code written by Marc Goerigk, who also contributed extensively to the writing of Section 5.2 and Section 5.3. Chapter 6 builds on the original idea of Trivikram Dokka which has eventually led to Lemma 6.1. All these parts are included in this thesis to maintain completeness of the research contributions and to keep a natural flow of presentation of ideas

The work presented in Chapter 3 was published as [69]. The work presented in Chapter 4 was published as [68]. The work presented in Chapter 5 was published as [70] and has been submitted for publication in Journal of the Operational Research Society. The work presented in Chapter 6 was published as [60].

Francis Izuagbe Garuba

April 2020

Acknowledgments

Quite a number of people have contributed to making this bold journey a worthwhile adventure. I still remembered the first mail I wrote to Lancaster University making contact with Prof. Adam Letchford, the encouragement, and acceptance by him mark the beginning of this journey. He made Lancaster welcoming to me and I would want to say a very big thank you for all the initial assistance and support that came from him.

Then, come Dr. Peter Jacko and Dr. Marc Goerigk, who is now Prof. Goerigk at Siegen University, contacting me from Lancaster as my supervisors and telling me what I need to do to progress with my application. These are truly Jesus' apostles sent from God to make my journey very smooth. Right from the first day, guiding me through the detailed proposal, to the last day of submitting this work, they have been very supportive and amazing. It was obvious, I was in safe hands and all through this journey, the supervision has been excellent. I am also grateful to Dr. Vikram Dokka, who joined the supervision team in the last year of my Ph.D. and provided inspiration for the last paper.

I also want to thank Dr. Christopher Kirkbride for his valuable feedback at the end of every appraisal, who also was my internal examiner. Thank goes also to Dr. Jamie Fairbrother for those valuable discussions on Julia and Prof. Guglielmo Lulli for those inspiring discussions over coffee/lunch. To everyone who made my journey in Lancaster memorable including Mrs. Gay Bentinck for her support, I say thank you. I am also very grateful to the Management Science department for the funding support provided for this research.

I do appreciate my parents for laying the foundation for this pursuit, and special appreciation to my beloved wife, Peniel, and my wonderful son, Fredrick, for their understanding and support all through this journey.

Abstract

This thesis considers the network capacity design problem with demand uncertainty using the stochastic, robust and distributionally robust stochastic optimization approaches (DRSO). Network modeling in itself has found wide areas of application in most fields of human endeavor. The network would normally consist of source (origin) and sink (destination) nodes connected by arcs that allow for flows of an entity from the origin to the destination nodes.

In this thesis, a special type of the minimum cost flow problem is addressed, the multi-commodity network flow problem. Commodities are the flow types that are transported on a shared network. Offered demands are, for the most part, unknown or uncertain, hence a model that immune against this uncertainty becomes the focus as well as the practicability of such models in the industry. This problem falls under the two-stage optimization framework where a decision is delayed in time to adjust for the first decision earlier made. The first stage decision is called the ”here and now”, while the second stage traffic re-adjustment is the ”wait and see” decision. In the literature, the decision-maker is often believed to know the shape of the uncertainty, hence we address this by considering a data-driven uncertainty set. The research also addressed the non-linearity of cost function despite the abundance of literature assuming linearity and models proposed for this.

This thesis consist of four main chapters excluding the ”Introduction” chapter and the ”Approaches to Optimization under Uncertainty” chapter where the methodologies are reviewed. The first of these four, Chapter 3, proposes the two models for the

Robust Network Capacity Expansion Problem (RNCEP) with cost non-linearity. These two are the RNCEP with fixed-charge cost and RNCEP with piecewise-linear cost. The next chapter, Chapter 4, compares the RNCEP models under two types of uncertainties in order to address the issue of usefulness in a real world setting. The resulting two robust models are also comapared with the stochastic optimization model with distribution mean. Chapter 5 re-examines the earlier problem using machine learning approaches to generate the two uncertainty sets while the last of these chapters, Chapter 6, investigates DRSO model to network capacity planning and proposes an efficient solution technique.

Contents

List of Figures

List of Tables

In the beginning God... Gen 1:1

Chapter 1

Introduction

1.1 Overview

The requirement for the efficient allocation of limited resources with its inherent challenge has always been a part of human activities and it's as old as human existence. Different generations have had to deal with this within their limit of available tools and resources, human, societal and technological development. The requirement for this gave birth to the division of labor during the industrial revolution occasioned by the coal power. Thus allocating limited resources appropriately and efficiently among competing needs is what we call optimization. Hence, an optimization framework can be defined as the tools, techniques and process around the efficient resource allocation and sharing of resources. The field of optimization has never been 'cast in stone' or perhaps no stone has been able to confine this field rather it has been an ever expanding and increasingly relevant field in almost all areas of human endeavor.

Worthy of note also, is that the field has grown to be almost all inclusive of professionals that cut across different fields of endeavor, from psychology to engineering. This field has thus engendered a lot of opportunities in different areas of its application. In this work, the optimization framework is applied to the Network Capacity Expansion Problem using stochastic programming and robust optimization approaches. Network in itself is an offshoot of the work that began with Leonhard Euler (1736) in his study of the bridges of Königsberg [80]. Ever since, this has exploded and evolved to many other areas and fields of endeavor. Network theory has thus grown into a vibrant and robust field covering wide area of applications. Its legacy is rooted in mechanics, engineering and applied mathematics as well as the contemporary field of operational research. These have helped to model many aspect of human activities and engagements arising from real world [2]. Some of these areas are social networks, biological (genetic) networks, computer networks, power networks, transportation networks (airline, rail, waterways), supply chain networks, water supply networks, gas pipeline networks and communications networks. This subject area can be traced back to the early works of Gustav Robert Kirchhoff [87] who developed the concept of tree in relation to electric current flow in network circuit: how to efficiently and effectively transport flow of electric current in the electric network and circuit, building on the work of Leonhard Euler in 1736, the father of graph theory and topology [2, 80]. Several breakthrough results in graph theory were made in the middle of the 20th century closely linked with the developments in computing technology, such as [66, 65] and [62].

Network theory is able to model the interaction between complex systems in such a way that allows a better understanding of such systems. It provides a uniform language of expression for wide variety of systems despite differences in composition, nature and domain. Models generally provide a system or mechanism that allows a representation of real word problem with mathematical objects like arcs and nodes. Network problems seek to find the most efficient way to transfer some items or entity from one location to another in a network. The movement of these entities is what allows for the flows in networks, while the entity are usually refereed to as the commodities. Network flow problem will typically consists of origin and destination nodes with several routes or links that allow for the flow of entity from the origin to destination nodes.

The network flow problem of this thesis falls under a special type of the minimum cost flow problem known as the minimum cost multicommodity flow problem. This is a network flow problem where many commodities are routed through shared resources in the underlying network. Commodities are characterized by the type of flow and different origination-destination pairs. In a typical communication network, the commodities will be the different traffic types: voice, data, video and short messaging service (sms). The objective here just like for any minimum cost flow network, is to minimized the total cost of routing the different class of flows from the origins to satisfy demands at the destinations. The origin is also referred to as the supply node and the destination as the demand node.

1.2 A Typical Network Flow Representation

Network flow models can be represented as a directed graph $G = (\mathcal{V}, \mathcal{A})$ where $v \in \mathcal{V}$ represents the set of nodes in the network and $a \in \mathcal{A}$ represents the set of directed arcs connecting the nodes in the network. (s^k, t^k) represent the origin-destination pairs for a set of commodities $k \in \mathcal{K}$ with d^k being the demand from s^k to t^k. The model below is representation of a network capacity design problem. u_a is the installed capacity on arc a while x_a is the capacity expansion on a with incremental unit capacity cost c_a and f_a^k is the flow of commodity k on arc a. $\delta^+(v)$ and $\delta^-(v)$ are sets of outgoing and incomings arcs at node v respectively.

$$\min \sum_{a \in \mathcal{A}} c_a x_a \tag{1.1}$$

$$\text{s.t.} \sum_{a\in\delta^-(v)} f_a^k - \sum_{a\in\delta^+(v)} f_a^k = \begin{cases} -d^k & \text{if } v = s^k \\ d^k & \text{if } v = t^k \\ 0 & \text{otherwise} \end{cases} \qquad \forall k \in \mathcal{K}, v \in \mathcal{V} \tag{1.2}$$

$$\sum_{k\in\mathcal{K}} f_a^k \leq u_a + x_a \qquad \forall a \in \mathcal{A} \tag{1.3}$$

$$f_a^k \geq 0 \qquad \forall k \in \mathcal{K}, a \in \mathcal{A} \tag{1.4}$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{1.5}$$

The model objective (1.1) is to minimize the cost of additional network capacity while constraint (1.2) is the flow conservation and constraint (1.3) sets the flow bound as the sum of installed and expanded capacity.

1.3 The Structure and Contributions of Thesis

This thesis is made up of seven chapters which include the Introduction chapter while Chapter 2 reviews the literature and optimization approaches to addressing uncertainty. The rest of thesis is made up of four main chapters that relate to the general area of the research thrust, "The robust and stochastic approaches to network capacity design and expansion under demand uncertainty", while Chapter 7 concludes this work. Each of these four chapters is a separate and independent piece of research paper which is already in or submitted for publication. These chapters are a bit different from the actual papers in terms of the specific journal requirements that relates to formatting and for Chapter 3, also in terms of content where the computational results have been extended. The chapters arrangement reflects progress made over time, Chapter 3 being my first paper and Chapter 6 the last paper.

A number of contributions has already been listed in each of the research papers presented in Chapter 3 to Chapter 6. The focus of this thesis is to develop a practical working models that can find significant industry adoption among practitioners leveraging on the latest advancement in mathematical programming and optimization. The aim is to bridge the gap between academic exercise and industry practicability.

Chapter 3 addresses the issue of the non-linearity of the cost function in practice, contrary to the vast assumption in the literature of linearity. It proposes two non-linear cost models to addressing this vis-a-vis (i) a linear cost with a fixed charge that is triggered if any arc capacity is modified, and (ii) its generalization that is piecewise-linear in added capacity. As expected the mixed-integer programs that result from these models are more computationally demanding compared to that with linear cost models but solutions are found to be beneficial in practice. This paper has now been published in the proceeding of 19^{th} *Symposium on Algorithmic Approaches for Transportation Modeling, Optimization, and Systems* ***(ATMOS 2019)*** as [69] and was also presented at this conference.

In Chapter 4, we delved into the practicality of existing uncertainty models in the industry by practitioners. Hence, to address this, we consider a network design and expansion problem where capacity investment decisions are made in order to meet the demand of uncertain future traffic. Three approaches were considered with comparison made based on the computational result and these three are (i) using a discrete uncertainty set (ii) using a polyhedral uncertainty set and (iii) using the mean of a per-commodity fitted zero-inflated uniform distribution. Whereas the first two models are used as part of a robust optimization setting, the third model represents a simple stochastic optimization setting. The results show that discrete uncertainty is more practicable. This paper was presented at the *30th European Conference on Operation Research* ***(EURO2019)*** and now published as a technical report in *ArXiv* online repository as [68].

Chapter 5 addresses the problem raised earlier in Chapter 4 using machine learning algorithms while developing a new formulation for the network design and expansion problem using a path based formulation. We employ clustering to generate a discrete uncertainty set, and supervised learning to generate a polyhedral uncertainty set. Performance of the resulting robust solutions for these two types of models were then compared on real-world data. The computational result shows that solutions based on discrete uncertainty generated by clustering outperform solutions based on polyhedral uncertainty generated by supervised learning on all performance metrics. This paper has been submitted for publication *in Journal of the Operational Research Society* ***(JORS)*** and is also available as a technical report in *ArXiv* online repository as [70].

Chapter 6 presents the model and algorithm for solving the network design and expansion problem using the distributionally robust stochastic optimization framework. This alternative becomes imperative with growing criticism of the robust optimization approach in the area of solution conservatism for not leveraging on the distributional knowledge of the uncertain variable. The solutions from the resulting efficient algorithm were compared to the solutions from robust optimization approach and shows that the DRSO model to be a better model in terms of both the in-sample and out of sample performance. This is published as a technical report in *ArXiv* online repository as [60].

Chapter 7 concludes the thesis and provides direction for possible future research pursuit.

Study to show thyself approved... 2 Tim 2:15

Chapter 2

Approaches to Optimization under Uncertainty

2.1 Introduction

This research is focused on the area of robust and stochastic optimization approaches to network capacity expansion under demand uncertainty [5] [112] [114] [51]. Making decision under uncertainty is as old as human existence where choices and decisions are often made in full uncertainty. Blaise Pascal's famous Wager, marked the first formal application of this in order to maximize expectation, part of the Penses published in 1670 [110]. Daniel Bernouli, in 1728, proposed a new theory, expected utility [21], while Adam Smith [120] has now been credited as the first academic or philosopher to make a distinction between uncertainty and risk [38]. [85] and [88] pioneered the economics of uncertainty and ambiguity.

However, in operations research, it was [56] and [39] who pioneered uncertainty in real world decision making. This laid the foundation for both stochastic programming and optimization under probabilistic constraints. This assumes the complete knowledge of the probability distribution of the random variables or uncertain data. Over the years a number of stochastic optimization theories and tools have been developed which have made stochastic programming a powerful and potent tool in optimization. However, the assumption of the full distribution knowledge of every data in the uncertain variable, in real world, seems to be far from truth as decision makers rarely have this information. Moreover it has also been found that computational challenges do come up that lead to hard optimization problem, which are so difficult to solve, see [84] and especially in the case of multi-stage which are ”notoriously” difficult to solve with feasible solutions often a hard problem to find, see [45]. [101, 102] showed that two-stage optimization problems are NP hard.

The limitations of stochastic programming make the need for an alternative approach to decision making under uncertainty highly imperative [34]. The framework for this alternative approach was pioneered by [121] who laid the ground work for the robust optimization approach. Here, the framework does not assume the knowledge of any probability distribution but rather that the uncertain data lie within a predetermined bounded and convex set which is often referred to as the uncertainty set (”scenario set”) [77]. A feasible solution and preferably near optimal for all possible realizations of the data in this predetermined set is termed a *robust solution* [20]. Hence, the robust seeks to maximize the worst-case utility while the stochastic programming seeks to maximize the average utility. The robust thus uses the worst-case oriented approach while the stochastic uses the average case approach.The last 20 years have witnessed a renewed interest in this framework which was led by the works of [17], [73] and [90] and followed by [32] with many other collaborators and researchers. All these have made robust optimization arguably a powerful tool to reckon with for addressing and handling uncertainty decision models [5].

Uncertainties arise for a number of reasons, namely: uncertainty about the future, measurement and computation errors and implementation errors [14]. Among researchers and practitioners, aside stochastic programming, and robust optimization, uncertainties are also dealt with using stochastic dynamic programming and the methodology of sensitivity analysis (SA). The SA is largely used to test the ”robustness” of an optimal solution and it is a ”post-mortem” tool [18]. SA is used when the optimization model is deterministic hence limited to infinitesimal changes in data perturbation. Sensitivity analysis, therefore, becomes insufficient in the face of large perturbation affecting data [73]. Intervals programming/arithmetic is also another proposed to address uncertainty, see [14].

It is worth mentioning that robust optimization framework and stochastic programming approach could be viewed as complementary approaches to addressing data perturbation as they both try to answer the same question but obviously in different ways [15]. For instance, applications in waves theory, signal processing and radio propagation which are stochastic in nature, with fairly known probability distribution would easily fit into the stochastic programming approach which has also been applied in capacity allocation problems. The capacity allocation can also be addressed under robust optimization framework.

The rest of this chapter is organized as follows. Section 2.2 presents the robust optimization methodology with its literature review. In Section 2.3, the mathematical formulation of the robust network capacity expansion problem is presented. In Section 2.4, the stochastic programming approach is presented. Section 2.5 discusses the distributionally robust stochastic optimization framework which combines the robust with the stochastic approach.

2.2 Robust Optimization Methodology and Review

In this section, a review of the robust optimization approach is carried out in more detail being one of the tools to be used in addressing the network capacity expansion problem. Robust optimization can be thought of as a framework that produces an uncertainty-immune solution to an uncertain problem [18]. There are different ways of implementation in the robust optimization framework and these are dependent on the way data uncertainty is treated: ”unknown-but-bounded” uncertainty or as random symmetric uncertainty. In the former, the true value of the uncertain parameter lies within a closed range of the nominal value, while in the latter, the true value is obtained from the nominal value by random perturbation [18, 20]. However, [90] use the uncertainty as a set of bounded scenarios in what is referred to as the scenario based set.

2.2.1 Classic Robust Optimisation

Consider the linear optimization (LO) problem of the form below;

$$\min_{x}\{c^T x : Ax \leq d\} \tag{2.1}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is decision or design vector, $\boldsymbol{c} \in \mathbb{R}^n$ is a given vector of the coefficient of the objective function, $A \in \mathbb{R}^{m \times n}$ is the constraint matrix and $\boldsymbol{d} \in \mathbb{R}^m$ is a given right hand side of the constraint [19].

The data associated with A and $\boldsymbol{d}$ in real life are uncertain in most of the cases and a vector $\boldsymbol{x}$ satisfying all possible realizations of the data in the uncertainty set Ξ given below will be the robust feasible solution to the LO problem.

$$Ax \leq d \qquad \forall \xi \equiv [A, d] \in \Xi \tag{2.2}$$

Hence we call the problem below the Robust Counterpart (RC) of the original LO problem [17].

$$\min_{x}\{c^T x : Ax \leq d \qquad \forall \xi \equiv [A, d] \in \Xi\} \tag{2.3}$$

In this RC, $\boldsymbol{x} \in \mathbb{R}^n$ is a vector of decision variable and $\boldsymbol{c} \in \mathbb{R}^n$ denotes a vector of certain coefficients while $A \in \mathbb{R}^{m \times n}$ is given matrix of uncertain coefficient, and $\boldsymbol{d} \in \mathbb{R}^m$ is a given uncertain vector belonging to an uncertainty set $\Xi \subset \mathbb{R}^{m \times n} \times \mathbb{R}^m$.

In the event that vector $\boldsymbol{c}$ in the objective function is also uncertain, the robust counterpart becomes:

$$\min_{x,t}\{t : t \geq c^T x : Ax \leq d \qquad \forall \xi \equiv [c, A, d] \in \Xi\} \tag{2.4}$$

where $A \in \mathbb{R}^{m \times n}$, $\boldsymbol{c} \in \mathbb{R}^n$ and $\boldsymbol{d} \in \mathbb{R}^m$ are uncertain coefficients belonging to an uncertainty set $\Xi \subset \mathbb{R}^{m \times n} \times \mathbb{R}^m \times \mathbb{R}^n$ with $\boldsymbol{x} \in \mathbb{R}^n$ and $t \in \mathbb{R}$ [16].

The instance above is a case of a single-stage optimization problem where all the decision variables need to be chosen before the actual realization of the uncertain data and fits perfectly to the "here and now" decision but can still fail to model the "wait and see" delayed decision of a multi-stage optimization problem, see [15].

2.2.2 Two-Stage Robust Optimization

A two-stage robust optimization problem is a typical case of a multi-stage optimization problem. In this case, the assumption of "here and now" in the single stage optimization considered in the previous subsection can be relaxed to accommodate the realistic "wait and see" decision which seems the normal occurrence in real life situation. This two-stage robust optimization was adapted from two-stage stochastic process where the second stage adjustment is referred to as a recourse.

The two-stage problems are generally characterized by the following events;

✓ The decision-maker selects the "here and now", the first stage, variables.

✓ He observes the realisation of the uncertain coefficients.

✓ He chooses the "wait and see" or the second stage variables after learning of the outcome of the random event.

A typical two-stage Robust optimisation also known as Adjustable Robust Optimisation (ARO) problem can be parameterized in the model below [16];

$$\min_{x,y(\xi)} \{c^T x : Ax + By(\xi) \leq d \quad \forall \xi \equiv [A, B, d] \in \Xi\} \tag{2.5}$$

where Ξ is a non-empty convex compact set containing all possible values of the uncertain data while $\boldsymbol{c}$, the cost coefficient vector, is assumed to be not uncertain. Equation 2.5 now yields the adjustable robust counterpart (ARC) detailed below;

$$\min_{x} \; c^T x \tag{2.6}$$

$$\text{s.t.} \quad \forall(\xi \equiv [A, B, d] \in \Xi) \; \exists y : \quad Ax + By \leq d; \tag{2.7}$$

$$x \in X \tag{2.8}$$

In the above equations, $\boldsymbol{x} \in \mathbb{R}^{n_1}$ represents the first stage decision vector, the "here and now" variable, which has to be made before the realization of the uncertain data $\xi \in \mathbb{R}^l$ with uncertainty set $\Xi \subset \mathbb{R}^l$, $\boldsymbol{y} \in \mathbb{R}^{n_2}$ represents the second stage decision, the "wait and see", which have to be made after the actual data realisation, $A \in \mathbb{R}^{m \times n_1}$ is the uncertain coefficient matrix of the "here and now" decision, $B \in \mathbb{R}^{m \times n_2}$ is the recourse coefficient matrix borrowing from two-stage stochastic programming, $X \subseteq \mathbb{R}^{n_1}$ is the feasible polyhedral set of the first stage decision.

The state of B determines the type of two-stage robust optimization. In equation Equation 2.7 above, where B is uncertain coefficient matrix, unrestricted second stage or free recourse, is typical of dynamic recourse [114]. If B becomes a certain matrix, restricted or perhaps fixed, then we have a case of a fixed, static or oblivious recourse.

However, dynamic recourse has been found generally to be computationally intractable and NP-hard, see [114, 102]. This challenge inherently makes the alternative of a restricted recourse a viable option. It was [16] that first introduce this concept in multi-stage robust optimization. In this paper, they addressed this challenge by restricting the second stage adjustable recourse to be an affine functions of the uncertain data, thereby introducing the concept of Affine Decision Rule (ADR). This restriction on B is what allowed for the framework of Affinely Adjusted Robust Counterpart (AARC). In general, decision rules provide a means to model adjustability of variables and hence are restriction on the decision-making policy. This framework is also applicable to the general multi-stage optimisation problem.

The key advantage here is that the resulting Affinely Adjustable Robust Counterpart is computationally tractable exhibiting the properties of classical robust counterparts. This affine dependency used a concept of dynamically controlled feedback system where a linear feedback is used as a controller to adjust for the desired output. This is not surprising as ADR was first proposed by [39] many years back in the framework of multi-stage stochastic programming [15]. Also, [114] cited [12] to an earlier usage of restrictive recourse in robust network design without relating to two-stage optimisation.

There are other rules apart from this affine decision rule with detailed treatment in [15] and these are Piece-Wise Affine Decision Rule, Separable Decision Rule and Quadratic Decision Rule. These alternatives are seemingly more sophisticated rules in this decision rules family and are already being investigated by researchers and practitioners to improve on the flexibility, performance and limitation of the linear ADR. Linear decision rule, like the AARC, has been found to be suboptimal in certain situations despite its optimality in other areas, hence the motivation for other variants of ADRs [28]. The search so far seems to have yielded desired result with improvement reported over linear ADR in such investigations. These variants include the piecewise decision rule which was introduced by [75] and [27], the segregated deflected linear decision rules by [45], the extended affinely adjustable robust counterpart by [46] and the constant decision rule on partition uncertainty set by [24] with further extension into multi-stage robust mixed integer problem by [25].

2.2.3 Choice of Uncertainty

The main purpose of robust optimization is to produce an uncertainty- immune solution, hence the choice of uncertainty model becomes a key consideration. There are several uncertainty models, from the Soyster's box uncertainty set to the Bertsimas' Gamma uncertainty set, regarding this which will now be discussed.

✓ **Scenario Uncertainty**. This is simply a finite set of scenarios. Given as $\Xi = \{\xi^1,,,,\xi^K\}$, where $\xi^k \in \mathbb{R}^l, k = 1, \cdots, K$. This uncertainty set is equivalent to a polytope with extreme points $\xi^1, \ldots, \xi^K$ for most classes of robust optimisation problems [90, 15].

✓ **Ellipsoidal Uncertainty**. This can be given as intersection of many finite ellipsoids. This can be denoted as $\Xi = \{\xi \in \mathbb{R}^l : \sqrt{\sum_{i=1}^{l} \xi_i^2/\sigma_i^2} \leq \Omega\}$ where $\Omega \geq 0$ [17, 74].

✓ **Polyhedral Uncertainty**. This can be viewed as the intersection of a finite number of (closed) half-spaces. This can be represented as $\Xi = \{\xi \in \mathbb{R}^l : V\xi \leq b\}$for some matrix $V \in \mathbb{R}^{i\times l}$ and right-hand side $b \in \mathbb{R}^i$ [15, 77].

✓ **Hose Model (Polyhedral model)** . This model, which is specific for network design, represents the demands as the upper bounds on traffic nodes. The symmetry form is given as $\Xi = \{\xi \in \mathbb{R}^l_+ : \xi_{ii} = 0, \sum_{j\in V}(\xi_{ij} + \xi_{ji}) \leq b_i, i \in V\}$ where b_i is a bound on the sum of the incoming and outgoing traffic on node i and $l = |V| \times |V|$ [98, 64].

✓ **Cardinal Constrained or Γ Uncertainty**. This is a type of polyhedron that encodes the budget of uncertainty as a cardinal constraint. A typical representation is as

$\Xi = \{d \in \mathbb{R}^l : \sum_{i \in \bar{V}} \lceil |d_i - \bar{d}_i| / h_i \rceil \leq \Gamma, \bar{d} - h \leq d \leq \bar{d} + h\}$
where $\bar{V} = \{i \in V : h_i > 0\}$. $l = V$ and Γ denotes the maximum number of demand that are allowed to vary from the nominal values, $\bar{d}$, while h is the bound on the nominal value [32, 4].

✓ **Budget Constraint Uncertainty**. This is a slight modification to the Γ model and can be represented by $\Xi = \{d \in \mathbb{R}^l : \sum_{i \in V} \pi_i d_i \leq \pi_0, \bar{d} - h \leq d \leq \bar{d} + h\}$
where $\bar{d} \pm h$ is the uncertainty bound and $\pi d \leq \pi_0$ is the combined allowed "budget" for uncertainty [4].

The RC also, is equivalent to a linear program (LP) where the uncertainty set is a polyhedron but becomes a conic quadratic program (quadratic constraint convex) under an ellipsoidal uncertainty set and can be solved in polynomial time. Moreover, the size of the RC is always polynomial in or bounded by a polynomial in the dimension of the original problem [17, 19]. Also, under an ellipsoidal uncertainty set, the RC of conic quadratic problem (CQP) or second order cone problem (SOCP) becomes a semi-definite problem (SDP) and the RC of an SDP is typically an NP-hard problem which is difficult to solve but approximately tractable and still solvable [30]. A feasible LO is computationally tractable where the uncertainty set meets minimal convexity requirement [17]. A feasible and bounded LO is always solvable and a polynomially solvable is also computationally tractable. It's polynomial solvable if there exist a polynomial time solution algorithm [15].

2.2.4 Robust Optimisation of Network Problems

Capacity expansion in telecoms is increasingly becoming a key strategic function within the planning and design responsibility. The explosion in demand for data with its associated applications and service at a lower telecom service rate has made it imperative for mobile operators to provide this additional capacity at an acceptable quality of service (QoS) while minimizing capital expenditure (CAPEX) investment. The field of radio communications and data/telecom network in particular has seen increased use of optimisation frameworks to address some of the challenges inherent in this technology and associated services [6, 13].

For networks, the long term strategic network planning can be viewed as the first stage "here and now", while the traffic redistribution and the consequent capacity expansion that occurs after the realisation of the traffic demand pattern would be the second stage "wait and see" adjustment decision, applying the [16] adjustable robust solution framework. The second stage adjustment relates to the concept of traffic engineering in network planning and design. This problem is a special case of a more general formulations of network design under uncertain demands and possibly uncertain travel times with a single commodity and multiple sources and sinks or with multiple commodities with single source and a single sink for each commodity. Unrestricted second stage recourse in robust network design is called dynamic routing [43]. The framework partitions the optimization variables into two sets: part of them must fix their values before the uncertainty is revealed while the rest of them can adjust themselves according to the values taken by the uncertain parameters. Most of the application of Adjustable Robust Optimisation have focused on approximations that put a restriction on the recourse.

Apparently, a special type of recourse restriction based on a specific type of uncertainty model, known as the Hose model, had been proposed independently by [61] and Fingerhut et al. [64] for Asynchronous Transfer Mode (ATM) and broadband traffic network. These also introduce the concept of static routing which [12] applied under their generalized polyhedral model using a column and constraint algorithm. The polyhedral model assumes that the demand traffic matrix belongs to a polytope set. Static routing fixes the routing for all possible realization of the demand traffic as well as the flow ratio split among these paths. [3] considered the radio network loading problem (RNL) with static routing under the hose model, [107] looked at Network capacity expansion under demand and cost uncertainty referencing [17], [89] considered the robust network design problem with static routing applying the works of [32] and lately, [111] used a cutting plane algorithm while taking into consideration the uncertainty in unmet demand outsourced cost.

On the affine routing space, there are a quite number of applications of ADR to network related problems. [109] introduce the affine routing in the their robust network capacity planning, [114] and [6] using particular uncertainty sets. As earlier identified by [28], the observed sub-optimality of ADR in certain scenarios makes the pursuit of exact solution attractive, hence new investigations into solving the exact are already being pursued.

For a linear constraint with polytopic uncertainty, the boundary of the set can be defined by the finite number of its extreme points. These are location points of a polyhedron that cannot be represented as convex combination of other points. According to Minkowski's theorem, the polytope of these set is thus the convex hull of its extreme points [12, 19]. So a method/algorithm that is able to generate the extreme point will definitely be a preferred one for solving the unrestricted ARC but the number of extreme points could be prohibitive and hence could be difficult to consider all the demand constraints of each point.

Despite this obvious challenge, recent works in the area of decomposition algorithms have helped overcome this barrier with promising and encouraging results reported so far using some algorithms. The Benders decomposition algorithm together with Kelley's cutting plane method is gradually becoming a potent tool to completely extract the full benefit of multi-stage optimization framework and especially the adjustable robust (two-stage) optimization model with extreme points generated on the fly. The decomposition algorithm uses the dual solution of the second stage (subproblem) decision to construct the value function of the first (master problem) [132]. The first attempt at a fully free recourse two-Stage optimization in Radio Network design problem was by [98] who applied a variant of branch and cut algorithm for exact solution to the splittable unrestricted recourse Network flow problem. Thereafter, [132] solved a general two-stage robust optimization problem using column and constraint generation (RCG) procedure and compared the latter numerically to row generation (RG). Recently, [5] present the general decomposition using the column-and-row generation and row generation algorithm.

2.3 Mathematical Formulation for the Robust Network Design Problem (RNDP)

2.3.1 Description

Robust Optimisation framework in particular has been implemented for Radio Telecommunications service in the area of Network Design and Expansion which falls under the two-stage optimisation problems. This helps to model effectively, decisions that are delayed in time as earlier stated. There are three closely related problems under the Radio Network problem and these are the Radio Network Design Problem (RND), the Radio Network Loading Problem (RNL) and the Virtual Private Network Problem (VPN). The RNL is a RND where the link capacities are restricted to be integers while the VPN is a RND with static routing and non splittable traffic flows [98].Another closely related area is the survivable network design (SND) which in this case allows for a robust network design in the face of node or edge failure [126].

The focus of this work is on Robust Network Capacity Expansion under demand un-

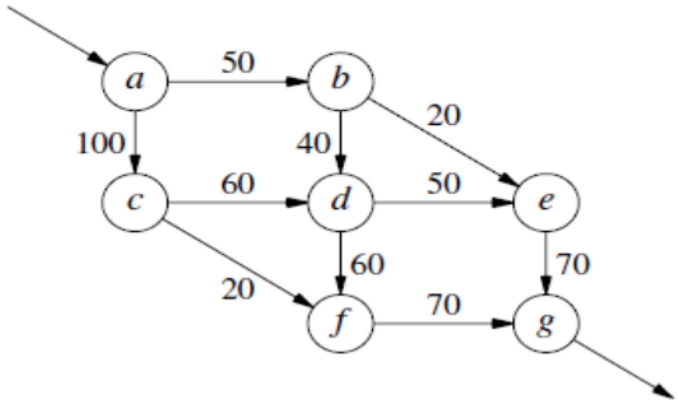


Figure 2.1: An example of a network with nodes $V = \{a, b, c, d, e, f, g\}$ with installed capacity x_a on links $a \in A$, and single-commodity demand from node $s^1 = a$ to node $t^1 = g$.

certainty for a multi-commodity flow problem. This can be represented by a directed connected graph $G = (\mathcal{V}, \mathcal{A})$ denoting a communications network topology. The Figure 4.3 is an example of a directed network with $|\mathcal{V}| = 7, |\mathcal{A}| = 10$. On each of the links $a \in \mathcal{A}$, capacities x_a can be installed which costs c_a per unit. A set of commodities K represents potential traffic demands. More precisely, a commodity $k \in K$ corresponds to node pair (s^k, t^k) and a demand $d^k \geq 0$ for traffic from $s^k \in \mathcal{V}$ to $t^k \in \mathcal{V}$. A multi-commodity flow is basically a collection of flows, one for each commodity in K. The actual demand values are considered to be uncertain and depend on the uncertain parameters $\xi \in \Xi$ that can take any value in a predetermined uncertainty set. For the moment we assume that the demand vector $\boldsymbol{d} \in \mathbb{R}^{|K|}$ corresponding to the demand values $d^k, k \in K$ lies in a given polytope $D \subset \mathbb{R}^{|K|}$ without explicitly specifying D. The actual multi-commodity flow, in any case, depends on the realization of the demand $\boldsymbol{d} \in D$.

The robust network capacity expansion problem is to find a minimum-cost for installation of integral capacities while satisfying all traffic demands d^k for $k \in K$ such that actual flow does not exceed the link capacities independently of the realization of demands in D. In this respect, robust network capacity expansion is a two-stage robust program with recourse, following the more general framework described by [16]. The capacity of edge a, x_a is the first stage decision variable, which has to be fixed before observing a demand realization $\boldsymbol{d}$. Thereafter, the routing vector $\boldsymbol{f}$, the second stage decision variable, is adjusted to compensate for gap in the first stage decision. Let $\delta^+(v)$ and $\delta^-(v)$ be the sets of outgoing and incoming links at node v, respectively. The problem can now be formulated as the following integer linear program below, a radio network design problem (RNDP);

$$\min \sum_{a \in A} c_a x_a \tag{2.9}$$

$$\text{s.t.} \sum_{a \in \delta^-(v)} f_a^k(\xi) - \sum_{a \in \delta^+(v)} f_a^k(\xi) = \begin{cases} -d^k(\xi) & \text{if } v = s^k \\ d^k(\xi) & \text{if } v = t^k, \\ 0 & \text{otherwise} \end{cases} \quad v \in V, k \in K, \xi \in \Xi \tag{2.10}$$

$$\sum_{k \in K} f_a^k(\xi) \leq x_a \qquad a \in A, \xi \in \Xi \tag{2.11}$$

$$\boldsymbol{f}, \boldsymbol{x} \geq 0 \tag{2.12}$$

Demand $\boldsymbol{d}^k$ is the traffic demand usually in Erlang for voice traffic for instance while the function $\boldsymbol{f}^k$ is the traffic flow or traffic routing. The goal, as earlier stated, is to minimize the total cost of capacity installation represented by the function (2.9) subject to flow conservation constraints (2.10) while constraints (2.11) impose that the amount of flow does not exceed the link capacity. The flow here is called dynamic routing since there is no further restriction on the routing and depends on the demand realization ξ which thus make the optimization problem intractable.

The decomposition algorithm, however, can be used to address this dynamic routing flow problem so as to generate the extreme points of the uncertain polytope on the fly [5]. A different approach was used earlier by [114] for a problem with a limited number of extreme points. Alternatively, a restriction could be placed on the routing, yielding a simple function of $\boldsymbol{d}$ which was also used in [114] for a larger number of extreme points.

2.3.2 Robust Network Capacity Expansion Problem Model (RNCEP)

A number of models were considered for investigation in addressing the topic of the research and the following model was finally proposed. In this model, referred to as RNCEP, unmet demand activates the required and optimal capacity upgrade on the existing network installed base, u_a. Hence, replacing constraint (2.11) in the RNDP, contraints (2.9 - 2.12), with the below constraint results in the RNCEP model.

$$\sum_{k \in K} f_a^k(\xi) \leq x_a + u_a \qquad a \in A, \xi \in \Xi \tag{2.13}$$

where constraint (2.13) imposes that the amount of flow does not exceed the sum of the installed and added link capacity. Constraint (2.12) imposes no upper limit on the added capacity x_a as this is needed to meet the offered traffic $\boldsymbol{d}$. However, upper bound could well be imposed if there is a constraint on the available capital budget. Also in our application, the continuous nature of this variable suffices in this instance as capacities are installed as bulk capacities in modules or boards.

Applying the framework of Affine Decision Rule (ADR) to the above RNCEP in constraint (2.10), the routing variable is thus restricted to be an affine function of the uncertainties, expressed as;

$$f_a^k(\xi) = y_a^{k0} + \sum_{h \in K} y_a^{kh} d^h(\xi), \quad a \in A, k \in K \tag{2.14}$$

where the variable y^k is the fraction of traffic demand routed which for static routing defines the routing table (the new decision variable). Other rendition of this RNCEP

[114] is by introducing a restriction on the second stage recourse thus known as static or oblivious routing which enforces f^k to be to be a linear function of ξ

$$f_a^k(\xi) = y_a^k d^k(\xi), \quad a \in A, k \in K \tag{2.15}$$

The static and affine routing RNDs are easier to solve than the original RND since the problem can now be reformulated as a static robust optimisation which can be solved by dualization.

2.4 Stochastic Programming

2.4.1 Overview

Stochastic programming as mentioned earlier in Section 2.1 is an approach used to model optimization problems with uncertain data. Most decisions in real life are made under uncertainty, basically all the variables in the decision process are unknown and even if these are known, they are not to certainty. This approach thrives on the belief or assumption that the underlying distribution of the random parameter is known or can be accurately estimated [45]. Though it was [40] who first introduced the concept of probabilistic constraints (chance constraint) in mathematical programming, the concept of stochastic programming dates back to [56] and [10]. A linear program (LP) where some of the variables are best described using random variables result in a stochastic linear program (SLP). Hence, Stochastic programming can be seen as an innovative combination of the traditional mathematical programs and stochastic models thus drawing upon traditional operations research optimization techniques [82].
Stochastic programming need to be differentiated from stochastic optimization which is basically optimization methods for minimizing or maximizing an objective function when randomness is present [79]. Stochastic programming can also be grouped as a single stage or multistage models. The most popular of the multistage is the two-stage

stochastic programming. Stochastic programming has come under criticism of late for its too conservative approach and seemingly unrealistic assumption of the real world instances, see [34].

2.4.2 A Simple Stochastic Programming Model

Consider the linear optimization (LO) problem of the form below;

$$LO = \min_{x}\{c^T x : Ax \leq d\} \tag{2.16}$$

where $\boldsymbol{x} \in \mathbb{R}^n$ is decision or design vector, $\boldsymbol{c} \in \mathbb{R}^n$ is a given vector of the coefficient of the objective function, $A \in \mathbb{R}^{m\times n}$ is the constraint matrix and $\boldsymbol{d} \in \mathbb{R}^m$ is a given right hand side of the constraint. The formulation below will describe the stochastic programming problem for this LO where the expectation is taking w.r.t the random parameter $\xi \equiv [c, A, d] \in \Xi$

$$\min_{x} \mathbb{E}_{\mathbb{F}}[\{c^T x : Ax \leq d\}] \tag{2.17}$$

where $\mathbb{F}$ is the "true" probability distribution.

2.5 Distributionally Robust Stochastic Optimization

2.5.1 Motivation

This renewed interest in the distributionally robust stochastic optimization (DRSO) framework stems from the limitations of the stochastic programming and of recent, that of robust optimization approach. The fact that DRSO approach has earlier been implemented by [118] in his min max news-vendor inventory problem and by [131] in his extension to minimax stochastic model, even before [121] robust optimization debut, is a pointer to the huge success of the robust framework. Despite the robust optimization computational tractability for wide arrays of problem instances in providing feasible solutions, it has come under criticism of late for its inability to take advantage of the underlying probability distribution of the random variable when it's available. In such instance, the robust framework produces overly conservative solution [75]. Stochastic programming has also been found to result in computationally intractable solutions as the dynamic models that result from this approach default to Monte Carlo approximations for its solution algorithm which is the case for problems with high number of scenarios [58].

The distributionally robust framework seeks to take advantage of both the probability information of the stochastic programming and the robustness of robust optimization paradigm. This brings us to the concept of risk and ambiguity in the seminal work of [88] where uncertainty is classified as risk, if the probability information is known and as ambiguity if this distribution is uncertain or unknown. In this sense, stochastic programming tends to address risk while robust optimization deals with ambiguity [33]. The need to evaluate decision makers preference over these two uncertainty types especially in the era of growing big data and business analytics has made the case for this new approach even the more compelling.

This DRSO approach does not assume to know the true distribution of the random variable but rather, that this lies within a family of distributions which is characterized by its support and moments information. This approach therefore tries to find a solution that is feasible for the worst case probability distribution in the family of distributions thus leveraging on the robust optimization framework. In the context of this new approach, the aim is to arrive at a distributionally robust solution with respect to the family of distributions. This is taking maximum expectation with respect to the worst probability distribution in this family of distributions. Although the true probability may not be known, decision makers are able to partially characterize this from domain knowledge or perhaps from historical data with support, mean and covariance [129].

The family of distributions is referred to as the ambiguity set in the same spirit with the uncertainty set in the robust optimization framework. The implication here is that the true distribution is now also subjected to uncertainty within this ambiguity set.

2.5.2 A Simple Distributionally Robust Stochastic Model

The resulting distributionally robust stochastic optimization problem can be represented below as;

$$\text{(DRSO)} \qquad \min_{x} \left(\sup_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}}[\{c^T x(\boldsymbol{\xi}) : Ax(\boldsymbol{\xi}) \leq \boldsymbol{d}\}] \right) \tag{2.18}$$

The above is an evaluation of the worst case expectation over ambiguity set $\mathcal{P}$ given that $\boldsymbol{\xi}$ follows a probability $\mathbb{P}$. The true probability $\mathbb{F}$, though unknown, lies within this ambiguity set. This set is as represented by the below for a support $\mathcal{S}$, mean $\boldsymbol{\mu}$ and covariance Σ.

$$\mathcal{P}(\mathcal{S}, \boldsymbol{\mu}, \Sigma) = \left\{ \mathbb{P} \in P \;\middle|\; \begin{array}{l} \mathbb{P}(\boldsymbol{\xi} \in \mathcal{S}) = 1 \\ \mathbb{E}_{\mathbb{P}}[\boldsymbol{\xi}] = \boldsymbol{\mu} \\ \mathbb{E}_{\mathbb{P}}[(\boldsymbol{\xi} - \boldsymbol{\mu})(\boldsymbol{\xi} - \boldsymbol{\mu})^T] \preceq \Sigma \end{array} \right\} \tag{2.19}$$

This above is called the cross-moment ambiguity set by [33] while [129] introduced the standardized ambiguity sets. Different versions have been studied in the earlier literature [130, 113, 58]. These are generally solved by relaxation to semidefinite problem (SDP) or via a partial cross moment approximation [59]. On the other hand [96] did ignore the information on the covariance and rather used the variance of the distribution as shown below, see also [33];

$$\mathcal{P}(\mathcal{S}, \boldsymbol{\mu}, \Sigma) = \left\{ \mathbb{P} \in P \;\middle|\; \begin{array}{l} \mathbb{P}(\boldsymbol{\xi} \in \mathcal{S}) = 1 \\ \mathbb{E}_{\mathbb{P}}[\boldsymbol{\xi}] = \boldsymbol{\mu} \\ \mathbb{E}_{\mathbb{P}}[(\xi_k - \mu_k)^2] = \sigma_k^2, \forall k \in [K] \end{array} \right\}$$

where σ_k^2, $k \in [K]$ is the variance of ξ_i and K be the number of elements in the vector. $\mathbb{P}(\xi \in \mathcal{S}) = 1$, represent the probability of $\boldsymbol{\xi}$ being in the set $\mathcal{S}$, a closed convex set known as the support, evaluated on the distribution $\mathbb{P}$, where P is the set of all probability measures on the measurable space $(\mathbb{R}^K, \mathcal{B})$ with $\mathcal{B}$ the Borel σ-algebra on $\mathbb{R}^K$.
If we restrict the ambiguity set $\mathcal{P}$ in (2.19) to discrete probability measures then the ambiguity set for the discrete case can be represented as below, ignoring the information on the covariance, in line with [96];

$$\mathcal{P}(\mathcal{S}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \left\{ \boldsymbol{p} \in [0,1]^{N \times K} \;\middle|\; \begin{array}{l} \sum_{i \in [N]} p_i^k = 1, \ \forall k \in K \\ \sum_{i \in [N]} p_i^k \cdot d_i^k = \mu_k, \ \forall k \in K \\ \sum_{i \in [N]} p_i^k \cdot (d_i^k)^2 = \mu_k^2 + \sigma_k^2, \ \forall k \in [K] \end{array} \right\}$$

where $\boldsymbol{p}$ is a discrete probability distribution of the random variable $\boldsymbol{d}$ with support points indices $i \in [N]$ and support points d_i^k.

And God said, Let there be light... Gen 1:3

Chapter 3

Robust Network Capacity Expansion with Non-linear Costs

Abstract. The *network capacity expansion problem* can be considered one of the key network optimization problems practitioners are expected to regularly face in present and future. There is an uncertainty associated with the capacity expansion problem in the future traffic demand, which we address using a scenario-based robust optimization approach. In most literature on network design, the costs are assumed to be linear functions of the added capacity, which is not true in practice. To address this, two non-linear cost functions are investigated: (i) a linear cost with a fixed charge that is triggered if any arc capacity is modified, and (ii) its generalization that is piecewise-linear in added capacity. The resulting mixed-integer programming model is developed with the objective of minimizing the costs, which leads to a more computationally-demanding model than the one with linear costs. We implement the model using a general purpose solver and numerical experiments have been carried out for four network structures taken from the online SNDlib database, with 26–39 nodes, 84–172 arcs, and 67–1,482 commodities. We show in a wide variety of scenarios that networks of realistic sizes can be designed using non-linear cost functions on a standard computer in a practical amount of time within negligible suboptimality. For instance, when con-

sidering the fixed charge cost under two demand scenarios, the model can be solved 68.54% of 480 instances to optimality within the time limit of 4,000sec. Although solution times increase in comparison to a linear-cost or to a non-robust model optimized for the expected traffic demand, we find solutions to be beneficial in practice. We further illustrate that including additional scenarios approximately follows the law of diminishing returns, indicating that little is gained by considering more than a handful of scenarios. Finally, we show that the results of a robust optimization model (with an increasing number of scenarios) compare favourably to the traditional deterministic model optimized for the best-case, expected, or worst-case traffic demand, suggesting that it should be used whenever computationally feasible.

Keywords: Robust Optimization; Network Design; Network Capacity Expansion; Non-linear Cost; Traffic and Transport Routing

3.1 Introduction

The demand for capacity in mobile wireless networks has seen an ever-growing trend in the last couple of decades and growth rate is expected to be even higher going into the future. This demand is fueled by the dramatic mobile traffic increase caused by the increasing availability of smartphones and other wireless devices using cloud services, social networks, Internet of things, voice over IP or video on demand. In a recent mobility report, [63], smartphones subscription are set to double the 2015 figure by 2022 to an all time high of over 6.8 billion representing over 76% of global mobile devices. According to [52], the global mobile data is expected to grow to 49 exabytes per month by 2021, at a compound annual growth rate of 47% from 2016 to 2021. This explosion in demand for data is coming at a lower cost rate. This means that in order to provide an acceptable quality of service, capacity will need to be regularly extended with optimal investment in capital expenditure. This balancing act vis-à-vis demand (traffic volume), quality of service and capital expenditure has made network capacity expansion a key strategic function resulting in billions of USD in global telecoms investment.

Similar capacity expansion challenges are present to network designers and operators in other types of networks as well, such as the Internet, transport networks, power grid, etc. The *network capacity expansion problem* can hence be considered one of the key network optimization problems practitioners are expected to regularly face in present and future.

Network design and capacity planning/expansion process is of strategic nature. It thus needs to be decided far ahead of time based on the estimation of future traffic demand. This planning also comes with an operational context. Projection for future traffic volume is usually done using traffic measurements and population statistics in combination with other marketing data. This often results in a large discrepancy between planned and actual carried traffic volume and distribution. According to [6], the difference could be as large as 10%. In the developed economies, the prevalent error in the design is traffic overestimation, resulting in over-dimensioned networks with adverse cost implications. In the developing economies, the error has been mostly traffic underestimation where the demand outstrips supply with adverse quality of service implication. Hence, the re-forecasting and re-planning becomes a continuous exercise using the traffic measurements and off-the-shelf traffic optimization tools which are based on the concept of deterministic mathematical programming which assumes that the traffic demand is estimated without error.

3.1.1 Contributions and Paper Structure

In order to have a network that is robust to the uncertainty in estimated traffic demand, this uncertainty needs to be factored in already during the planning and design process, which we address using a scenario-based robust optimization approach. This methodology is geared towards producing results that are insensitive to the uncertain demand, by considering several possible demand variation scenarios, and solves the problem considering two separate time stages, where in the first stage we determine the capacity expansion and in the second stage demand scenarios are realized. The resulting mixed-integer programming model is developed with the objective of minimizing the costs.

In most literature on network design, the costs are assumed to be linear functions of the added capacity, which is not true in practice. Real-world costs typically follow a volume discount regime which is reflected by a non-linear function. Hence, the cost function associated with network design and capacity upgrade may not be linear as such. This can be attributed to the volume discount nature of bulk buy where the customer wants to gain some cost advantage while supplier is assured of an agreed minimum order volume over a time period. To address this, two non-linear cost functions are investigated: (i) a linear cost with a fixed charge that is triggered if any arc capacity is modified, and (ii) its generalization that is piecewise-linear in added capacity.

To the best of our knowledge, this is the first paper that includes non-linear cost functions in the robust network capacity planning problem. This extension leads to a more computationally-demanding model than the one with linear cost, and the particular contributions of our paper are as follows:

- we show in a wide variety of scenarios that networks of realistic sizes can be designed using non-linear cost functions on a standard computer in a practical amount of time within negligible suboptimality;
- we present the benefits of considering a robust optimization model (even with two scenarios) instead of the traditional deterministic model optimized for the expected traffic demand;
- we present the benefits of considering non-linear costs instead of the usual linear costs which are non-realistic;

- we illustrate that including additional scenarios approximately follows the law of diminishing returns, indicating that little is gained by considering more than a handful of scenarios in as much we don't expect an infinite number of scenarios for an optimal solution;

- we show that the results of a robust optimization model (with an increasing number of scenarios) compare favourably to the traditional deterministic model optimized for the best-case, expected, or worst-case traffic demand, suggesting that it should be used whenever computationally feasible.

The rest of this paper is organized as follows. Section 6.1 presents a literature review of related research. In Section 3.3, we then introduce the problem description of robust network capacity expansion and mathematical models. Experimental results using networks from the SNDLib (see [108]) are discussed in Section 5.4. Finally, Section 3.5 concludes our work and points out future research directions.

3.2 Literature Review

In this section, we give a review of related literature on robust network design and network design under non-linear cost functions. In particular for the latter topic, research has been limited.

3.2.1 Robust Optimization in Network Design

In robust optimization, we assume that all possible data scenarios are given in the form of an uncertainty set. For general surveys, we refer, e.g., to [67, 74]. The classic approach as pioneered in [14] aims at finding a solution that is feasible for all scenarios from the uncertainty set, while optimizing a worst-case performance. This approach is relaxed through two-stage robust optimization, where not all decisions need to be taken in advance, see [16, 15]. Instead, one distinguishes between "here and now" decisions that need to be fixed in advance, and "wait and see" variables that are determined once a scenario has been revealed. Two-stage robust optimization problems are also known as adjustable robust counterparts (ARC). A general two-stage robust optimisation problem is of the following form ([16]):

$$\min_{x \in \mathcal{X}, y(\xi)} \{c^T x : Ax + By(\xi) \leq d \quad \forall \xi \equiv [A, B, d] \in \Xi\} \tag{3.1}$$

where Ξ is a set containing all possible scenarios of the uncertain data and the cost coefficient vector $\boldsymbol{c}$ is assumed to be not uncertain, without loss of generality. In this setting, $\mathcal{X} \subseteq \mathbb{R}^n$ represents the decision space for the first-stage variables, and $\boldsymbol{y}(\boldsymbol{\xi}) \in \mathbb{R}^k$ represents the second stage decision, which has to be made after the actual data realization. The matrix $B \in \mathbb{R}^{m \times k}$ is the recourse coefficient matrix.

Adjustable robust optimization has been applied to radio telecommunication services in the area of network design and expansion. This helps to model decisions that are delayed in time, e.g., traffic needs to be routed only once the demand scenario is known. Three closely related problems are the radio network design problem, the radio network loading problem and the virtual private network problem ([98]).

In telecoms, the long term strategic network planning can be viewed as the first stage "here and now" decision making, while the traffic redistribution that occurs after the realisation of the traffic demand pattern would be the second stage "wait and see" adjustment decision. The second stage adjustment, also called *recourse*, thus relates to the concept of traffic engineering in radio network planning and design. This problem is a special case of more general formulations of network design under uncertain demands and possibly uncertain travel times with a single commodity and multiple sources and sinks or with multiple commodities with single source and a single sink for each commodity. Unrestricted second stage recourse in robust network design is called dynamic routing, see [43]. Most applications of adjustable robust optimization have focused on approximations that put a restriction on the recourse.

A special type of recourse restriction based on a specific type of uncertainty model, known as the hose model, has been proposed independently by [61] and [64] for an asynchronous transfer mode and broadband traffic network. They also introduce the concept of static routing, which [12] applied under their generalized polyhedral uncertainty model using a column and constraint generation algorithm. Static routing fixes the traffic routing for all possible realization of the demand as well as the flow ratio split among these paths. [3] considered the radio network loading problem with static routing under the hose model, while [107] investigated network capacity expansion under demand and cost uncertainty. [89] considered robust network design problems with static routing applying the ideas of [32]. Lately, [111] used a cutting plane algorithm while taking into consideration the outsourcing costs for unmet demand.

Some papers use an affine decision rule to restrict the recourse decisions, thus creating a tractable robust counterpart. [109] introduced affine routing in the their robust network capacity planning model, while [114] and [6] used polyhedral uncertainty sets.

The sub-optimality of affine decision rules in certain scenarios (see [28]) makes the pursuit of exact solutions attractive. Recent works in the area of decomposition algorithms, such as Benders decomposition, have shown promising results. The first attempt at a fully free recourse two-stage optimization in radio network design problem was by Sara [98] who applied a variant of branch and cut algorithm for exact solution to the splittable unrestricted recourse network flow problem. Thereafter, [132] solved a general two-stage robust optimization problem using column and constraint generation procedure and comparing the latter numerically to row generation. Recently, [5] present a general decomposition using a column and row generation, and row generation algorithm.

3.2.2 Related Work on Non-linear Cost Functions

In general, routing costs, transportation costs or capacity costs can be a non linear functions of traffic flows. The non-linear costs we focus on in this work are given as a concave function which reflects the economies of scale present in a telecommunication network. However, non-linear routing costs could well be a convex function in a capacitated network scenario, where users pay more for increasing traffic congestion so as to allow for a balanced traffic distribution in the network with efficiency in the use of network resources. This can be viewed as a cost penalty in order to alleviate congestion. In the following, we review literature on fixed-charge costs (where there is a one-time cost associated with changing the capacity of an arc), and piecewise-linear costs.

Fixed-Charge Cost Models

In a network with fixed-charge costs, costs are not just linear in the amount of installed capacity of an arc, but also include the initial outlay cost incurred to make this arc available. In this setting, one needs to pay a fixed initial cost in addition to the arc expansion cost. The fixed costs could be the installation costs, cabinet outlay costs, additional energy or utility costs and line replacement costs. Applications are found in wide areas of network design problems and not limited to energy networks, transportation and communication. A survey is provided by [93] that demonstrate many applications in logistics, transportation and communications. The fixed-charge cost network design problem (FCND) has been found to be NP-hard, see [93, 104].

Literature on the FCND has concentrated on solution algorithms for the different model variants. [76] introduced a relax and fix heuristic for the uncapacitated FCND. [49, 50] addressed the multi-commodity capacitated FCND using a cutting plane algorithm with an improvement on the mixed-integer programming (MIP) formulation. A dual ascent approach was used by [81] for finding near optimal solutions extending on the earlier work of [8] for uncapacitated networks. However, [71] was able to show that Herrmann et al's approach was incorrect and proposed a modification. Despite this correction, Gendron claims that this procedure is still not effective as it provides a less tight bound compared to the linear programming relaxation. [53] presented a detailed and elaborate survey on the use of Benders decomposition to solving a wide range of FCND's which includes two facility networks. This can be viewed as a two-commodity network with a variant that introduces a quality of service measure, see [94]. In [1], a heuristic approach for separating and adding violated partition inequalities was implemented.

[123] solved a FCND using a variant of Benders decomposition which they referred to as the Bender-and-cut technique; a combination of Benders cut and polyhedral cut implemented in a branch and bound algorithm. They showed that the performance for the FCND changes significantly over a wide range of traffic demands. It was observed that Benders cuts are more effective for high traffic volume while polyhedral cuts (cut set inequalities) are more effective in low traffic volume situations. The two types of cut were both found to be effective in medium traffic volume situation.

The closest work to our model is [103]. Here, they formulate a robust network design problem with both transportation cost and demand uncertainty. Investment in arc capacity is modeled as a binary decision (i.e., expansion or no expansion). The model is approximated using an affine decision rule.

Piecewise-Linear Cost Models

The piecewise-linear cost model (PLC) can be used to model costs with economies of scale. In general, optimization problems involving PLC arise in domains including transportation, communications networks, large scale integrated circuits, supply chain management and logistics planning. They are usually modeled as MIPs, see [122]. The problem has been proven to be NP-hard for general concave cost objective functions, see [78].

As is the case for fixed-charge costs, most literature in this domain tends to focus on solution algorithms, see [55]. A continuous relaxation technique for solving network design with piecewise-linear costs was presented by [104]. A number of solution techniques were presented in [93], which include an adjacent extreme point search heuristic, branch and bound procedures and dynamic programming with Benders decomposition for convex costs. This paper also highlighted the extreme difficulty in solving the concave cost problem and the sub-optimality of Benders decomposition in this scenario. [78] noted that exact techniques based on dynamic programming and branch and bound are only efficient for specific subclasses of the problem. However, [127] noted that PLC modeled as a MIP can be solved with general purpose MIP solvers leveraging on technology advancement in solver development.

A number of MIP model formulations exist for piecewise-linear functions. The names for these were unified in [127]. These are Incremental or Delta (Inc) by [97], Multiple Choice Model (MCM) by [8], Convex Combination Model (CC) by [57], Special Ordered Set Type II (SOS2) by [9], Disaggregated Convex Combination Model (DCC) by [99], Logarithmic Disaggregated Convex Combination Model (DLog) by [83] and [128], and Logarithmic Covex Combination Model (Log) by [128]. [54] did a comparison of three of these models (Inc, MCM and CC), and showed their equivalence for piecewise-linear problems defined over bounded polyhedra. [127], on the other hand, provided a performance comparison of the new and existing models with extension to lower semi-continuous piecewise-linear functions. In terms of execution speed, they recommended the use of MCM or Inc for small number segments and Log or Dlog for large segments. It was also established that Dlog, DCC and MCM can be used for lower semi-continuous functions. [122] made modifications to the standard MIP models for piecewise-linear functions with a locally ideal formulation. This modification results in superior computational performance that is 40 times faster, on average, to arrive at optimal solutions.

3.3 Problem Formulation

The problem under consideration is a multi-commodity network design problem where capacities are to be added on top of existing ones on a subset of arcs, with the aim of minimizing the total cost involved and so that routing of traffic for the different commodities over the arcs subject to design and network constraints is possible. Providing this capacity should be such that it allows for robustness in the network's ability to respond to varying uncertain traffic demands. We call this problem the *Robust Network Capacity Expansion Problem* (RNCEP). We first introduce the basic problem version with linear costs, before introducing two non-linear cost extensions.

3.3.1 RNCEP with Linear Costs

A communications network topology can be represented by a directed connected graph $G = (\mathcal{V}, \mathcal{A})$. Each of the arcs $a \in \mathcal{A}$ has an original capacity u_a. The original capacity on each arc a can be expanded at a cost c_a per each additional unit of capacity. A set of commodities $\mathcal{K}$ represents potential traffic demands. A commodity $k \in \mathcal{K}$ corresponds to node pair $(s^k, t^k) \in \mathcal{V} \times \mathcal{V}$ and a demand $d^k \geq 0$ for traffic from s^k to t^k. The actual demand values are considered to be uncertain and depend on random scenarios $\xi \in \Xi$. We assume a finite set $\Xi = \{\xi^1, \dots, \xi^N\}$ of possible demand scenarios and write $d^k(\xi)$ for the demand of pair (s^k, t^k) in scenario ξ.

The robust network capacity expansion problem then is to find a minimum-cost installation of additional capacities while satisfying all traffic demands $d^k(\xi)$ for all $k \in \mathcal{K}$ and all $\xi \in \Xi$. In this respect, RNCEP is a two-stage robust program. The additional capacity we install on arc $a \in \mathcal{A}$ is denoted by x_a and is a first stage decision variable, which has to be fixed before observing a demand realization $\xi \in \Xi$. Once the demand scenario ξ becomes known, traffic is routed through a multi-commodity flow with variables $f_a^k(\xi)$.

Let $\delta^+(v)$ and $\delta^-(v)$ denote the sets of outgoing and incoming arcs at node $v \in \mathcal{V}$, respectively. The problem can now be formulated as the following linear program.

$$\min \sum_{a \in \mathcal{A}} c_a x_a \tag{3.2}$$

$$\text{s.t.} \sum_{a \in \delta^-(v)} f_a^k(\xi) - \sum_{a \in \delta^+(v)} f_a^k(\xi) = \begin{cases} -d^k(\xi) & \text{if } v = s^k \\ d^k(\xi) & \text{if } v = t^k \\ 0 & \text{otherwise} \end{cases} \quad \forall v \in \mathcal{V}, k \in \mathcal{K}, \xi \in \Xi \tag{3.3}$$

$$\sum_{k \in \mathcal{K}} f_a^k(\xi) \leq u_a + x_a \quad \forall \xi \in \Xi, a \in \mathcal{A} \tag{3.4}$$

$$f_a^k(\xi) \geq 0 \quad \forall k \in \mathcal{K}, \xi \in \Xi, a \in \mathcal{A} \tag{3.5}$$

$$x_a \geq 0 \quad \forall a \in \mathcal{A} \tag{3.6}$$

Objective function (3.2) is to minimize the total cost of capacity expansion subject to flow conservation constraint (3.3), while constraint (3.4) imposes that the amount of flow does not exceed the sum of existing and added arc capacity. For an easy reference, all the notation is also listed in Section 3.6.

3.3.2 RNCEP with Fixed-Charge Costs

We now introduce an extension of the previous model, where a fixed charge occurs if the capacity of an arc is modified. To this end, let p_a be this fixed charge associated with arc $a \in \mathcal{A}$. The resulting cost function is illustrated in Figure 3.1.

We introduce a new variable $h_a \in \{0, 1\}$ to denote if the capacity of arc a is modified. The *RNCEP with fixed-charge costs* can then be formulated as the following mixed-integer program:

$$\min \sum_{a \in \mathcal{A}} (c_a x_a + h_a p_a) \tag{3.7}$$

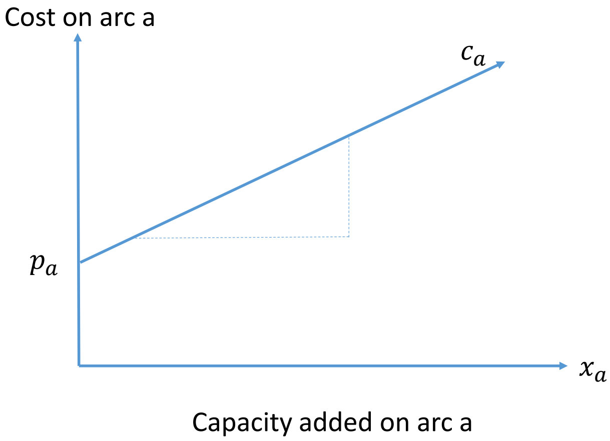


Figure 3.1: Illustration of cost function in RNCEP with fixed-charge cost.

$$\text{s.t.} \sum_{a \in \delta^-(v)} f_a^k(\xi) - \sum_{a \in \delta^+(v)} f_a^k(\xi) = \begin{cases} -d^k(\xi) & \text{if } v = s^k \\ d^k(\xi) & \text{if } v = t^k \\ 0 & \text{otherwise} \end{cases} \quad \forall v \in \mathcal{V}, k \in \mathcal{K}, \xi \in \Xi \tag{3.8}$$

$$\sum_{k \in \mathcal{K}} f_a^k(\xi) \leq x_a + u_a \quad \forall \xi \in \Xi, a \in \mathcal{A} \tag{3.9}$$

$$x_a \leq M_a h_a \quad \forall a \in \mathcal{A} \tag{3.10}$$

$$f_a^k(\xi) \geq 0 \quad \forall k \in \mathcal{K}, \xi \in \Xi, a \in \mathcal{A} \tag{3.11}$$

$$x_a \geq 0 \quad \forall a \in \mathcal{A} \tag{3.12}$$

$$h_a \in \{0, 1\} \quad \forall a \in \mathcal{A} \tag{3.13}$$

Here, M_a for all a are constants that are sufficiently large not to restrict the solution. For instance, taking any $M_a \geq \max_{\xi \in \Xi} \sum_{k \in \mathcal{K}} d^k(\xi)$ for all a is valid.

3.3.3 RNCEP with Piecewise-Linear Cost

We further extend the RNCEP by introducing a piecewise-linear cost function. To this end, we apply the multiple choice model (MCM), a general piecewise-linear model, as mentioned in the literature review. We assume that for every arc, there are up to S segments with different slopes in the cost function. Let us write $\mathcal{S} = \{1, \ldots, S\}$.

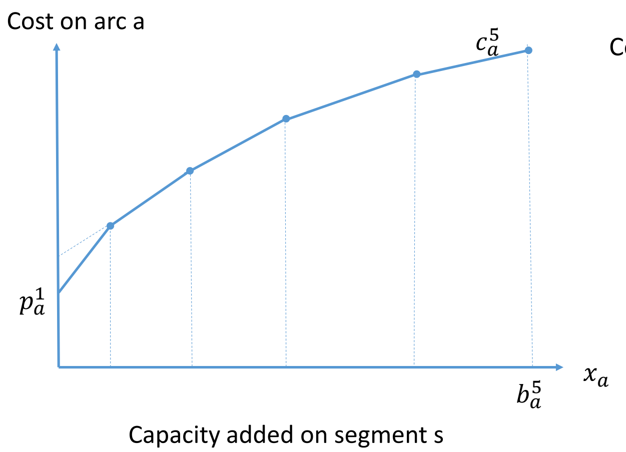


Figure 3.2: A piecewise-linear cost function.

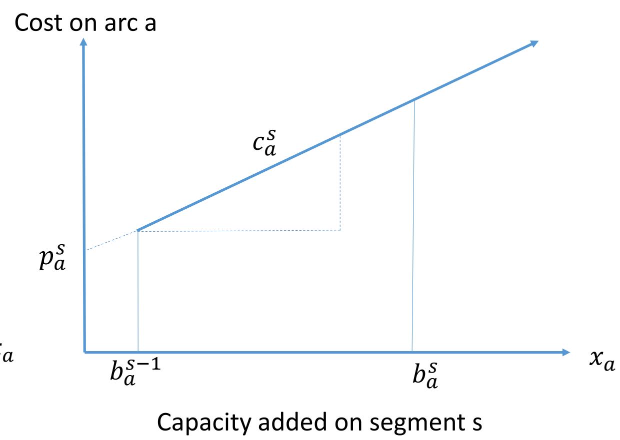


Figure 3.3: Cost segment notations.

Figure 3.4: Illustration of the cost function in RNCEP with piecewise-linear costs.

For every arc a and segment s, let b_a^s denote the load breakpoint, with an additionally defined $b_a^0 := 0$. Let c_a^s denote the cost slope of segment s, and p_a^s its y-intercept. Figure 3.4 illustrates these parameters.

In addition to the variables of RNCEP, we introduce two new sets of auxiliary variables. Variables h_a^s are binary variables that select the cost segment where the added capacity x_a falls in. Variables x_a^s denote the amount of capacity that is added within each cost segment. This gives the following mixed-integer programming formulation for the *RNCEP with piecewise-linear costs*:

$$\min \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} (c_a^s x_a^s + h_a^s p_a^s) \tag{3.14}$$

$$\text{s.t.} \sum_{a\in\delta^-(v)} f_a^k(\xi) - \sum_{a\in\delta^+(v)} f_a^k(\xi) = \begin{cases} -d^k(\xi) & \text{if } v = s^k \\ d^k(\xi) & \text{if } v = t^k \\ 0 & \text{otherwise} \end{cases} \qquad \forall v \in \mathcal{V}, k \in \mathcal{K}, \xi \in \Xi \tag{3.15}$$

$$\sum_{k\in\mathcal{K}} f_a^k(\xi) \le x_a + u_a \qquad \forall \xi \in \Xi, a \in \mathcal{A} \tag{3.16}$$

$$x_a = \sum_{s \in \mathcal{S}} x_a^s \qquad \forall a \in \mathcal{A} \tag{3.17}$$

$$b_a^{s-1} h_a^s \leq x_a^s \leq b_a^s h_a^s \qquad \forall a \in \mathcal{A}, s \in \mathcal{S} \tag{3.18}$$

$$\sum_{s \in \mathcal{S}} h_a^s \leq 1 \qquad \forall a \in \mathcal{A} \tag{3.19}$$

$$x_a \leq M_a \sum_{s \in \mathcal{S}} h_a^s \qquad \forall a \in \mathcal{A} \tag{3.20}$$

$$f_a^k(\xi) \geq 0 \qquad \forall k \in \mathcal{K}, \xi \in \Xi, a \in \mathcal{A} \tag{3.21}$$

$$x_a^s \geq 0 \qquad \forall a \in \mathcal{A}, s \in \mathcal{S} \tag{3.22}$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{3.23}$$

$$h_a^s \in \{0, 1\} \qquad \forall a \in \mathcal{A}, s \in \mathcal{S} \tag{3.24}$$

3.4 Experimental Results

We have implemented the fixed-charge cost model and the piecewise-linear cost model using instances from the SNDLib library by [108]. Network parameters characteristics on the four considered networks from SNDLib are presented in Table 3.1.

Models were implemented using Julia and Gurobi version 7.5 on a Lenovo desktop machine with 8 GB RAM and Intel Core i5-6500 CPU at 2.50Ghz on 4 Cores. While we used the same machine for all experiments, Windows 7 enterprise OS 64-bit was used for the fixed-charge cost model, and Windows 10 OS 64-bit was used for the piecewise-

Table 3.1: Network parameters characteristics (rounded to integers)

Network	Janos26	Janos39	Sun27	Node39
Number of nodes $\|\mathcal{V}\|$	26	39	27	39
Number of arcs $\|\mathcal{A}\|$	84	122	102	172
Number of commodities $\|\mathcal{K}\|$	650	1,482	67	1,471
Base demand d^k (mean±SD)	123±198	69±243	28±16	5±2
Base demand d^k (5-number summary)	12:40:60:100:1,516	1:10:22:54:5,204	4:16:28:40:56	2:3:4:6:12
Original capacity u_a (mean±SD)	64±0	1,008±0	40±0	160±0
Unit capacity expansion cost c_a (mean±SD)	468±225	313±162	19±10	23±11

linear cost model. In Gurobi, we have used a time limit of $4000s$ for each problem instance and optimality is achieved once the optimality gap is below 0.01%.

3.4.1 Experimental Setup

Both the fixed-charge cost and the piecewise-linear cost models were implemented with one scenario (single-scenario) and with two scenarios (double-scenario). The base demand scenario was provided from the SNDLib library, which we have randomly modified to generate additional demand scenarios. The amount of modification is controlled by a parameter λ, the maximum deviation of modified demand from the base demand. The parameter λ is chosen to be a fraction of the mean base demand $\widehat{d}$; we consider $\lambda = \text{round}(0.3\widehat{d})$ and $\lambda = 2 \cdot \text{round}(0.3\widehat{d})$, corresponding to small uncertainty and large uncertainty, respectively. The value is then used as a bound for uniformly generating the modified demands around the base demand of every arc. Note that it can happen that a particular demand becomes negative, which is accommodated by switching the origin and destination nodes, and which further implies that a demand between two nodes can actually increase (in its absolute value) even if a negative random number was added to the base demand.

We take $M_a := 2\max_{\xi\in\Xi}\sum_{k\in\mathcal{K}} d^k(\xi)$ for all a. We have found in our experiments (not reported here) that taking larger M_a results in shorter solution time, although taking a too large M_a results in an imprecise solution.

We summarize the experimental setup in Table 3.2. For each of the four networks,

Table 3.2: Experimental setup for generating 120 problem instances for each network.

Parameters	**# options**	**Options**
Number of scenarios	2	1 (single) / 2 (double)
Scenario variability λ	2	$0.3\widehat{d}$ / $0.6\widehat{d}$
Fixed-charge factor P	3	0 / 10 / 100
Number of runs	10	—

Table 3.3: Proportion of instances not solved to optimality within the time limit (rounded to one decimal).

Network	**Janos26**	**Janos39**	**Sun27**	**Node39**
Total	0.0%	24.2%	35.0%	66.7%
$P = 0$	0.0%	0.0%	0.0%	0.0%
$P = 10$	0.0%	0.0%	12.5%	100.0%
$P = 100$	0.0%	72.5%	92.5%	100.0%
Single-scenario	0.0%	15.0%	28.3%	66.7%
Double-scenario	0.0%	33.3%	41.7%	66.7%

we consider the single-scenario and the double-scenario case, as well as small and large uncertainty. Additionally, for fixed-cost models we use three different fixed-charge factors P. These are used to calculate the fixed charges p_a of arc a by setting $p_a = Pc_a$. With $P = 0$, we recover the basic linear cost model without fixed charge. All networks and parameter settings are run 10 times to reduce variability in the results. In total, this gives $4 \cdot 2 \cdot 2 \cdot 3 \cdot 10 = 480$ optimization problem instances that need to be solved for the fixed charge case. For the piecewise-linear case, we follow the same setup with $4 \cdot 2 \cdot 2 \cdot 10 = 160$ instances. Each arc has three cost segments where the cost of each segment is calculated as ratio of the nominal arc cost. This gives segment costs as $c_a^s = c_a \cdot r_s$ where $r \in \{1.00, 0.90, 0.75\}$. The breakpoints are derived from the load points while each of intercept is calculated from the breakpoint and segment's slope.

3.4.2 Results for RNCEP with Fixed-Charge Cost

Single- and Double-Scenario Results

Table 3.3 summarizes the results of the 480 problem instances, reporting the *proportion of instances that were not solved to optimality within the time limit*. We can see the opti-

Table 3.4: Single-scenario results (rounded to one decimal).

		Janos26	Janos39	Sun27	Node39
Optimality gap	$P = 0$	0.0%	0.0%	0.0%	0.0%
	$P = 10$	0.0%	0.0%	0.0%	7.7±2.9%
	$P = 100$	0.0%	0.3±0.6%	5.0±2.8%	51.9±4.8%
Solution time	$P = 0$	6.5±0.5	156.9±17.0	0.3±0.1	536.4±82.2
	$P = 10$	7.4±0.6	227.1±86.0	201.7±201.4	4,000.1±0.0
	$P = 100$	10.8±2.1	3,120.9±1,088.0	3,694.8±815.9	4,000.1±0.1
Capacity added	$P = 0$	268,698±23,970	331,864±57,041	3,043±271	1,194±357
	$P = 10$	270,931±23,195	329,330±54,751	2,925±412	1,204±281
	$P = 100$	275,409±23,476	321,808±53,261	3,652±447	1,167±357

Table 3.5: Double-scenario results (rounded to one decimal).

		Janos26	Janos39	Sun27	Node39
Optimality gap	$P = 0$	0.0%	0.0%	0.0%	0.0%
	$P = 10$	0.0%	0.0%	0.1±0.2%	11.0±1.8%
	$P = 100$	0.0%	1.3±0.5%	10.8±1.4%	57.1±3.3%
Solution time	$P = 0$	88.4±25.1	1,285.6±349.5	1.2±0.2	2,256.6±317.9
	$P = 10$	92.2±21.0	2,373.9±770.5	1,729.0±1,418.2	4,000.2±0.1
	$P = 100$	189.0±57.7	4,000.3±0.2	4,000.1±0.1	4,000.2±0.1
Capacity added	$P = 0$	278,358±8,988	363,225±26,348	4,399±304	1,185±154
	$P = 10$	278,031±7,857	367,324±18,522	4,635±329	1,286±254
	$P = 100$	282,467±9,830	368,547±19,887	5,668±503	1,236±254

mization performance of problem instances in total, for different values of P, and for different number of scenarios. This performance measure gives a high-level summary of the hardness of particular instances. We can conclude that the instances become harder to solve as P increases, and/or as the number of scenarios increases.

Other performance metrics are presented in more detail in Table 3.4 and Table 3.5, where each cell gives an average and standard deviation from a sample of 20 problem instances. *Optimality gap* refers to the sub-optimality estimated and reported by Gurobi using the built-in procedure for lower-bounding the objective. *Solution time* is the time reported by Gurobi, capped by the time limit. *Capacity added* is the overall network capacity added on top of the original capacity (which can be calculated as $|\mathcal{A}|u_a$ from Table 3.1).

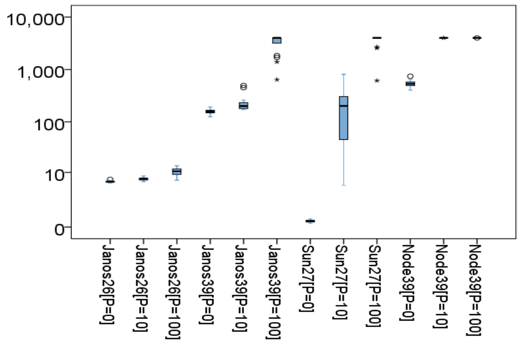
Interestingly, network Sun27 shows large variability in solution time, for both single-scenario and double-scenario settings. While with $P = 0$ it is the quickest to solve out of all networks, for larger values of P it is roughly similar to Janos39, despite dealing with a smaller number of commodities. On the other hand, solution time of Janos26 is affected very little by different values of P. Because of highly non-symmetric distributions of solution time, we present these in boxplots at logarithmic scale in Figure 3.5 and Figure 3.6, for the single- and double-scenario settings, respectively, which provide a deeper insight into this performance measure. The single-scenario boxplots show large variability, including outliers, for Janos39 and Sun27 with $P = 10$ and $P = 100$. This variability explains why the optimality gap of Sun27 is much higher in case $P = 100$ than that of Janos39, despite only a small proportion of instances is reported to be solved to optimality.

Comparing the solution time reported in Table 3.4 and Table 3.5, the double-scenario model, as expected, takes longer to solve to optimality as the goal here is to factor in robustness into the solution. On average, this double-scenario model resulted in 7.39% additional capacity across the networks for instances that were solved to optimality. The increase in solution time is illustrated in the boxplots in Figure 3.5 and Figure 3.6, and the average across the networks for instances that were solved to optimality is 828.24%.

It is also interesting to note that capacity added is highly network dependent. The capacity of Janos26 and Janos39 is expanded dramatically due to the high variability in the demand, which for some commodities significantly exceeds the original capacity (see Table 3.1). On the other hand, the demands in Sun27 and Node39 are small compared to the original capacity, so the capacity added is relatively small.

Not reported elsewhere is the effect of scenario variability λ: the solution time becomes smaller if the uncertainty is larger, i.e., on the average for all the networks and parameter settings, the $0.6\widehat{d}$ variability results in lower solution times than for the $0.3\widehat{d}$ variability. This was also found to be the trend when looking at single networks. This is summarized in Table 3.6.

Overall, it is possible to solve most of the problem instances to optimality within



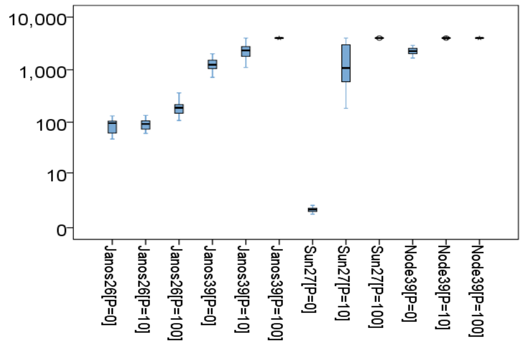


Figure 3.5: Single-scenario solution times. Figure 3.6: Double-scenario solution times.

the time limit, and even most of those not solved to optimality report very small optimality gap. The only settings that would significantly benefit from an increased time limit are Sun27 at $P = 100$ and Node39 at $P = 10$ and $P = 100$. The solution time on the average increases with the total number of variables the solver has to deal with for the different instances. The solver was able to solve problems with up to approximately $350{,}000$ variables to optimality within the time limit.

Effect of Number of Scenarios

While the previous discussion focused only on single- and double-scenario instances, it is also of interest to understand how an increased number of scenarios, which leads to a more robust solution by better covering the demand uncertainty, affects the performance measures. Considering more scenarios is expected to lead to a solution which in practical terms guarantees the network ability to accommodate a higher level of demand variation, increases the network resilience, provides additional capacity, in-

Table 3.6: Effect of higher λ on solution time.

Solution Time	**Single Scenario**	**Double Scenario**
$\lambda = 0.3\widehat{d}$	527.31	3,010.85
$\lambda = 0.6\widehat{d}$	346.62	2,299.23
% Improvement	34.3%	23.6%

Table 3.7: Results on network Janos26 with fixed-charge cost ($P = 10$) for different numbers of scenarios.

# Scenarios	Cost	ΔCost	Added Capacity	ΔAdded Capacity	Time (sec.)	∝Time
1 (optimistic)	83,001,184	-10.9%	192,610	-9.2%	8	1x
1 (expected)	93,115,940	—	212,104	—	8	—
2	127,483,886	36.9%	292,893	38.1%	59	8x
3	129,804,380	39.4%	298,131	40.6%	376	50x
4	130,264,516	39.9%	300,426	41.6%	768	102x
5	130,271,550	39.9%	300,492	41.7%	1,080	143x
6	130,461,776	40.1%	300,913	41.9%	3,124	413x
7	130,753,454	40.4%	301,598	42.2%	2,488	329x
8	131,206,186	40.9%	301,936	42.4%	4,456	589x
9	131,254,563	41.0%	301,715	42.2%	8,869	1173x
1 (pessimistic)	200,592,832	115.4%	456,182	115.1%	8	1x

creases availability, and improves efficiency and overall quality of the network.

To illustrate that, we have tested network Janos26 with fixed charge $P = 10$. We have started with a single-scenario instance, where the base scenario considered reflects the *expected* demand (this is the original demand from SNDLib). We have then generated and gradually added additional scenarios by randomly perturbating all the demands of the base scenario within $\pm\lambda$, in the large uncertainty setting ($\lambda := 0.6\widehat{d}$).

For comparison, we have also considered the *optimistic* instance, which is a single-scenario instance in which the demand is generated by subtracting λ from the expected demand on every arc. This instance expands the capacity of the network to satisfy only the smallest demand scenario, and would be almost surely unable to satisfy the realized demand. Finally, we have considered the *pessimistic* instance, which is a single-scenario instance in which the demand is generated by adding λ to the expected demand on every arc. This instance expands the capacity of the network to satisfy all the possible demand scenarios.

The results are presented in Table 3.7. These results are representative; similar results were obtained when we replicated the experiment with other randomly generated scenarios. The key observations are as follows:

- By gradually expanding the set of scenarios, the cost (our minimization objective) non-decreases;

- The added capacity follows a similar trend, but is not necessarily monotone (cf. 8 vs 9 scenarios);
- The solution time (reported in seconds and as a multiple of the expected scenario instance) increases exponentially;
- Expansion by adding more scenarios approximately follows the law of diminishing returns in both the cost and added capacity: the increase is highest when expanding from 1 (expected) scenario to 2 scenarios (which includes the expected scenario and one randomly generated), with only a minor increase when considering more than 3 scenarios, indicating the value of considering a robust optimization approach even with few scenarios;
- The increase in both the cost and added capacity is dramatic (36.9%) when expanding from 1 (expected) scenario to 2 scenarios (which includes the expected scenario and one randomly generated), indicating that optimizing the network based on the expected scenario (i.e. on point forecasts) only may be an inappropriate approach, leading to a large amount of unsatisfied realized demand;
- Optimizing the network for the pessimistic scenario is very expensive (the increase in both the cost and added capacity is about 115% compared to the expected scenario), indicating the value of considering a robust optimization approach even with few scenarios;
- Optimizing the network for the optimistic scenario leads to savings (the decrease in both the cost and added capacity is about 10% compared to the expected scenario), but may not be acceptable in practice if the consequences of having practically no satisfied realized demand are non-negligible;

These results provide an indication of the ability of our model to become more robust by including more demand scenarios. The solution time of solving our model

Table 3.8: Solution results for piecewise-linear cost.

Single-Scenario	**Sun27**	**Janos26**	**Janos39**	**Node39**
Optimality Gap	0.00%	2.90%	10.43%	22.43%
Solution time	653.67±640.84	4000.22±0.11	4000.22±0.06	4000.16±0.04
Capacity Added	2,863±539	276,172±26,036	335,258±58,895	1,472±574
Double-Scenario				
Optimality Gap	1.43%	6.73%	37.44%	77.99%
Solution time	4000.04±0.01	4000.21±0.23	4000.10±0.03	4000.12±0.04
Capacity Added	4,380±278	296,354±11,398	472,889±110,491	4,117±2,601

with 20 scenarios using Gurobi to the same precision would likely be of a few weeks. We note that Gurobi was able to deal with up to approximately 200 scenarios for this network without giving an out-of-memory error, however, it would be unlikely to compute a close-to-optimal solution in a reasonable amount of time.

3.4.3 Results for RNCEP with Piecewise-Linear Costs

Next we consider the robust network capacity expansion problem with piecewise-linear costs. The experimental result summary is presented in Figure 3.7, showing the proportions of instances solved to optimality (Optimal), not solved to optimality within the time limit but returning a non-optimal solution (Non-Optimal), and not solved to optimality within the time limit but returning no solution due to still being in the root relaxation phase (Root Relaxation). The results indicate that solving problem instances with piecewise-linear costs is significantly more difficult compared to using fixed-charge costs.

Overall, 12.5% of all problem instances were solved to optimality within the time limit, 77.5% returned a non-optimal solution, and 10% were timed out already during the root relaxation. None of the double-scenario problem instances reached optimality within the time limit. Only one of the networks, Sun27, reached optimality and this was for all the problem instances in the single-scenario case. Two networks, Janos39 and Node39, had instances timing out under the root relaxation phase.

Table 3.8 presents more detailed results of this model for each network. The opti-

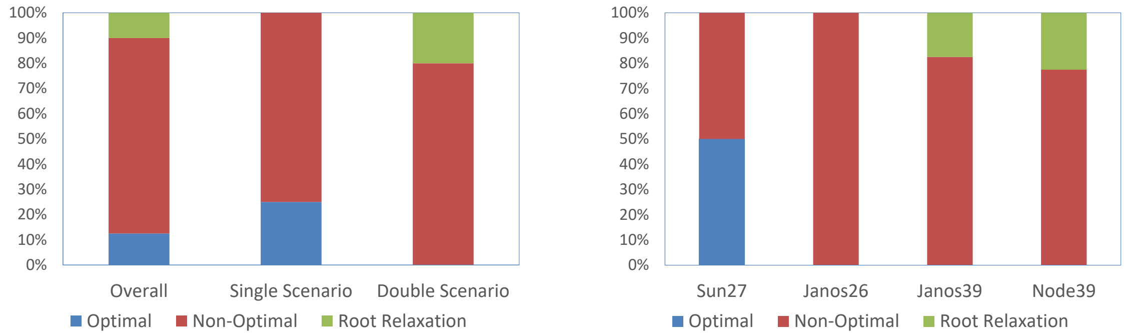


Figure 3.7: Solution results summary for piecewise-linear cost

mality gap is further illustrated in Figure 3.8, Figure 3.10, and Figure 3.11, indicating that the optimality gap may be acceptable because of small values and small variability for Sun27 and Janos26 in the single-scenario setting and for Sun27 in the double-scenario setting. Better solutions can of course be achieved by increasing the time limit, which would be recommendable in the remaining settings.

The optimality gap provides insight into the increased difficulty of solving these problem instances, which also translates into longer solution time. It takes at least 512% more time to solve the double-scenario models compared to the single-scenario using Sun27 network, which is the easiest setting considering its very low optimality gap of 1.43% for the double-scenario instances. The boxplot in Figure 3.9 considers network Sun27 and suggests that there is a much higher variability in solution time if we consider more variable demand scenarios. A further analysis was performed on the solution time using the paired sample t-Test which indicates no significant difference between solution time returned by $0.3\widehat{d}$ and $0.6\widehat{d}$ with a t-statistic of -0.2047 and a p-value 0.8423.

3.5 Conclusions

In this paper, a robust approach to network capacity expansion with non-linear cost functions was investigated. We developed robust models with fixed-charge costs and

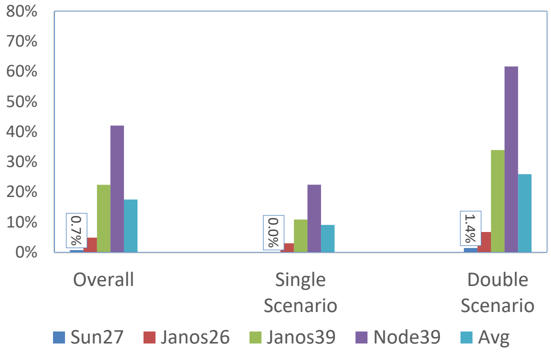


Figure 3.8: Optimality gap for piecewise-linear cost.

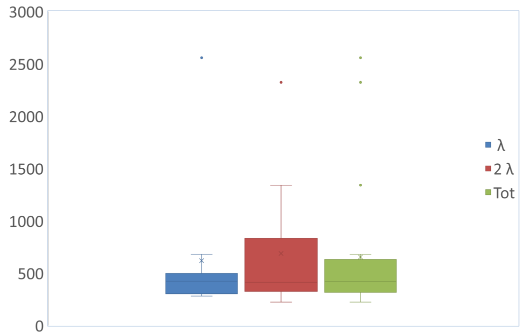


Figure 3.9: Sun27 Network Solution Time

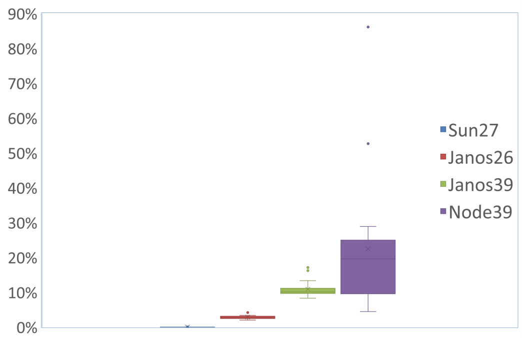


Figure 3.10: Single-scenario Optimality gap for piecewise-linear cost.

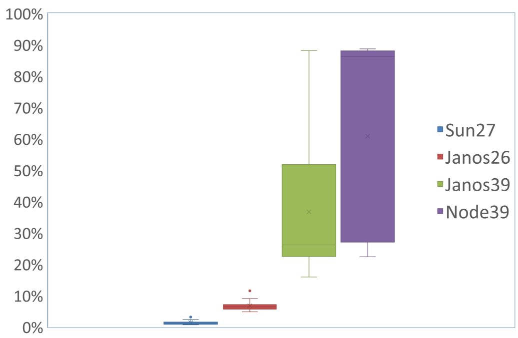


Figure 3.11: Double-scenario Optimality gap for piecewise-linear cost.

with piecewise-linear costs. They were implemented on four networks taken from the SNDlib, [108], with results compared to using linear costs. In the experimental setup, a number of possible parameter configurations was considered, including different demand variability and fixed-charges.

The fixed-charge cost model was able to solve problem instances with up to approximately $300{,}000$ variables within the time limit of $4000s$, and is able to handle problem instances with over 9 million variables, corresponding to 200 demand scenarios for Janos26, memory-wise. The number of variables associated with most complex network Node39 can also explain the reason for the inability to solve to optimality within the time limit, especially in the double-scenario setting. Similarly, the piecewise-linear cost solution reaching optimality within the time limit also depends on the number of variables, which is affected by the network complexity and the number of cost segments.

When further increasing the number of scenearios, we have found that results follow a law a diminishing returns. While objective values and added capacity change little beyond five scenarios, computation times increase considerably. This is an indicator that already few scenarios suffice to find solutions that are robust against uncertainty in demand. The next pursuit will be to further improve the solution time for these models by developing specialised algorithms.

And God said, Let there be light... Gen 1:3

Appendix

3.6 Notation

$\mathcal{V}$	Set of nodes
$\mathcal{A}$	Set of arcs
$\mathcal{K}$	Set of commodities
u_a	Original capacity on arc a
δ_v^-	Set of incoming arcs at node v
δ_v^+	Set of outgoing arcs from node v
s^k	Origin node for commodity k
t^k	Destination node for commodity k
d^k	Demand for commodity k

Table 3.9: Common notation

c_a	Cost per unit of added capacity on arc a
p_a	Fixed charge associated with arc a
h_a	Binary (design) variable that selects arc a to include in the solution
f_a^k	Flow variable of commodity k on arc a
x_a	Capacity expansion variable on arc a

Table 3.10: RNCEP with fixed-charge costs parameters and variables

$\mathcal{S}$	Set segments S
b_a^s	Load breakpoints where b_a^{s-1} & b_a^s are the l_b and u_b on segment s
c_a^s	Cost for segment s on arc a which is also the slope for the segment
p_a^s	Fixed cost component of the piecewise function of segment s on arc a
h_a^s	Binary variable that selects the segment where x falls in
f_a^k	Flow variable of commodity k on arc a
x_a^s	Expansion variable on arc a for price segment s

Table 3.11: RNCEP with piecewise-linear costs parameters and variables

The fear of the Lord is the beginning of wisdom...
Prov. 9:10

Chapter 4

A Comparison of Models for Uncertain Network Design

Abstract. To solve a real-world problem, the modeler usually needs to make a trade-off between model complexity and usefulness. This is also true for robust optimization, where a wide range of models for uncertainty, so-called uncertainty sets, have been proposed. However, while these sets have been mainly studied from a theoretical perspective, there is little research comparing different sets regarding their usefulness for a real-world problem.

In this paper we consider a network design problem in a telecommunications context. We need to invest into the infrastructure, such that there is sufficient capacity for future demand which is not known with certainty. There is a penalty for an unsatisfied realized demand, which needs to be outsourced. We consider three approaches to model demand: using a discrete uncertainty set, using a polyhedral uncertainty set, and using the mean of a per-commodity fitted zero-inflated uniform distribution. While the first two models are used as part of a robust optimization setting, the last model represents a simple stochastic optimization setting. We compare these approaches on an efficiency frontier real-world data taken from the online library SNDlib and observe that, contrary to current research trends, robust optimization using the

polyhedral uncertainty set may result in less efficient solutions.

Keywords: network design; robust optimization; optimization in telecommunications

4.1 Introduction

Network design models have found wide application in the planning, design and operations management of transportation, power & energy distribution, supply chain logistic and telecommunications networks. Usually, they are based on mixed-integer programming models, and many such models have been developed over the decades for network design and expansion problems, see, e.g., [93, 100, 22].

In telecommunications for instance, network design models can be used to curb congestion and to provide an acceptable quality of service to the subscribers. Effort to provide an acceptable service has resulted in capital expenditure of billions of USD in global telecoms investment. Optimization of investments has thus attained a key strategic role in this industry. Moreover, these decisions need to be made well ahead of time based on a forecast of future traffic demand.

Unfortunately, traffic demand has proven to be difficult to predict accurately. In order to factor in this uncertainty and design a network that is immune to traffic variability, robust optimization approaches have been proposed. For this purpose, a number of uncertainty models have already been developed and investigated (see [74, 15, 23]). The drawback of classic approaches, however, is that the uncertainty set is assumed to be given, i.e., the decision maker can advise us how the uncertainty is shaped. Moreover, an inappropriate choice of uncertainty set may result in models that are too conservative or in some cases computationally intractable. As the decision maker cannot be expected to make this choice in practice, data-driven and learning approaches have been recently proposed (see [29, 41]).

To the best of our knowledge, we follow this approach for the first time for network design problems, by comparing which uncertainty set actually fits real-world data. We compare two robust optimization approaches for a network capacity expansion model with outsourceable demand (see, e.g., [22, 11]). In this setting, we need to invest into the network infrastructure now, so that each commodity can be routed to satisfy its uncertain demand later. Demand which cannot be satisfied is outsourced, which is modeled through a linear penalty on its amount.

The two approaches under consideration are (1) a discrete uncertainty set which assumes that all demands are in closed form; and (2) a polyhedral set with wider range of possible scenarios which results in a heuristic mix-integer program to solve the resulting robust problem. These two are compared on real-world data taken from SNDlib and also compared with performance of a third model outside the robust framework, a simple stochastic optimization approach.

The rest of this paper is organized as follows. Section 4.2 presents a literature review of related research. In Section 4.3, we introduce the problem description of robust network capacity expansion with outsourcing and mathematical models for both the discrete and polyhedral uncertainty sets with detailed construction of the robust counterparts. Experimental results and main findings using data from the SNDlib (see [108]) are discussed in Section 4.4. Finally, Section 4.5 concludes our work and points out future research directions.

4.2 Literature Review

The study of uncertainties in decision problems has resulted in two broad areas of research, namely *stochastic* (see, e.g., [36]) and *robust* (see, e.g., [15]) optimization frameworks. While the stochastic approach usually assumes that a probability distribution of the uncertain data is known with precision, the robust approach assumes that the uncertain data lies within a predetermined set. The renewed interest in the latter can be attributed to the works of [17] and [73] with many other collaborators.

The two frameworks also have a dynamic context, where a part of the decision has to be made after the realization of the uncertain data. This is known as two-stage stochastic and robust optimization. Depending on the context, two-stage robust problems are also known as adjustable robust counterparts (ARC). Here, the decision variables are partitioned into two sets: the non-adjustable ones (”here and now” decisions) which must be fixed in advance before the realization of the uncertainty sets and the adjustable ones (”wait and see” decisions), which are computed after the uncertain parameters are revealed ([16]).

As the ARC is more representative of real life situations where decisions are made over multiple periods, this framework has attracted interest from the research community. However, its general form is known to be computationally intractable, which has led to an approximate model using affine decision rules *(ADR)*. In this affine adjustable robust counterpart *(AARC)*, the adjustable part of the decision is assumed to be an affine linear function of the uncertain data ([16]). This emulates a linear feedback as a controller to adjust for the desired output.

Just like in many other fields, robust optimization has found increasing use and application in the network design area. [4] considered a two-stage robust network flow problem under demand uncertainty following the work of [16], while [109] introduced affine routing in the their robust network capacity planning model. [107] looked at network capacity expansion under both demand and cost uncertainty. [89] considered a robust network design problem with static routing in the setting of [32]. [114] apply the AARC to robust network design with polyhedral uncertainty and [6] used a refined version of ADR in their robust capacity assignment for networks with uncertain demand. Recently, [111] used a cutting plane algorithm while taking into consideration the uncertainty in unmet demand outsourced cost.

Regarding uncertainty sets, polyhedral sets are most frequently used in radio network design, along with hose models from the works of [61, 64], budget uncertainty by [4] and cardinal constrained uncertainty by [32], and interval uncertainty among others.

Little research compares these models. [4] compared their single-stage robust model using budget uncertainty with a scenario-based two-stage stochastic approach. [41] constructed different uncertainty sets from real world data and compared performance within and outside sample for shortest path problems. Our focus is to compare the discrete and the polyhedral uncertainty sets in network capacity expansion, to arrive at which one better fits real-world data, while also comparing to the performance of a simple stochastic model using the mean demand.

4.3 Problem Description

We consider a multi-commodity network flow design problem where incremental capacities are installed in response to uncertain traffic demand. The problem is modeled in a way that allows for capacity expansion such that routing of traffic for the different commodities over the arcs subject to design and network constraints is possible while minimizing the total cost involved. We refer to this model as the robust network capacity expansion problem *(RNCEP)*.

4.3.1 The Basic RNCEP

The network under consideration can be represented by a directed graph, $G = (\mathcal{V}, \mathcal{A})$. Each of the arcs $a \in \mathcal{A}$ has an original capacity u_a. The original capacity on each arc a can be upgraded at a cost c_a per each additional unit x_a of capacity. There is a set of commodities $\mathcal{K} = \{1, \dots, K\} =: [K]$ which need to be routed across the network, each commodity $k \in \mathcal{K}$ consisting of a demand $d^k \geq 0$, a source node $s^k \in \mathcal{V}$, and a sink node $t^k \in \mathcal{V}$. Additionally, let σ be the cost of not satisfying one unit of demand over the planning horizon (i.e., by outsourcing it). If all demands are known, the nominal network capacity expansion problem can then be formulated as follows:

$$\min \sum_{a\in\mathcal{A}} c_a x_a + \sigma \sum_{k\in\mathcal{K}} \left[d_k - \sum_{a\in\delta^-(t^k)} f_a^k + \sum_{a\in\delta^+(t^k)} f_a^k \right]_+ \tag{4.1}$$

$$\text{s.t.} \sum_{a\in\delta^-(v)} f_a^k - \sum_{a\in\delta^+(v)} f_a^k \geq 0 \qquad \forall k \in \mathcal{K}, v \in \mathcal{V} \setminus \{s^k, t^k\} \tag{4.2}$$

$$\sum_{k\in\mathcal{K}} f_a^k \leq u_a + x_a \qquad \forall a \in \mathcal{A} \tag{4.3}$$

$$f_a^k \geq 0 \qquad \forall k \in \mathcal{K}, a \in \mathcal{A} \tag{4.4}$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{4.5}$$

Here, $[y]_+$ denotes $\max\{0, y\}$, while $\delta^+(v)$ and $\delta^-(v)$ are the sets of the outgoing and incoming arc at node $v \in \mathcal{V}$, respectively. Variables f_a^k denote the flow of commodity $k \in \mathcal{K}$ along edge $a \in \mathcal{A}$, while x_a models the amount of capacity being added to arc a. The objective function (4.1) is to minimize the sum of capacity expansion cost and outsourcing costs. Constraints (4.2) are a variant of flow constraints, where we allow an arbitrary amount of flow to leave the source node s^k. Through the objective, only the flow arriving in t^k is counted. It is allowed to diminish the flow outside of s^k and t^k; note that there is an optimal solution where this does not happen. We do not assume equality in Constraints (4.2) to apply our robust optimization approach in the following section. Finally, Constraints (4.3) model the capacity on each edge.

The actual demand values $\boldsymbol{d}$ are uncertain, and can take any value in a predetermined uncertainty set $\mathcal{U}$. The two sets under consideration in this work are the *discrete uncertainty set*, which can be represented as $\mathcal{U} = \{\boldsymbol{d}^1, \ldots, \boldsymbol{d}^N\}$, and the *polyhedral uncertainty set*, which can be represented as $\mathcal{U} = \left\{\boldsymbol{d} \in \mathbb{R}_+^K : V\boldsymbol{d} \leq \boldsymbol{b}, d_k \in [\underline{d}_k, \overline{d}_k]\right\}$.

The robust network capacity expansion problem then is to find a minimum installation cost of additional capacities while satisfying all potential traffic demands such that actual flows do not exceed cumulative link capacities whatever the realization of demands in $\mathcal{U}$. Thus, the RNCEP is a two stage robust problem with recourse applying the general framework of [16]. The capacity expansion represented by variables $\boldsymbol{x}$ is the first stage decision variable which has to be fixed before the realization of $\boldsymbol{d} \in \mathcal{U}$. Once the uncertain demand data is revealed, the traffic adjustment takes place by routing a multi-commodity flow with second stage variable $f_a^k(\boldsymbol{d})$. This can be modeled as follows:

$$\min \sum_{a\in\mathcal{A}} c_a x_a + \max_{\boldsymbol{d}\in\mathcal{U}} \sigma \sum_{k\in\mathcal{K}} \left[d_k - \sum_{a\in\delta^-(t^k)} f_a^k(\boldsymbol{d}) + \sum_{a\in\delta^+(t^k)} f_a^k(\boldsymbol{d}) \right]_+ \tag{4.6}$$

$$\text{s.t.} \sum_{a\in\delta^-(v)} f_a^k(\boldsymbol{d}) - \sum_{a\in\delta^+(v)} f_a^k(\boldsymbol{d}) \geq 0 \qquad \forall k \in \mathcal{K}, \boldsymbol{d} \in \mathcal{U}, v \in \mathcal{V} \setminus \{s^k, t^k\} \tag{4.7}$$

$$\sum_{k\in\mathcal{K}} f_a^k(\boldsymbol{d}) \leq u_a + x_a \qquad \forall \boldsymbol{d} \in \mathcal{U}, a \in \mathcal{A} \tag{4.8}$$

$$f_a^k(\boldsymbol{d}) \geq 0 \qquad \forall k \in \mathcal{K}, \boldsymbol{d} \in \mathcal{U}, a \in \mathcal{A} \tag{4.9}$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{4.10}$$

Here, we have modified Constraints (4.1-4.5) to take all scenarios into account. Being a robust model, we consider the worst-case costs in Objective (4.6), while all constraints need to hold for all scenarios $\boldsymbol{d} \in \mathcal{U}$. In the following, we reformulate the general model (4.6-4.10) for specific uncertainty sets.

4.3.2 Robust Optimization with Discrete Uncertainty

Model

Let $\mathcal{U} = \{\boldsymbol{d}^1, \ldots, \boldsymbol{d}^N\}$ be a discrete uncertainty set, where N is the number of scenarios. In this case, variables $f_a^k(\boldsymbol{d})$ become $f_a^{k,i}$ for all $i \in [N]$. The robust objective function (4.6) is reformulated using additional variables $h^{k,i} := [d_k^i - \sum_{a\in\delta^-(t^k)} f_a^{k,i} +$

$\sum_{a\in\delta^+(t^k)} f_a^{k,i}]_+$ for $k \in \mathcal{K}$, $i \in [N]$, and $\tau := \max_{i\in[N]} \sum_{k\in\mathcal{K}} h^{k,i}$. The problem then becomes:

$$\min \sum_{a\in\mathcal{A}} c_a x_a + \sigma\tau \tag{4.11}$$

$$\text{s.t. } \tau \geq \sum_{k\in\mathcal{K}} h^{k,i} \qquad \forall i \in [N] \tag{4.12}$$

$$h^{k,i} \geq d_k^i - \sum_{a\in\delta^-(t^k)} f_a^{k,i} + \sum_{a\in\delta^+(t^k)} f_a^{k,i} \qquad \forall i \in [N], k \in \mathcal{K} \tag{4.13}$$

$$\sum_{a\in\delta^-(v)} f_a^{k,i} - \sum_{a\in\delta^+(v)} f_a^{k,i} \geq 0 \qquad \forall k \in \mathcal{K}, i \in [N], v \in \mathcal{V} \setminus \{s^k, t^k\} \tag{4.14}$$

$$\sum_{k\in\mathcal{K}} f_a^{k,i} \leq u_a + x_a \qquad \forall i \in [N], a \in \mathcal{A} \tag{4.15}$$

$$f_a^{k,i} \geq 0 \qquad \forall k \in \mathcal{K}, i \in [N], a \in \mathcal{A} \tag{4.16}$$

$$h^{k,i} \geq 0 \qquad \forall k \in \mathcal{K}, i \in [N] \tag{4.17}$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{4.18}$$

Here, Constraints (4.14) and (4.15) correspond to Constraints (4.7) and (4.8), whereas the additional Constraints (4.12) and (4.13) are used to ensure variables τ and $h^{k,i}$ have the intended effect. Note that, as we minimize, the maximum operator can be expressed by using $\geq$-constraints over the set.

Constructing Data-Based Discrete Uncertainty

To construct discrete uncertainties uncertainties, we assume that scenarios

$$\mathcal{R} = \{\boldsymbol{r}^1, \ldots, \boldsymbol{r}^N\}$$

of real demands with $\boldsymbol{r}^i \in \mathbb{R}_+^K$ are given, along with the respective source and sink nodes. The trivial approach would be to use directly $\mathcal{U} = \mathcal{R}$. However, previous research (see [41]) has shown that this may result in an overfitting to the available data. Instead, we consider different scalings. For a fixed commodity $k \in \mathcal{K}$, let $N' \leq N$ denote the absolute frequency that $r^{i,k} > 0$ over all $i \in [N]$. Then

$$\hat{r}^k = \frac{1}{N'} \sum_{i \in [N]} r^{i,k}$$

be the average of the demand scenarios for each $k \in \mathcal{K}$. For a given $\lambda \in [0,1]$, we set $d^{i,k}(\lambda) = \lambda r^{i,k} + (1-\lambda)\hat{r}^k$ and

$$\mathcal{U}(\lambda) = \left\{ \boldsymbol{d}^1(\lambda), \dots, \boldsymbol{d}^N(\lambda) \right\}.$$

The case $\lambda = 0$ means that we ignore uncertainty and use the average case, while $\lambda = 1$ uses the original demand scenarios $\mathcal{R}$.

4.3.3 Robust Optimization with Polyhedral Uncertainty

Model

We now assume the demand uncertainty is given through a general polyhedron of the form below as opposed to any special polyhedral type like the Γ-uncertainty,

$$\mathcal{U} = \left\{ \boldsymbol{d} \in \mathbb{R}_+^K : V\boldsymbol{d} \leq \boldsymbol{b}, d_k \in [\underline{d}_k, \overline{d}_k] \right\}$$

where $V = (v_{ik})$ is a matrix in $\mathbb{R}^{M \times K}$ and $\boldsymbol{b}$ is a vector in $\mathbb{R}^M$ (i.e., there are M linear constraints on the demand vector). To find a tractable robust counterpart, we apply the framework of affine decision rules (ADR) by restricting the flow variables to be affine functions of the uncertainty, i.e.,

$$f_a^k(\boldsymbol{d}) = \phi_a^k + \sum_{\ell \in \mathcal{K}} \Phi_a^{k,\ell} d_\ell$$

with ϕ_a^k and $\Phi_a^{k,\ell}$ being unknown coefficients of the affine linear function in $\boldsymbol{d}$. We now consider each constraint and the objective of problem (4.6-4.10) and reformulate them using strong duality.

By substituting for $f_a^k(\boldsymbol{d})$, the flow constraints (4.7) become:

$$\sum_{a\in\delta^-(v)}\left(\phi_a^k+\sum_{\ell\in\mathcal{K}}\Phi_a^{k,\ell}d_\ell\right)-\sum_{a\in\delta^+(v)}\left(\phi_a^k+\sum_{\ell\in\mathcal{K}}\Phi_a^{k,\ell}d_\ell\right)\geq 0 \qquad \forall k\in\mathcal{K}, v\in\mathcal{V}\backslash\{s^k,t^k\}, \boldsymbol{d}\in\mathcal{U},$$

which is equivalent to

$$\sum_{a\in\delta^-(v)}\phi_a^k-\sum_{a\in\delta^+(v)}\phi_a^k\geq\sum_{\ell\in\mathcal{K}}\left(\sum_{a\in\delta^+(v)}\Phi_a^{k,\ell}-\sum_{a\in\delta^-(v)}\Phi_a^{k,\ell}\right)d_\ell \qquad \forall k\in\mathcal{K}, v\in\mathcal{V}\backslash\{s^k,t^k\}, \boldsymbol{d}\in\mathcal{U}. \tag{4.19}$$

For each $k\in\mathcal{K}, v\in\mathcal{V}\setminus\{s^k,t^k\}$ we can write the worst-case problem as

$$\begin{aligned}
\max\ & \sum_{\ell\in\mathcal{K}}\left(\sum_{a\in\delta^+(v)}\Phi_a^{k,\ell}-\sum_{a\in\delta^-(v)}\Phi_a^{k,\ell}\right)d_\ell \\
\text{s.t. } & \sum_{\ell\in\mathcal{K}}v_{i\ell}d_\ell\leq b_i && \forall i\in[M] && [\alpha_i^{k,v}] \\
& d_\ell\leq\overline{d}_\ell && \forall \ell\in\mathcal{K} && [\overline{\beta}_\ell^{k,v}] \\
& -d_\ell\leq-\underline{d}_\ell && \forall \ell\in\mathcal{K} && [\underline{\beta}_\ell^{k,v}]
\end{aligned}$$

We now consider the dual of this linear optimization problem. In brackets behind every constraint of the primal problem, we have listed the corresponding dual variable. The dual problem then becomes

$$\begin{aligned}
\min\ & \sum_{i\in[M]}b_i\alpha_i^{k,v}+\sum_{\ell\in\mathcal{K}}(\overline{d}_l\overline{\beta}_\ell^{k,v}-\underline{d}_\ell\underline{\beta}_\ell^{k,v}) \\
\text{s.t. } & \sum_{i\in[M]}v_{i\ell}\alpha_i^{k,v}+\overline{\beta}_\ell^{k,v}-\underline{\beta}_\ell^{k,v}\geq\sum_{a\in\delta^+(v)}\Phi_a^{k,\ell}-\sum_{a\in\delta^-(v)}\Phi_a^{k,\ell} && \forall \ell\in\mathcal{K} \\
& \alpha_i^{k,v}\geq 0 && \forall i\in[M]
\end{aligned}$$

$$\overline{\beta}_\ell^{k,v} \geq 0 \qquad \forall \ell \in \mathcal{K}$$
$$\underline{\beta}_\ell^{k,v} \geq 0 \qquad \forall \ell \in \mathcal{K}.$$

By applying strong duality, we can conclude that the optimal objective value of this dual problem is equal to the worst-case of the right-hand side of Constraint (4.8).

Overall, **Constraint** (4.7) is replaced by the following set of constraints and variables:

$$\sum_{a\in\delta^-(v)} \phi_a^k - \sum_{a\in\delta^+(v)} \phi_a^k \geq \sum_{i\in[M]} b_i \alpha_i^{k,v} + \sum_{\ell\in\mathcal{K}} (\overline{d}_l \overline{\beta}_\ell^{k,v} - \underline{d}_\ell \underline{\beta}_\ell^{k,v}) \qquad \forall k \in \mathcal{K}, v \in \mathcal{V}\setminus\{s^k, t^k\}$$
$$\sum_{i\in[M]} v_{i\ell}\alpha_i^{k,v} + \overline{\beta}_\ell^{k,v} - \underline{\beta}_\ell^{k,v} \geq \sum_{a\in\delta^+(v)} \Phi_a^{k,\ell} - \sum_{a\in\delta^-(v)} \Phi_a^{k,\ell} \qquad \forall k,\ell \in \mathcal{K}, v \in \mathcal{V}\setminus\{s^k, t^k\}$$
$$\alpha_i^{k,v} \geq 0 \qquad \forall i \in [M], k \in \mathcal{K}, v \in \mathcal{V}\setminus\{s^k, t^k\}$$
$$\overline{\beta}_\ell^{k,v} \geq 0 \qquad \forall k,\ell \in \mathcal{K}, v \in \mathcal{V}\setminus\{s^k, t^k\}$$
$$\underline{\beta}_\ell^{k,v} \geq 0 \qquad \forall k,\ell \in \mathcal{K}, v \in \mathcal{V}\setminus\{s^k, t^k\}$$

We follow a similar procedure for the other constraints. Constraint (4.8) can be rewritten as

$$\sum_{k\in\mathcal{K}} \left(\phi_a^k + \sum_{\ell\in\mathcal{K}} \Phi_a^{k,\ell} d_\ell \right) \leq u_a + x_a \qquad \forall d \in \mathcal{U}, a \in \mathcal{A}$$

The subproblem

$$\begin{aligned} \max \ & \sum_{\ell\in\mathcal{K}} (\sum_{k\in\mathcal{K}} \Phi_a^{k,\ell}) d_\ell \\ \text{s.t. } & \boldsymbol{d} \in \mathcal{U} \end{aligned}$$

has the same structure as before. Using dual variables $\pi_i^a, \overline{\rho}_\ell^a, \underline{\rho}_\ell^a$, we can replace **Constraint** (4.8) with the following:

$$\sum_{k\in\mathcal{K}} \phi_a^k + \sum_{i\in[M]} b_i \pi_i^a + \sum_{\ell\in\mathcal{K}} (\overline{d}_\ell \overline{\rho}_\ell^a - \underline{d}_\ell \underline{\rho}_\ell^a) \leq u_a + x_a \qquad \forall a \in \mathcal{A}$$

$$\begin{aligned}
&\sum_{i\in[M]} v_{i\ell}\pi_i^a + \overline{\rho}_\ell^a - \underline{\rho}_\ell^a \geq \sum_{k\in\mathcal{K}} \Phi_a^{k,\ell} && \forall \ell\in\mathcal{K}, a\in\mathcal{A}\\
&\pi_i^a \geq 0 && \forall i\in[M], a\in\mathcal{A}\\
&\overline{\rho}_\ell^a \geq 0 && \forall a\in\mathcal{A}, \ell\in\mathcal{K}\\
&\underline{\rho}_\ell^a \geq 0 && \forall a\in\mathcal{A}, \ell\in\mathcal{K}
\end{aligned}$$

We now consider the positivity constraint (4.9). This becomes

$$\phi_a^k + \sum_{\ell\in\mathcal{K}} \Phi_a^{k,\ell} d_\ell \geq 0 \qquad \forall k\in\mathcal{K}, a\in\mathcal{A}, \boldsymbol{d}\in\mathcal{U}$$

Using duality with variables $\xi_i^{k,a}, \overline{\zeta}_\ell^{k,a}, \underline{\zeta}_\ell^{k,a}$ we replace **Constraint** (4.9) with the following:

$$\begin{aligned}
&\phi_a^k \geq \sum_{i\in[M]} b_i\xi_i^{k,a} + \sum_{\ell\in\mathcal{K}} (\overline{d}_\ell\overline{\zeta}_\ell^{k,a} - \underline{d}_\ell\underline{\zeta}_\ell^{k,a}) && \forall k\in\mathcal{K}, a\in\mathcal{A}\\
&\sum_{i\in[M]} v_{i\ell}\xi_i^{k,a} + \overline{\zeta}_\ell^{k,a} - \underline{\zeta}_\ell^{k,a} \geq -\Phi_a^{k,\ell} && \forall k,\ell\in\mathcal{K}, a\in\mathcal{A}\\
&\xi_i^{k,a} \geq 0 && \forall k\in\mathcal{K}, a\in\mathcal{A}, i\in[M]\\
&\overline{\zeta}_\ell^{k,a} \geq 0 && \forall k,\ell\in\mathcal{K}, a\in\mathcal{A}\\
&\underline{\zeta}_\ell^{k,a} \geq 0 && \forall k,\ell\in\mathcal{K}, a\in\mathcal{A}
\end{aligned}$$

Finally, we consider the objective function (4.6). We need to solve the following problem:

$$\begin{aligned}
\max\ & \sum_{k\in\mathcal{K}} \left[d_k - \sum_{a\in\delta^-(t^k)} \left(\phi_a^k + \sum_{\ell\in\mathcal{K}} \Phi_a^{k,\ell} d_\ell\right) + \sum_{a\in\delta^+(t^k)} \left(\phi_a^k + \sum_{\ell\in\mathcal{K}} \Phi_a^{k,\ell} d_\ell\right)\right]_+ \\
\text{s.t. } & \sum_{\ell\in\mathcal{K}} v_{i\ell} d_\ell \leq b_i && \forall i\in[M]\\
& d_\ell \leq \overline{d}_\ell && \forall \ell\in\mathcal{K}\\
& -d_\ell \leq -\underline{d}_\ell && \forall \ell\in\mathcal{K}
\end{aligned}$$

We introduce new variables $z_k \in \{0,1\}$ to remove the positivity bracket from the objective.

$$\begin{aligned}
\max \sum_{k\in\mathcal{K}} & \left(d_k - \sum_{a\in\delta^-(t^k)} \left(\phi_a^k + \sum_{\ell\in\mathcal{K}} \Phi_a^{k,\ell} d_\ell \right) + \sum_{a\in\delta^+(t^k)} \left(\phi_a^k + \sum_{\ell\in\mathcal{K}} \Phi_a^{k,\ell} d_\ell \right) \right) z_k \\
\text{s.t. } & \sum_{\ell\in\mathcal{K}} v_{i\ell} d_\ell \le b_i && \forall i \in [M] \\
& d_\ell \le \overline{d}_\ell && \forall \ell \in \mathcal{K} \\
& - d_\ell \le -\underline{d}_\ell && \forall \ell \in \mathcal{K} \\
& z_k \in \{0,1\} && \forall k \in \mathcal{K}
\end{aligned}$$

We set $z'_{k,\ell} := d_\ell z_k$ and get

$$\begin{aligned}
\max \sum_{k\in\mathcal{K}} & \left(z'_{kk} - \sum_{a\in\delta^-(t^k)} \left(\phi_a^k z_k + \sum_{\ell\in\mathcal{K}} \Phi_a^{k,\ell} z'_{k\ell} \right) + \sum_{a\in\delta^+(t^k)} \left(\phi_a^k z_k + \sum_{\ell\in\mathcal{K}} \Phi_a^{k,\ell} z'_{k\ell} \right) \right) \\
\text{s.t. } & \sum_{\ell\in\mathcal{K}} v_{i\ell} d_\ell \le b_i && \forall i \in [M] && [\mathfrak{q}_i] \\
& z'_{k\ell} \le d_\ell && \forall k,\ell \in \mathcal{K} && [\mathfrak{r}_{k\ell}] \\
& z'_{k\ell} \le \overline{d}_\ell z_k && \forall k,\ell \in \mathcal{K} && [\mathfrak{s}_{k\ell}] \\
& d_\ell + \overline{d}_\ell z_k - z'_{k\ell} \le \overline{d}_\ell && \forall k,\ell \in \mathcal{K} && [\mathfrak{t}_{k\ell}] \\
& d_\ell \le \overline{d}_\ell && \forall \ell \in \mathcal{K} && [\mathfrak{u}_\ell] \\
& - d_\ell \le -\underline{d}_\ell && \forall \ell \in \mathcal{K} && [\mathfrak{v}_\ell] \\
& z_k \in \{0,1\} && \forall k \in \mathcal{K} && [\mathfrak{w}_k] \\
& z'_{k\ell} \ge 0 && \forall k,\ell \in \mathcal{K}
\end{aligned}$$

By relaxing constraints $z_k \in \{0,1\}$ to $z_k \in [0,1]$ for a conservative approximation and dualizing the problem, we arrive at

$$\min \sum_{i\in[M]} b_i \mathfrak{q}_i + \sum_{k\in\mathcal{K}} \sum_{\ell\in\mathcal{K}} \overline{d}_\ell \mathfrak{t}_{k\ell} + \sum_{\ell\in\mathcal{K}} \overline{d}_\ell \mathfrak{u}_\ell - \sum_{\ell\in\mathcal{K}} \underline{d}_\ell \mathfrak{v}_\ell + \sum_{k\in\mathcal{K}} \mathfrak{w}_k$$

$$
\begin{aligned}
\text{s.t.} \sum_{i\in[M]} v_{i\ell}\mathfrak{q}_i - \sum_{k\in\mathcal{K}} \mathfrak{r}_{k\ell} + \sum_{k\in\mathcal{K}} \mathfrak{t}_{k\ell} + \mathfrak{u}_\ell - \mathfrak{v}_\ell \ge 0 & \qquad \forall \ell \in \mathcal{K} \\
-\sum_{\ell\in\mathcal{K}} \overline{d}_\ell \mathfrak{s}_{k\ell} + \sum_{\ell\in\mathcal{K}} \overline{d}_\ell \mathfrak{t}_{k\ell} + \mathfrak{w}_k \ge \sum_{a\in\delta^+(t^k)} \phi_a^k - \sum_{a\in\delta^-(t^k)} \phi_a^k & \qquad \forall k \in \mathcal{K} \\
\mathfrak{r}_{k\ell} + \mathfrak{s}_{k\ell} - \mathfrak{t}_{k\ell} \ge 1_{k=\ell} + \sum_{a\in\delta^+(t^k)} \Phi_a^{k,\ell} - \sum_{a\in\delta^-(t^k)} \Phi_a^{k,\ell} & \qquad \forall k,\ell \in \mathcal{K} \\
\mathfrak{q}_i \ge 0 & \qquad \forall i \in [M] \\
\mathfrak{r}_{k\ell}, \mathfrak{s}_{k\ell}, \mathfrak{t}_{k\ell} \ge 0 & \qquad \forall k,\ell \in \mathcal{K} \\
\mathfrak{u}_\ell, \mathfrak{v}_\ell, \mathfrak{w}_\ell \ge 0 & \qquad \forall \ell \in \mathcal{K}
\end{aligned}
$$

Overall, we get the following affine adjustable robust counterpart to Problem (4.6-4.10):

$$
\begin{aligned}
\min \sum_{a\in\mathcal{A}} c_a x_a + \sigma \left(\sum_{i\in[M]} b_i \mathfrak{q}_i + \sum_{k\in\mathcal{K}} \sum_{\ell\in\mathcal{K}} \overline{d}_\ell \mathfrak{t}_{k\ell} + \sum_{\ell\in\mathcal{K}} \overline{d}_\ell \mathfrak{u}_\ell - \sum_{\ell\in\mathcal{K}} \underline{d}_\ell \mathfrak{v}_\ell + \sum_{k\in\mathcal{K}} \mathfrak{w}_k \right) & \\
\text{s.t.} \sum_{i\in[M]} v_{i\ell}\mathfrak{q}_i - \sum_{k\in\mathcal{K}} \mathfrak{r}_{k\ell} + \sum_{k\in\mathcal{K}} \mathfrak{t}_{k\ell} + \mathfrak{u}_\ell - \mathfrak{v}_\ell \ge 0 & \qquad \forall \ell \in \mathcal{K} \\
-\sum_{\ell\in\mathcal{K}} \overline{d}_\ell \mathfrak{s}_{k\ell} + \sum_{\ell\in\mathcal{K}} \overline{d}_\ell \mathfrak{t}_{k\ell} + \mathfrak{w}_k \ge \sum_{a\in\delta^+(t^k)} \phi_a^k - \sum_{a\in\delta^-(t^k)} \phi_a^k & \qquad \forall k \in \mathcal{K} \\
\mathfrak{r}_{k\ell} + \mathfrak{s}_{k\ell} - \mathfrak{t}_{k\ell} \ge 1_{k=\ell} + \sum_{a\in\delta^+(t^k)} \Phi_a^{k,\ell} - \sum_{a\in\delta^-(t^k)} \Phi_a^{k,\ell} & \qquad \forall k,\ell \in \mathcal{K} \\
\sum_{a\in\delta^-(v)} \phi_a^k - \sum_{a\in\delta^+(v)} \phi_a^k \ge \sum_{i\in[M]} b_i \alpha_i^{k,v} + \sum_{\ell\in\mathcal{K}} (\overline{d}_l \overline{\beta}_\ell^{k,v} - \underline{d}_\ell \underline{\beta}_\ell^{k,v}) & \qquad \forall k \in \mathcal{K}, v \in \mathcal{V} \setminus \{s^k, t^k\} \\
\sum_{i\in[M]} v_{i\ell}\alpha_i^{k,v} + \overline{\beta}_\ell^{k,v} - \underline{\beta}_\ell^{k,v} \ge \sum_{a\in\delta^+(v)} \Phi_a^{k,\ell} - \sum_{a\in\delta^-(v)} \Phi_a^{k,\ell} & \qquad \forall k,\ell \in \mathcal{K}, v \in \mathcal{V} \setminus \{s^k, t^k\} \\
\sum_{k\in\mathcal{K}} \phi_a^k + \sum_{i\in[M]} b_i \pi_i^a + \sum_{\ell\in\mathcal{K}} (\overline{d}_\ell \overline{\rho}_\ell^a - \underline{d}_\ell \underline{\rho}_\ell^a) \le u_a + x_a & \qquad \forall a \in \mathcal{A} \\
\sum_{i\in[M]} v_{i\ell}\pi_i^a + \overline{\rho}_\ell^a - \underline{\rho}_\ell^a \ge \sum_{k\in\mathcal{K}} \Phi_a^{k,\ell} & \qquad \forall \ell \in \mathcal{K}, a \in \mathcal{A} \\
\phi_a^k \ge \sum_{i\in[M]} b_i \xi_i^{k,a} + \sum_{\ell\in\mathcal{K}} (\overline{d}_\ell \overline{\zeta}_\ell^{k,a} - \underline{d}_\ell \underline{\zeta}_\ell^{k,a}) & \qquad \forall k \in \mathcal{K}, a \in \mathcal{A} \\
\sum_{i\in[M]} v_{i\ell}\xi_i^{k,a} + \overline{\zeta}_\ell^{k,a} - \underline{\zeta}_\ell^{k,a} \ge -\Phi_a^{k,\ell} & \qquad \forall k,\ell \in \mathcal{K}, a \in \mathcal{A}
\end{aligned}
$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A}$$

$$\mathfrak{q}_i \geq 0 \qquad \forall i \in [M]$$

$$\mathfrak{r}_{k\ell}, \mathfrak{s}_{k\ell}, \mathfrak{t}_{k\ell} \geq 0 \qquad \forall k, \ell \in \mathcal{K}$$

$$\mathfrak{u}_\ell, \mathfrak{v}_\ell, \mathfrak{w}_\ell \geq 0 \qquad \forall \ell \in \mathcal{K}$$

$$\alpha_i^{k,v} \geq 0 \qquad \forall i \in [M], k \in \mathcal{K}, v \in \mathcal{V} \setminus \{s^k, t^k\}$$

$$\overline{\beta}_\ell^{k,v}, \underline{\beta}_\ell^{k,v} \geq 0 \qquad \forall k, \ell \in \mathcal{K}, v \in \mathcal{V} \setminus \{s^k, t^k\}$$

$$\pi_i^a \geq 0 \qquad \forall i \in [M], a \in \mathcal{A}$$

$$\overline{\rho}_\ell^a, \underline{\rho}_\ell^a \geq 0 \qquad \forall a \in \mathcal{A}, \ell \in \mathcal{K}$$

$$\xi_i^{k,a} \geq 0 \qquad \forall k \in \mathcal{K}, a \in \mathcal{A}, i \in [M]$$

$$\overline{\zeta}_\ell^{k,a}, \underline{\zeta}_\ell^{k,a} \geq 0 \qquad \forall k, \ell \in \mathcal{K}, a \in \mathcal{A}$$

Constructing Data-Based Polyhedral Uncertainty

Constructing a polyhedron that contains the demand scenarios $\mathcal{R}$ can be considered as an optimization problem on its own. We would like to determine constraint coefficients $(v_{i1}, \ldots, v_{iK}, b_i)$ that determine a polyhedron $\mathcal{U}$ such that the distance of $\mathcal{R}$ to the boundary of $\mathcal{U}$ with respect to a norm $\|\cdot\|$ is as small as possible.

Recall that the distance between a point $\boldsymbol{p}$ and a hyperplane $(a_1, \ldots, a_K, b)$ is given through

$$\frac{|\sum_{i\in[K]} a_i p_i - b|}{\|\boldsymbol{a}\|^*}$$

where $\|\cdot\|^*$ is the dual norm of $\|\cdot\|$. An optimization model to determine $\mathcal{U}$ is hence:

$$\begin{aligned} \min \ & \sum_{i\in[N]} \min_{j\in[M]} (b_j - \sum_{k\in[K]} r^{i,k} v_{jk}) \\ \text{s.t.} \ & \sum_{k\in[K]} r^{i,k} v_{jk} \leq b_j && \forall i \in [N], j \in [M] \\ & \|\boldsymbol{v}_{j\cdot}\|^* = 1 \end{aligned}$$

where $\boldsymbol{v}_{j\cdot}$ denotes the jth row of V. While such an approach is useful for low-dimensional data (i.e., few commodities K), it is less efficient for high-dimensional data. In fact, the additional lower and upper bounds $[\underline{d}_k, \overline{d}_k]$ may already suffice to determine a polyhedron where every point in $\mathcal{R}$ is on its boundary. Therefore, we also consider randomly generated hyperplanes. To this end, we sample each v_{ik} randomly uniformly from $[0, 1]$. Then we set

$$b_i := \max_{j\in[N]} \sum_{k\in[K]} v_{ik} r^{j,k}$$

to find a tight constraint. In particular, we always contain the sum-constraint where $v_{ik} = 1/K$ for all $k \in [K]$.

4.3.4 Stochastic Optimization with Distribution Mean

Model

Let $\overline{\boldsymbol{d}}$ be the vector of mean demands of distributions fitted independently to every commodity using demand scenarios $\mathcal{R} = \{\boldsymbol{r}^1, \ldots, \boldsymbol{r}^N\}$. We reformulate Problem (4.1-4.5) using only this single mean demand scenario. To linearize the positivity brackets $[\cdot]_+$, we introduce variables h^k for every commodity $k \in \mathcal{K}$. The problem then becomes:

$$\begin{aligned}
\min \; & \sum_{a\in\mathcal{A}} c_a x_a + \sigma \sum_{k\in\mathcal{K}} h^k \\
\text{s.t. } & h^k \geq \overline{d}^k - \sum_{a\in\delta^-(t^k)} f_a^k + \sum_{a\in\delta^+(t^k)} f_a^k && \forall k \in \mathcal{K} \\
& \sum_{a\in\delta^-(v)} f_a^k - \sum_{a\in\delta^+(v)} f_a^k \geq 0 && \forall k \in \mathcal{K}, v \in \mathcal{V} \setminus \{s^k, t^k\} \\
& \sum_{k\in\mathcal{K}} f_a^k \leq u_a + x_a && \forall a \in \mathcal{A} \\
& f_a^k \geq 0 && \forall k \in \mathcal{K}, a \in \mathcal{A} \\
& h^k \geq 0 && \forall k \in \mathcal{K} \\
& x_a \geq 0 && \forall a \in \mathcal{A}
\end{aligned}$$

Generating Data-Based Distribution Mean

The demand for the stochastic optimization model is generated from the demand scenarios $\mathcal{R}$ using the mean of the zero-inflated uniform distribution in the following way. For a fixed commodity $k \in \mathcal{K}$, let $N' \leq N$ denote the absolute frequency that $r^{i,k} > 0$ over all $i \in [N]$. To fit a uniform distribution, set

$$r^k_{min} = \min_{i \in [N]: r^{i,k} > 0} r^{i,k} \text{ and } r^k_{max} = \max_{i \in [N]} r^{i,k}$$

and the mean of the uniform distribution is $\bar{r}^k = 1/2(r^k_{min} + r^k_{max})$. The remaining absolute frequency, $N - N'$, is considered for observing a zero demand, yielding the mean demand of the zero-inflated uniform distribution $\overline{d}^k = \bar{r}^k N'/N$.

4.4 Computational Experiments

4.4.1 Setup

The aim of our experiments is to determine which model gives the best solution to uncertain network design in terms of the measured performance metrics. On the one hand, the discrete uncertainty model is simpler than the polyhedral uncertainty model, and we can expect it to be solvable using more commodities, thus giving a more detailed description of the uncertainty. The polyhedral model on the other hand will use less commodities, but has a more complex description of the uncertainty available. As noted in the literature review (Section 4.2), polyhedral models are popular in current research.

We consider the following experimental setup to address our question. Using a data set of real-world scenarios, we separate it into a training set and an evaluation set. We construct different uncertainty sets only based on the training set, and solve the resulting robust (or stochastic) optimization problems. We then only keep the here-and-now part of the solution, i.e., the decision $\boldsymbol{x}$ on the infrastructure investment. This

Table 4.1: Experimental Setup

Experiment	Nr of σ values	σ values	nr λ values	λ values	Nr of hyperplanes
Discrete_1	2	12,450 and 24,900	11	0.0 to 1.0	
Discrete_2	11	0 to 24,900	2	0.5 and 1.0	
Stochastic	11	0 to 24,900	1	1.0	
Polyhedral_1	11	0 to 24,900			1
Polyhedral_2	8	0 to 17,430			2
Polyhedral_3	7	0 to 14,940			7

investment is then assessed on the evaluation set by calculating optimal flows for each scenario. As the first-stage investment costs are already fixed, the flow problem only aims at minimizing the outsourced demand. We then compare investment costs and outsourced demand for all models.

The experimental setup is summarized in Table 4.1. The Discrete_1 experiment fixes two σ values for varying values of λ, while the Discrete_2 experiment fixes λ for varying values of σ. The Polyhedral_1 experiment uses a polyhedron with only one constraint (the sum-constraint) for all eleven values of σ, Polyhedral_2 uses a polyhedron with two hyperplanes and eight values for σ, while Polyhedral_3 uses a polyhedron with eight hyperplanes and seven possible σ values. The reduced choice for σ values with increasing number of hyperplanes was due to increased computation times.

The Discrete_1 experiment therefore has to solve 22 optimization models, and each of these 22 results was then evaluated on each of the demand scenarios from the evaluation set. The same was carried out for the other five experiments. The choice of σ, which represent the penalty for unmet demand, was a key consideration for these models and hence in the experimental setup. If σ is too small there is incentive for unmet demand where almost all demand are outsourced with no addition of new installed capacity to the network while with a large σ, the incentive is for negative violation of the constraint which encourages the deployment of new network capacity.

Several values of σ were tested in a preliminary experiment using discrete uncertainty, see Table 4.2. Based on the outcomes, the value range for σ was selected, taking the 95^{th} percentile of the capacity cost distribution into account.

Table 4.2: Impact of σ on outsourced demand.

Objective	Commodity	Capacity Add	Sol Time	Outsourced D	Penalty(σ)
530,226.88	400	0.00	199.56	127,254.45	100
5,302,268.77	400	0.00	127.48	127,254.45	1,000
49,191,424.59	400	2,191.83	443.11	99,409.83	10,000
68,676,799.21	400	16,139.27	372.44	13,262.58	20,000
72,391,760.98	400	18,151.57	575.90	6,074.70	30,000
74,113,211.23	400	19,736.41	455.89	2,335.66	40,000
74,479,094.31	400	20,873.46	378.29	0.00	50,000

In total, over $34,000$ numerical experiments were carried out according to the setup. Models were implemented using Julia and Gurobi version 7.5 on a Lenovo desktop machine with 8 GB RAM and Intel Core i5-65 CPU with 2.50GHz using Windows 10 OS 64-bit. In Gurobi, we have used a time limit of $9000s$ for each problem instance and optimality is achieved once the optimality gap is below 0.01%.

4.4.2 Data

We tested the discrete, polyhedral and stochastic models using network data instances taken from the online SNDlib library[1], see [108]. The particular network data considered in this work is Germany-50 with 50 nodes and 176 directed arcs (we also included arcs in opposite directions). There are three levels of aggregation for real-world traffic measurement data available. These are one full day (in 5 minute intervals), one full month (in 1 day intervals) and one whole year (in 1 month intervals).

For our experiment, we focus on the full day dataset, consisting of $N = 288$ scenarios. The peak demand of $7,649.83$ was recorded at 3pm for the demand profile, see Figure 4.1. We separate the scenarios into a training set consisting of 24 scenarios, which is generated by taking every 12th demand scenario (i.e., one scenario per hour), and the evaluation set consisting of the remaining 264 scenarios. We refer to the training set as MS-24.

Each scenario has a different number of commodities, see Figure 4.2. Some of the

[1] See http://sndlib.zib.de

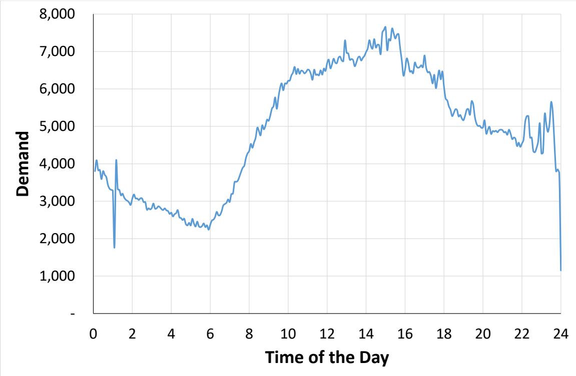


Figure 4.1: A full day demand profile.

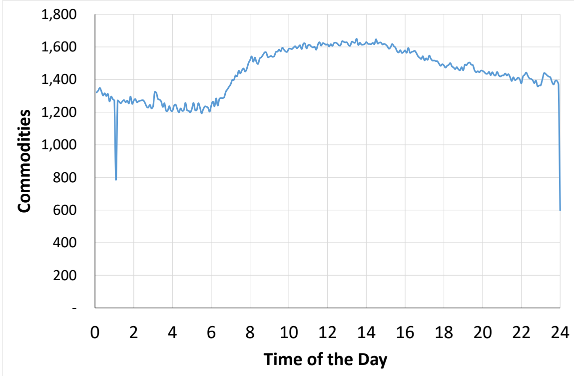


Figure 4.2: A full day commodities profile.

Table 4.3: Impact of choice of K on the presorted original data.

Options with MS-24	Commodity	% of Original Data Captured
All Demand	300	97.48%
All Demand	400	98.88%
All Demand	450	99.25%
All Demand	500	99.50%
All Demand	900	99.88%

demand values were found to be very small. While the 99^{th} percentile of all demand values is 0.415, some values are in the range of 10^{-6}. To simplify the optimization problems, we sort the commodities in descending order of demand and then choose a fixed value of commodities for all demand scenarios that covers over 98% of the original demand data, which is the case for 400 commodities. Table 4.3 shows the different numbers of commodities against the percentage of original data captured in the streamlined data. This approach was implemented instead of allowing for varying commodities per demand scenario and allows us to consider all significant demands values while discarding very low ones, thus significantly reducing the average numbers of commodities per demand scenario.

We observed that a model based on a polyhedron with 400 commodities computed from the training set demand matrix could not be solved in reasonable time, hence the polyhedron was generated for a reduced number of commodities to allow for an optimal solution in a reasonable amount of time that will encourage its practical usage in the industry. Instead, we work with $K = 20$ that captures the top commodities in the training set. This reflects that the more complex the model for the uncertainty, the harder becomes the optimization model itself, and the less data we can use for building our sets. This trade-off is investigated in our experiments. Additionally, our polyhedrons were calcualted using the random constraint sampling method from Section 4.3.3, as lower and upper bounds already gave an optimal solution to the optimization approach for constructing polyhedra.

4.4.3 Computational Results

We consider the performance of the capacity expansion solutions on the evaluation scenarios. We used four metrics on these 264 scenarios: The average, the maximum, the average of the worst 10% (known as conditional-value-at-risk, or CVaR), and the standard deviation. Note that all these measures were calculated for scenarios that were not known to the models at the time of solution.

We first of all note that all polyhedral models Polyhedral_1 to Polyhedral_3 gave the same results, so we do not differentiate between them in the following. In Figure 4.3 to Figure 4.6, the four metrics are shown against the first-stage investment costs for two values of σ (i.e., using Discrete_1). As expected, increasing σ results in building more capacity in the network and hence reducing the amount of demand being outsourced. This is true for both robust the stochastic models. For the discrete uncertainties, network capacity built increases with increasing value of λ from 0 (ignoring uncertainty) to 1 (using the real demands) for a fixed σ value.

In Figure 4.7 to Figure 4.10, varying penalty values σ were considered for the three models while fixing λ for the discrete uncertainty model (using Discrete_2). The outsourced demand τ decreases with an increase in σ value. The implication of higher penalty is that overall risk is minimized deploying additional infrastructure in capacity for the network rather than outsourcing demand. In Figure 4.5 and Figure 4.9, the CVaR was observed to decrease with increasing robustness of the models. Figure 4.4 seems to be providing almost the same information as the CVaR, and it turned out that

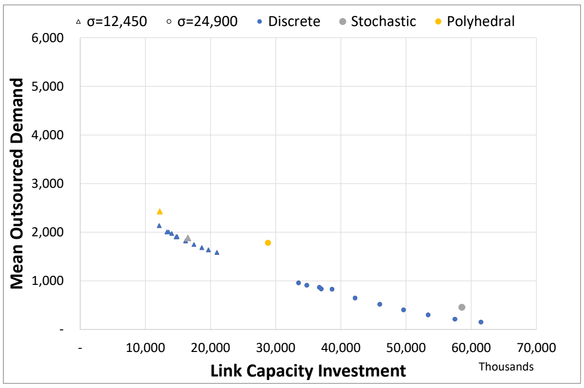


Figure 4.3: Mean outsourced demand. Discrete model uses varying values of λ.

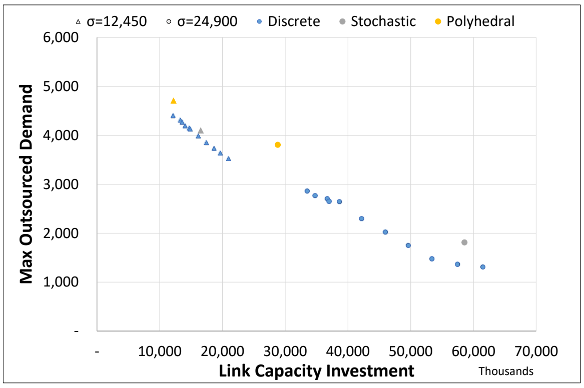


Figure 4.4: Maximum outsourced demand. Discrete model uses varying values of λ.

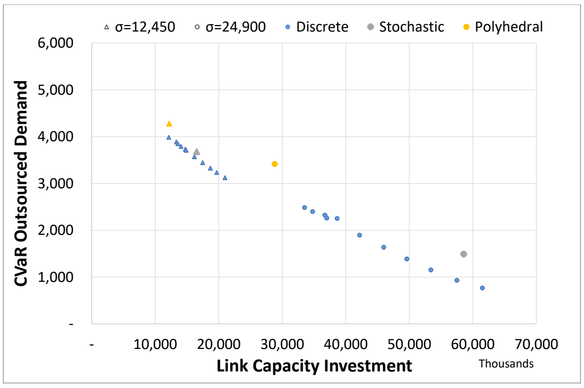


Figure 4.5: CVaR of outsourced demand. Discrete model uses varying values of λ.

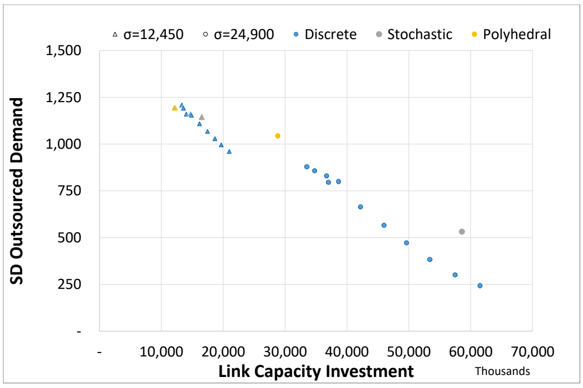


Figure 4.6: Standard deviation of outsourced Demand. Discrete model uses varying values of λ.

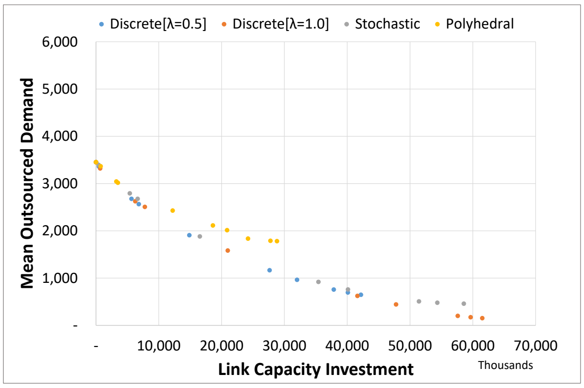


Figure 4.7: Mean outsourced demand. All models use varying values of σ.

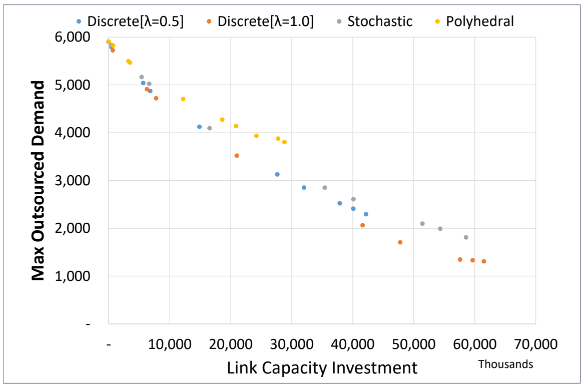


Figure 4.8: Maximum outsourced demand. All models use varying values of σ.

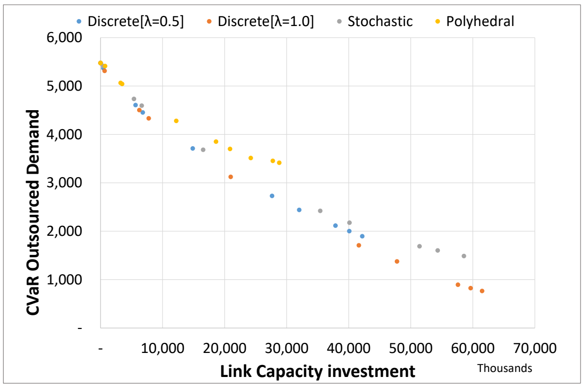


Figure 4.9: CVaR of outsourced demand. All models use varying values of σ.

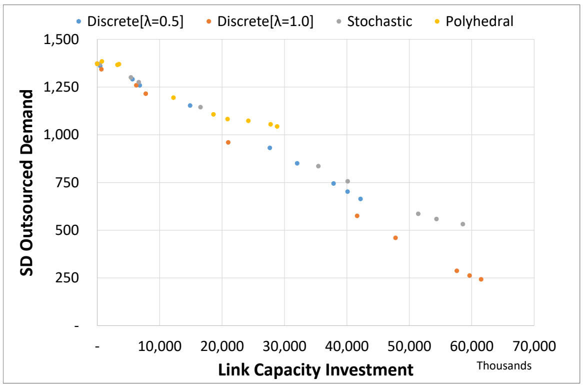


Figure 4.10: Standard deviation of outsourced demand. All models use varying values of σ.

the two metrics are highly correlated having a correlation coefficient of 0.9993 with a gradient of approximately 1 as shown in Figure 4.11. Though the analysis done was for the discrete model, the same result is consistence with that from the other two models.

Ideally, a good solution is in the bottom left corner of these plots. We note that some of the points corresponding to polyhedral models are dominated, and so are the stochastic solutions. The discrete model produces the best trade-off solutions between investment and outsourcing. For instance in Figure 4.9, with the same link capacity investment of $\$40$ million, the stochastic model has a higher CVaR figure. The data point line for this discrete model with $\lambda = 0.5$ is below that for the stochastic model

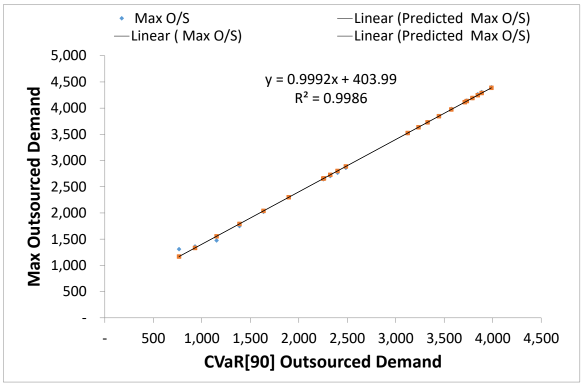


Figure 4.11: CVaR of outsourced demand and max outsourced demand correlation.

and this can bee seen in Figure 4.7 to Figure 4.10. Hence, the discrete model provides the best compromise between a too simple and a too complex approach for the data under consideration.

4.5 Conclusion

In the robust optimization literature, the shape of uncertainty is often an assumption made without any grounding in actually available data. This also holds for network expansion problems, where polyhedral models have been popular. In this paper, we considered the question whether such an approach leads to solutions which perform well on unseen data, i.e., what kind of uncertainty sets are most appropriate for our model.

We developed robust (using discrete and polyhedral uncertainty sets) and stochastic approaches to a multi-commodity network capacity expansion problem with the option of demand outsourcing. These models were implemented for a real-world network data taken from the SNDlib and their results were subsequently compared.

In the experimental setup, a number of penalty values for demand outsourcing were considered while also varying the robustness of the discrete model with different sizes of the uncertainty set. Increasing the penalty results in additional capital expenditure for network capacity build as this reduces the amount of demand outsourced as well as the conditional-value-at-risk (CVaR). However, of these three models, the robust model with discrete uncertainty set produced the best trade-off solutions on all performance metrics. It was also observed that the discrete set seems easy to generate (as expected, since the original data is already in this form), the model is simple and produces optimal result faster. Robust model with polyhedral uncertainty set, on the other hand, is more complex and with more options to describe data, and it results in computationally more challenging problems. In our case, the extra effort associated with polyhedral model may not be really worth it in the end. Surprisingly, the simple stochastic optimization model which we have used for benchmarking was relatively competitive, and thus might be appropriate for use in more complex situations in which the uncertainty-based robust models are computationally intractable.

..the knowledge of the Holy One is understanding.
Prov. 9:10

Chapter 5

A Comparison of Data-Driven Uncertainty Sets for Robust Network Design

Abstract. We consider a network design and expansion problem, where we need to make a capacity investment now, such that uncertain future demand can be satisfied as closely as possible. To use a robust optimization approach, we need to construct an uncertainty set that contains all scenarios that we believe to be possible. In this paper we discuss how to actually construct two common models of uncertainty set, discrete and polyhedral uncertainty, using data-driven techniques on real-world data. We employ clustering to generate a discrete uncertainty set, and supervised learning to generate a polyhedral uncertainty set. We then compare the performance of the resulting robust solutions for these two types of models on real-world data. Our results indicate that polyhedral models, while being popular in the recent literature, are less effective than discrete models both in terms of computational burden and solution quality regardless of the performance measure considered (worst-case, conditional value-at-risk, average).

Keywords: network design; robust optimization; optimization in telecommunications; data-driven optimization; clustering; supervised learning

5.1 Introduction

Operations Research approaches have found wide application in the planning, design and operations management of transportation, power and energy distribution, supply chain logistics and telecommunications networks. In particular, many types of optimization models have been developed over the last decades for network design and expansion problems, see, e.g., [93, 100, 22].

In telecommunications, for instance, network design models can be used to curb congestion and to provide an acceptable quality of service to the subscribers. Effort to provide an acceptable service has resulted in capital expenditure of billions of USD in global telecoms investment. Optimization of investments has thus attained a key strategic role in this industry. Moreover, these decisions need to be made well ahead of time based on a forecast of future traffic demand.

Unfortunately, traffic demand has proven to be difficult to predict accurately. In order to factor in this uncertainty and design a network that is immune to traffic variability, robust optimization approaches have been proposed. For this purpose, a number of uncertainty models have already been developed and investigated (see [74, 15, 23]). The drawback of classic approaches, however, is that the uncertainty set is assumed to be given, i.e., the decision maker can advise on how the uncertainty is shaped. Moreover, an inappropriate choice of uncertainty set may result in models that are too conservative or in some cases computationally intractable. As the decision maker cannot be expected to make this choice in practice, data-driven and learning approaches have been recently proposed (see [29, 41]).

This paper contributes to this recent line of research proposing a clustering approach to generate discrete uncertainty sets from real data viewed as a set of scenarios.

We use the K-means clustering method which results in aggregating similar scenarios into clusters and representing each cluster of scenarios by its centroid, with the intention to reduce the problem complexity on the one hand, and to become less dependent on data noise on the other hand.

The basic network design problem that we consider in this paper is as follows. Given an undirected graph $G = (\mathcal{V}, \mathcal{E})$ and currently installed capacity u_e for each edge $e \in \mathcal{E}$, we would like to determine an amount of capacity x_e to be installed additionally. For each edge e, we are given investment cost c_e per unit of additionally installed capacity. As the graph is undirected, the direction flow is not relevant for our model, and we define $\mathcal{K} = \{\{i,j\} : i,j \in \mathcal{V}, i < j\}$ as the set of commodities, where each commodity k is identified by an unordered pair of nodes $\{i,j\}$ between which a given demand needs to be satisfied. Let d_k be the demand corresponding to commodity $k \in \mathcal{K}$, and let $\mathcal{P}_k$ be the set of simple paths in G connecting the nodes of the commodity. The aim is to find capacities $\boldsymbol{x}$ such that all demands are fulfilled and the capacity expansion costs are as small as possible. Formally, the baseline model can thus be written as follows.

$$\min \sum_{e \in \mathcal{E}} c_e x_e \tag{5.1}$$

$$\text{s.t.} \sum_{p \in \mathcal{P}_k} f_{kp} \geq d_k \qquad \forall k \in \mathcal{K} \tag{5.2}$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}_k : e \in p} f_{kp} \leq u_e + x_e \qquad \forall e \in \mathcal{E} \tag{5.3}$$

$$x_e \geq 0 \qquad \forall e \in \mathcal{E} \tag{5.4}$$

$$f_{kp} \geq 0 \qquad \forall k \in \mathcal{K}, p \in \mathcal{P}_k \tag{5.5}$$

The variables f_{kp} model the amount of flow along path p for commodity k. Here, Constraints (5.2) ensure that a sufficient amount of flow is transported along all paths connecting source and sink of commodity $k \in \mathcal{K}$, while Constraints (5.3) model that each edge needs to provide sufficient capacity. Instead of using a path-based formulation, it is also possible to use a model with flow variables for every edge in the network (see, e.g., [72]). In this paper, we focus on the path-based formulation, as it performed better in our computational experiments.

In practice, the demand $\boldsymbol{d}$ changes over time and is not known precisely. Thus, a two-stage model is required, where we decide now where to build how much capacity (the strategic decision $\boldsymbol{x}$), and we can decide where to route the flow once the demand is known (the operational decision $\boldsymbol{f}$). Let us assume that a set $\mathcal{U}$ can be constructed that contains all demand scenarios $\boldsymbol{d}$ that we would like to take into account for our planning. The two-stage robust network design problem is then to solve

$$
\begin{aligned}
\min\ & \sum_{e\in\mathcal{E}} c_e x_e && && (5.6)\\
\text{s.t.}\ & \sum_{p\in\mathcal{P}_k} f_{kp}(\boldsymbol{d}) \geq d_k && \forall k\in\mathcal{K}, \boldsymbol{d}\in\mathcal{U} && (5.7)\\
& \sum_{k\in\mathcal{K}} \sum_{p\in\mathcal{P}_k : e\in p} f_{kp}(\boldsymbol{d}) \leq u_e + x_e && \forall e\in\mathcal{E}, \boldsymbol{d}\in\mathcal{U} && (5.8)\\
& x_e \geq 0 && \forall e\in\mathcal{E} && (5.9)\\
& f_{kp}(\boldsymbol{d}) \geq 0 && \forall k\in\mathcal{K}, p\in\mathcal{P}_k, \boldsymbol{d}\in\mathcal{U} && (5.10)
\end{aligned}
$$

In this setting, f_{kp} has become a function that depends on the scenario $\boldsymbol{d}$. Note that in Constraint (5.7) d_k is an element of $\boldsymbol{d}$, thus is also scenario-dependent.

Robust optimization in general has found increasing use and application in the network design area. [4] considered a two-stage robust network flow problem under demand uncertainty following the work of [16], while [109] introduced affine routing in the their robust network capacity planning model. [107] looked at network capacity expansion under both demand and cost uncertainty. [89] considered a robust network design problem with static routing in the setting of [32]. [114] considered robust network design with polyhedral uncertainty and [6] robust capacity assignment for networks with uncertain demand. [111] used a cutting plane algorithm while taking into consideration the uncertainty in unmet demand outsourced cost.

Regarding uncertainty sets, polyhedral models are most frequently used in radio network design, along with hose models from the works of [61, 64], budget uncertainty by [4], cardinal constrained uncertainty by [32], and interval uncertainty, among others. Little research compares these models of uncertainty. [4] compared their single-stage robust model using budget uncertainty with a scenario-based two-stage stochastic approach. [41] constructed different uncertainty sets from real world data and compared performance within and outside sample for shortest path problems.

In this paper we present the following contributions:

- We propose and develop a clustering approach (using the well-known K-means clustering data mining method) to generate discrete uncertainty sets from real data for a network design and network expansion problems; We use this approach to calculate the cluster centroids for real-world data taken from SNDlib (see [108]) and use these centroids to define a discrete uncertainty set which is used to compute the optimal network expansion;

- We compare this solution to the solution obtained using the state-of-the-art approach of modelling uncertainty using a polyhedral set, where constraints on the demand are given as hyperplanes generated dynamically using supervised learning (a machine learning method). To the best of our knowledge, this is for the first time that such a comparison is done for network design problems;

- For the real-world dataset we consider, we find in our numerical experiments that solutions based on discrete uncertainty found by clustering outperform solutions based on polyhedral uncertainty found by supervised learning when using high risk-adverse performance metrics such as maximum or $\mathrm{CVaR}_{0.95}$ of unsatisfied demand. This is less clear for less risk-averse metrics such as expected value or $\mathrm{CVaR}_{0.75}$ of unsatisfied demand, but the clustering approach is still superior in most cases. At the same time, solutions based on discrete uncertainty found by clustering can be computed two orders of magnitude faster than those based on polyhedral uncertainty found by supervised learning.

The rest of this paper is organized as follows. As the problem data is the center point of our research, we first discuss this in Section 5.2. In particular, we describe how to construct uncertainty sets $\mathcal{U}$ from the data. We then introduce models for robust network design for both discrete and polyhedral uncertainty sets in Section 5.3. Experimental results are discussed in Section 5.4. Finally, Section 5.5 concludes our work and points out future research directions.

5.2 Problem Data and Uncertainty Set Construction

We focus on the Abilene network based on data from the SNDlib (see [108]). It consists of 12 nodes connected by 15 edges, see Figure 5.1, which spread over the US. With 12 nodes, there exist $12 \cdot 11/2 = 66 =: \kappa$ different commodities.

Data was collected by Yin Zhang[1] in 5 minute intervals between 01.03.2004 and 10.09.2004 with some breaks in between. Table 5.1 shows the number of measurements that are available for each month. Note that one day can give 288 measurements in 5 minute intervals. Based on this number, we also show the maximum number of possible measurements that can be achieved each month, but note that not all data is available.

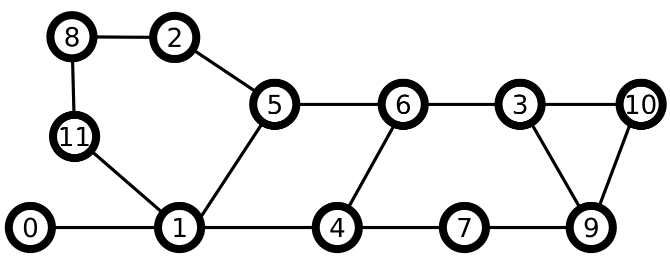


Figure 5.1: Abilene network topology.

We use an arbitrary set of T measurements for the purpose of model training. The rest of the data is then used for the evaluation of results. This means the training data

[1] http://www.cs.utexas.edu/~yzhang/

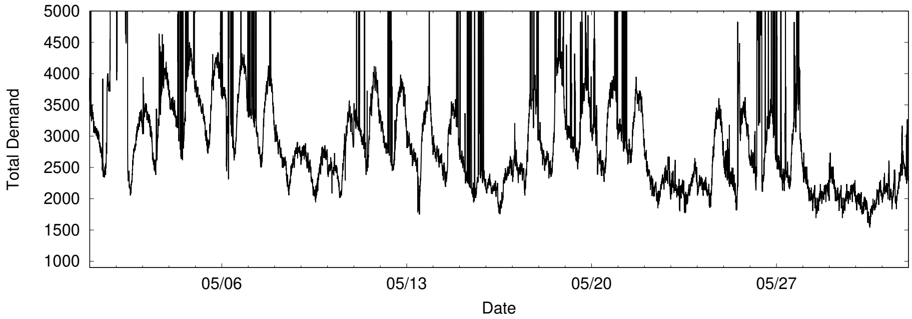
Month	03	04	05	06	07	08	09
# Measurements available	4,032	6,048	8,928	8,640	8,928	8,640	2,880
# Measurements possible	8,928	8,640	8,928	8,640	8,928	8,928	8,640

Table 5.1: Numbers of available measurements for each month.

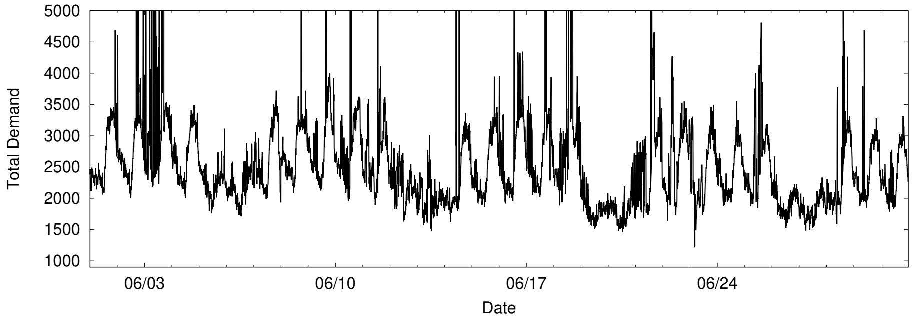
consists of T demand scenarios, each being a κ-dimensional vector of reals. The total demand per scenario, i.e., the sum of demand over all commodities, for months May-August which provide the most complete sets of data measurements, is presented in Figure 5.2.

We now discuss how to generate discrete and polyhedral uncertainty sets based on the training data points. Let $\mathcal{D} = \{\boldsymbol{d}^1, \ldots, \boldsymbol{d}^T\}$ denote this training set. For a discrete uncertainty set $\mathcal{U}^d$, where each scenario is explicitly listed, a natural approach is setting $\mathcal{U}^d = \mathcal{D}$. But it has been shown (see [41]) that this can lead to an overfitting effect, such that the resulting robust solutions do not perform well on out-of-sample data points. Furthermore, it is desirable to control the degree of conservatism. We therefore propose a clustering approach to generate discrete uncertainty sets. We aggregate similar scenarios together, with the intention to reduce the problem complexity on the one hand, and to become less dependent on data noise on the other hand. Scenario aggregation based on K-means clustering has been applied as an approximation method also to robust min-knapsack problems, see [42]. Let $\mathcal{U}^d_K$ denote a discrete uncertainty set derived from a K-means clustering of the set $\mathcal{D}$. Then on the one boundary case, $\mathcal{U}^d_T = \mathcal{D}$, i.e., we contain the original set of training points as a special case, and on the other boundary case, $\mathcal{U}^d_1$ consists of only the average case scenario.

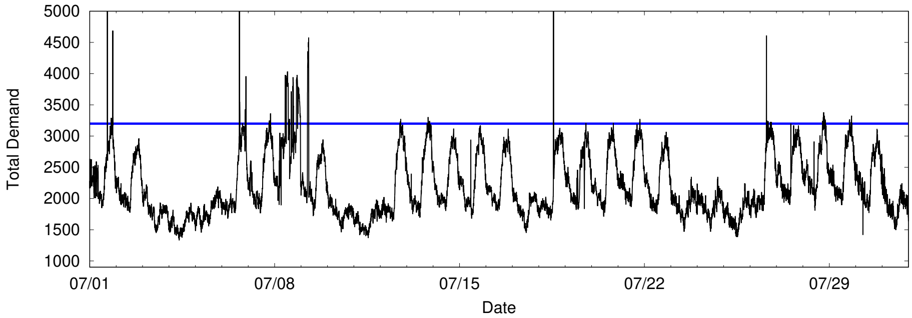
We show a simple example in Figure 5.3. In Figure 5.3a, we plot a subset of the training data, restricted to two arbitrarily chosen dimensions (recall that every commodity corresponds to a dimension of the demand vector). In Figure 5.3c, we show a discrete uncertainty $\mathcal{U}^d_5$ set based on a K-means clustering with $K = 5$ centers, which captures the training data only in a rough manner. With $K = 20$ (see Figure 5.3e), most features of the data have been captured.



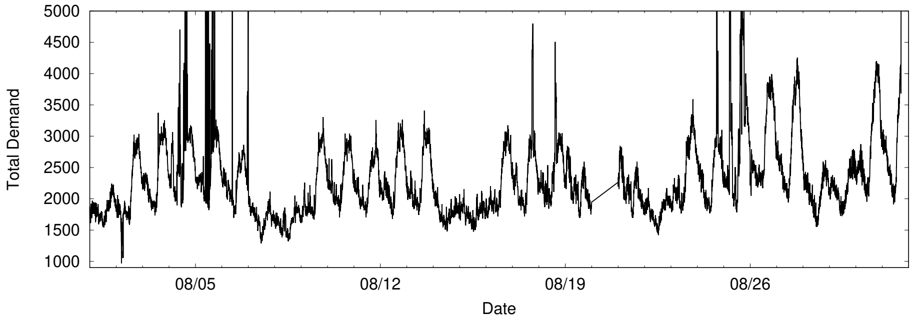
(a) Demand profile for month 05.



(b) Demand profile for month 06.

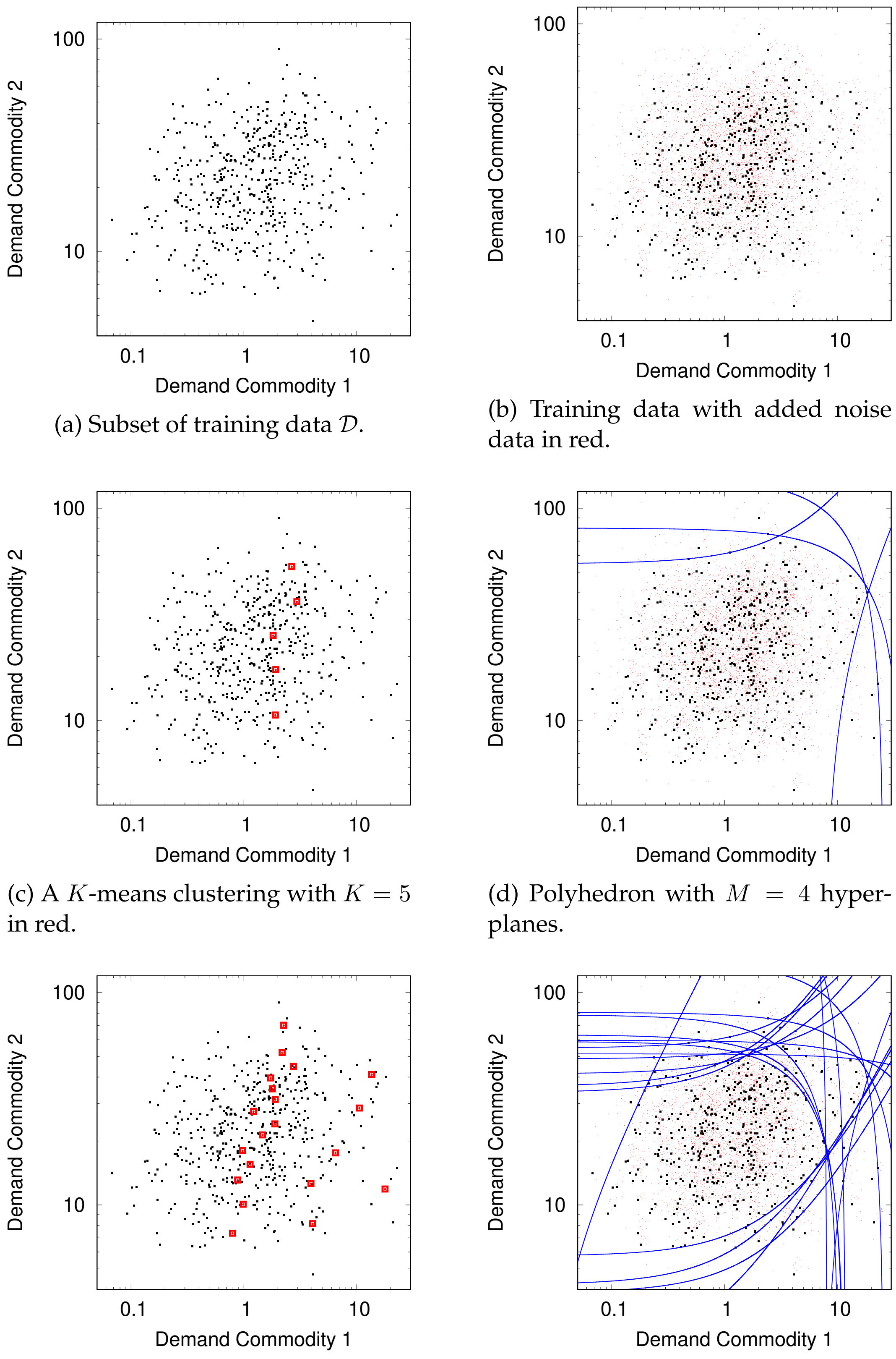


(c) Demand profile for month 07 (training set).



(d) Demand profile for month 08.

Figure 5.2: Total demand in the Abilene network.



(a) Subset of training data $\mathcal{D}$.

(b) Training data with added noise data in red.

(c) A K-means clustering with $K = 5$ in red.

(d) Polyhedron with $M = 4$ hyperplanes.

(e) A K-means clustering with $K = 20$ in red.

(f) Polyhedron with $M = 20$ hyperplanes.

Figure 5.3: Illustration of methods to generate discrete and polyhedral uncertainty on a subset of training data restricted to two dimensions (note the logarithmic scale of the axes).

We now consider the case of polyhedral uncertainty,

$$\mathcal{U}^p_M = \left\{ \boldsymbol{d} \in \mathbb{R}^\kappa_+ : V\boldsymbol{d} \leq \boldsymbol{b}, d_k \in [\underline{d}_k, \overline{d}_k] \right\}$$

where $V = (v_{ik})$ is a matrix in $\mathbb{R}^{M \times \kappa}$ and $\boldsymbol{b}$ is a vector in $\mathbb{R}^\kappa$ (i.e., there are M linear constraints on the demand vector). As the number of constraints M will have a significant impact on the solution time of the resulting robust model, we would like to find only few constraints which describe the training data $\mathcal{D}$ well. To this end, we apply a technique similar to supervised learning in machine learning. We generate a set of noise data points, which we would like to distinguish from the original training data by placing hyperplanes that put as many original points on one side, and as many noise points on the other side as possible. This trade-off is adjusted dynamically: for the first hyperplane, there is a high penalty for original points that are classified as noise. This way, we find an outer description of the data, which results in large and conservative uncertainty sets. This penalty is reduced over time, so that later hyperplanes become less conservative and cut away outliers in the training data. Noise points are generated by randomly increasing values of single training data points, and randomly using values from other data points in single dimensions with low probability.

In Figure 5.3, we use the same data as for the clustering example to illustrate this process. The random noise is shown as red points in Figure 5.3b. The first four hyperplanes we generate are given in Figure 5.3d. It can be seen that they form an outer approximation of the data, removing only few outliers in the process. With an increasing number of hyperplanes M, the polyhedron $\mathcal{U}^p_M$ becomes smaller and less conservative (see Figure 5.3f).

5.3 Robust Models

We now discuss how to reformulate the general model (5.6-5.10) for specific uncertainty sets $\mathcal{U}_K^d$ and $\mathcal{U}_M^p$.

5.3.1 Discrete Uncertainty

Let $\mathcal{U}_K^d = \{\boldsymbol{d}^1, \dots, \boldsymbol{d}^K\}$ be given. As this set is discrete, we can simply write $f_{kp}(\boldsymbol{d}) = f_{kp}^i$ for all $\boldsymbol{d} = \boldsymbol{d}^i \in \mathcal{U}_K^d$. We write $[K] := \{1, \dots, K\}$ in the following. The resulting compact optimization model is then given as follows:

$$\min \sum_{e \in \mathcal{E}} c_e x_e \tag{5.11}$$

$$\text{s.t.} \sum_{p \in \mathcal{P}_k} f_{kp}^i \geq d_k^i \qquad \forall k \in \mathcal{K}, i \in [K] \tag{5.12}$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}_k : e \in p} f_{kp}^i \leq u_e + x_e \qquad \forall e \in \mathcal{E}, i \in [K] \tag{5.13}$$

$$x_e \geq 0 \qquad \forall e \in \mathcal{E} \tag{5.14}$$

$$f_{kp}^i \geq 0 \qquad \forall k \in \mathcal{K}, p \in \mathcal{P}_k, i \in [K] \tag{5.15}$$

5.3.2 Polyhedral Uncertainty

Rewriting $f_{kp}(\boldsymbol{d})$ in a compact form is less straightforward for continuous uncertainty sets than in the previous, discrete case. We apply the well-known affine decision rules (also known as affine adjustable robust counterpart) approach, see [16]. To this end, we restrict $f_{kp}(\boldsymbol{d})$ to be an affine linear function in $\boldsymbol{d}$ by writing

$$f_{kp}(\boldsymbol{d}) = \phi_{kp} + \sum_{\ell \in \mathcal{K}} d_\ell \Phi_{kp\ell}$$

Here, $\phi_{kp} \gtrless 0$ and $\Phi_{kpl} \gtrless 0$ are new decision variables. By using affine decision rules, we restrict the set of feasible solutions, and thus form a conservative approximation to the original problem.

By substituting the $f_{kp}(\boldsymbol{d})$ variables in (5.6-5.10) and rearranging terms, the problem becomes:

$$\min \sum_{e \in \mathcal{E}} c_e x_e \tag{5.16}$$

$$\text{s.t.} \sum_{p \in \mathcal{P}_k} \phi_{kp} \geq \max_{\boldsymbol{d} \in \mathcal{U}_M^p} \sum_{\ell \in \mathcal{K}} \left(1_{\ell = k} - \sum_{p \in \mathcal{P}_k} \Phi_{kp\ell} \right) d_\ell \qquad \forall k \in \mathcal{K} \tag{5.17}$$

$$\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}_k : e \in p} \phi_{kp} + \max_{\boldsymbol{d} \in \mathcal{U}_M^p} \sum_{\ell \in \mathcal{K}} \left(\sum_{k \in \mathcal{K}} \sum_{p \in \mathcal{P}_k : e \in p} \Phi_{kp\ell} \right) d_\ell^e \leq u_e + x_e \qquad \forall e \in \mathcal{E} \tag{5.18}$$

$$\phi_{kp} + \min_{\boldsymbol{d} \in \mathcal{U}_M^p} \sum_{\ell \in \mathcal{K}} \Phi_{kp\ell} d_\ell \geq 0 \qquad \forall k \in \mathcal{K}, p \in \mathcal{P}_k \tag{5.19}$$

$$x_e \geq 0 \qquad \forall e \in \mathcal{E} \tag{5.20}$$

$$\phi_{kp} \gtrless 0 \qquad \forall k \in \mathcal{K}, p \in \mathcal{P}_k \tag{5.21}$$

$$\Phi_{kpl} \gtrless 0 \qquad \forall k \in \mathcal{K}, p \in \mathcal{P}_k, \ell \in \mathcal{K} \tag{5.22}$$

The inner maximization and minimization problems can then be reformulated using linear programming duality. As an example, consider Constraint (5.17) for a fixed $k \in \mathcal{K}$. The value of the right-hand side is

$$\max \sum_{\ell \in \mathcal{K}} \left(1_{\ell = k} - \sum_{p \in \mathcal{P}_k} \Phi_{kp\ell} \right) d_\ell \tag{5.23}$$

$$\text{s.t. } V\boldsymbol{d} \leq \boldsymbol{b} \tag{5.24}$$

$$d_\ell \in [\underline{d}_\ell, \overline{d}_\ell] \qquad \forall \ell \in \mathcal{K} \tag{5.25}$$

As $\mathcal{U}_M^p$ is polyhedral, this is a linear program, the dual of which is

$$\min \sum_{i\in[M]} b_i\alpha_i + \sum_{\ell\in\mathcal{K}} (\overline{d}_\ell \overline{\beta}_\ell - \underline{d}_\ell \underline{\beta}_\ell) \tag{5.26}$$

$$\text{s.t.} \sum_{i\in[M]} v_{i\ell}\alpha_i + \overline{\beta}_\ell - \underline{\beta}_\ell \geq 1_{\ell=k} - \sum_{p\in\mathcal{P}_k} \Phi_{kp\ell} \qquad \forall \ell \in \mathcal{K} \tag{5.27}$$

$$\boldsymbol{\alpha}, \overline{\boldsymbol{\beta}}, \underline{\boldsymbol{\beta}} \geq 0 \tag{5.28}$$

By weak duality, any feasible solution to (5.26-5.28) gives an upper bound on the value of (5.23-5.25). Thus we can substitute the formulation (5.26-5.28) into Constraint (5.17) to reach an equivalent linear reformulation. Repeating this process for all constraints, the robust network extension problem with polyhedral uncertainty can be rewritten in the following way:

$$\begin{aligned}
\min \ & \sum_{e\in\mathcal{E}} c_e x_e \\
\text{s.t.} \ & \sum_{p\in\mathcal{P}_k} \phi_{kp} \geq \sum_{i\in[M]} b_i\alpha_{ki} + \sum_{\ell\in\mathcal{K}} (\overline{d}_\ell \overline{\beta}_{k\ell} - \underline{d}_\ell \underline{\beta}_{k\ell}) && \forall k \in \mathcal{K} \\
& \sum_{i\in[M]} v_{i\ell}\alpha_{ki} + \overline{\beta}_{k\ell} - \underline{\beta}_{k\ell} \geq 1_{\ell=k} - \sum_{p\in\mathcal{P}_k} \Phi_{kp\ell} && \forall k, \ell \in \mathcal{K} \\
& \sum_{k\in\mathcal{K}} \sum_{p\in\mathcal{P}_k : e\in p} \phi_{kp} + \sum_{i\in[M]} b_i\pi_{ei} + \sum_{\ell\in\mathcal{K}} (\overline{d}_\ell \overline{\rho}_{e\ell} - \underline{d}_\ell \underline{\rho}_{e\ell}) \leq u_e + x_e && \forall e \in \mathcal{E} \\
& \sum_{i\in[M]} v_{i\ell}\pi_{ei} + \overline{\rho}_{e\ell} - \underline{\rho}_{e\ell} \geq \sum_{k\in\mathcal{K}} \sum_{p\in\mathcal{P}_k : e\in p} \Phi_{kp\ell} && \forall e \in \mathcal{E}, \ell \in \mathcal{K} \\
& \phi_p^k \geq \sum_{i\in[M]} b_i\xi_{kpi} + \sum_{\ell\in\mathcal{K}} (\overline{d}_\ell \overline{\zeta}_{kp\ell} - \underline{d}_\ell \underline{\zeta}_{kp\ell}) && \forall k \in \mathcal{K}, p \in \mathcal{P}_k \\
& \sum_{i\in[M]} v_{i\ell}\xi_{kpi} + \overline{\zeta}_{kp\ell} - \underline{\zeta}_{kp\ell} \geq -\Phi_{kp\ell} && \forall k, \ell \in \mathcal{K}, p \in \mathcal{P}_k \\
& x_e \geq 0 && \forall e \in \mathcal{E} \\
& \phi_{kp} \gtrless 0 && \forall k \in \mathcal{K}, p \in \mathcal{P}_k \\
& \Phi_{kp\ell} \gtrless 0 && \forall k \in \mathcal{K}, p \in \mathcal{P}_k, \ell \in \mathcal{K} \\
& \alpha_{ki} \geq 0 && \forall i \in [M], k \in \mathcal{K}
\end{aligned}$$

$$\overline{\beta}_{k\ell}, \underline{\beta}_{k\ell} \geq 0 \qquad \forall k, \ell \in \mathcal{K}$$

$$\pi_{ei} \geq 0 \qquad \forall e \in \mathcal{E}, i \in [M]$$

$$\overline{\rho}_{e\ell}, \underline{\rho}_{e\ell} \geq 0 \qquad \forall e \in \mathcal{E}, \ell \in \mathcal{K}$$

$$\xi_{kpi} \geq 0 \qquad \forall k \in \mathcal{K}, p \in \mathcal{P}_k, i \in [M]$$

$$\overline{\zeta}_{kp\ell}, \underline{\zeta}_{kp\ell} \geq 0 \qquad \forall k, \ell \in \mathcal{K}, p \in \mathcal{P}_k$$

5.4 Experiments

5.4.1 Setup

The aim of the experiments is to analyze the performance of solutions to the robust network design problem using discrete and polyhedral uncertainty sets, respectively. We set all existing capacities u_e to be zero, so that the effect of model choice becomes more visible.

Using the data described in Section 5.2, we focus on the four months from beginning of May until end of August which provide the most complete sets of data measurements. We based the training set on an arbitrarily chosen month, 07, which consists of 8,928 demand scenarios, but we removed outlier scenarios, which are defined as the top 2% of scenarios with regard to total demand, leaving us with $T = 8,750$ scenarios in the training set. The corresponding cut-off value is shown in Figure 5.2c as a horizontal blue line.

We calculate solutions based on the training set derived from month 07 measurements and then evaluate them on all the scenarios from the training set and from the three other months (05, 06, 08), minimizing unsatisfied demand. Only the first-stage $\boldsymbol{x}$-part of a solution is used for evaluation.

For discrete uncertainty, we calculate solutions based on clusterings with $K = 100$ up to $K = 8,600$ in steps of 100, and in addition using all $T = 8,750$ training scenarios (a total of 87 optimization problems and solutions). Clusters are calculated using the `kmeans` function of SciPy 1.2.1 under Python 3.7.

For polyhedral uncertainty, we placed hyperplanes using the `dual_annealing` function from SciPy. We generate 140 hyperplanes this way. They are collected in 28 polyhedra, where polyhedron i uses all hyperplanes of polyhedron $i-1$, and five more in the order that they were generated. In total, this means that 28 optimization problems with polyhedral uncertainty are solved.[2]

For a better comparison, the $87+28$ solutions are then also scaled up and down uniformly by multiplying the corresponding $\boldsymbol{x}$ vector with a factor $\lambda = 0.5$ up to $\lambda = 1.5$ (with step size 1/40). The reason to also consider these scaled versions is because, by construction, solutions based on polyhedra will be more conservative than those based on clusterings. By scaling solutions up and down, a more comprehensive comparison becomes possible.

Each of these $(87+28)\cdot 41$ solutions is then evaluated by calculating an optimal flow for each of the 8750 training scenarios and each of the $8928 + 8640 + 8928$ evaluation scenarios. In total, this means that over 166 million linear programs are solved for the evaluation by employing parallelization over ten CPU processors. As there may not be sufficient capacity available to route all demand, we minimize the unsatisfied demand in each optimization problem. The corresponding model to evaluate solutions $\boldsymbol{x}$ for a fixed scenario $\boldsymbol{d}$ is as follows:

$$
\begin{aligned}
\min\ & \sum_{k\in\mathcal{K}} h_k \\
\text{s.t. } & h_k \geq d_k - \sum_{p\in\mathcal{P}_k} f_{kp} && \forall k\in\mathcal{K} \\
& \sum_{k\in\mathcal{K}} \sum_{p\in\mathcal{P}_k : e\in p} f_{kp} \leq u_e + x_e && \forall e\in\mathcal{E} \\
& x_e \geq 0 && \forall e\in\mathcal{E}
\end{aligned}
$$

[2] All linear programs were solved using Cplex 12.8 on a virtual Ubuntu server with ten Xeon CPU E7-2850 processors at 2.00 GHz speed and 23.5 GB RAM using only one core each and spreading the jobs over ten CPU cores.

$$f_{kp} \geq 0 \qquad \forall k \in \mathcal{K}, p \in \mathcal{P}_k$$

where h_k denotes the unsatisfied demand in commodity k. Note that the cost of a solution only depends on the choice of $\boldsymbol{x}$. Additionally, for each month, we calculate four measures to characterize the performance of a solution with regard to unsatisfied demand: average, $\text{CVaR}_{0.75}$ (i.e., the average unsatisfied demand over the 25% largest values), $\text{CVaR}_{0.95}$, and the maximum.

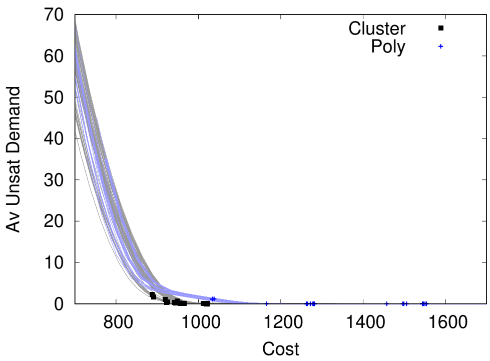
5.4.2 Results

We first discuss the performance of solutions on the training set (month 07) in the left column of Figure 5.4. On the horizontal axis, we show the costs of solutions, while the vertical axis shows the four measures of unsatisfied demand. Each point corresponds to a solution (87 black squares corresponding to the discrete uncertainty solutions, 28 blue crosses corresponding to the polyhedral uncertainty solutions). The lines show the performance of the scaled solutions.

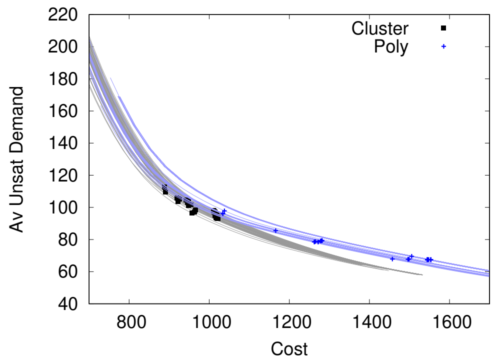
Consider Figure 5.4g. By construction, we know that the discrete uncertainty solution with $\mathcal{U}^d_{8750}$ has zero unsatisfied demand on the training set, and is the cheapest possible solution to do so. Most polyhedral solutions use conservative outer approximations of the training data and thus also have zero unmet demand, but at higher costs. We can also see that solutions that use fewer clusters or more hyperplanes become less conservative, allowing unsatisfied demand at lower solution costs. The behaviour we see in Figure 5.4g is to be expected by design. The open question is whether it can also be observed on evaluation data.

Figures 5.4a, 5.4c and 5.4e show the average, $\text{CVaR}_{0.75}$, and $\text{CVaR}_{0.95}$ performance on the training set, respectively. Here the differences between both types of solution are much less pronounced; we see that blue and black lines overlap, indicating a similar performance of solution types.

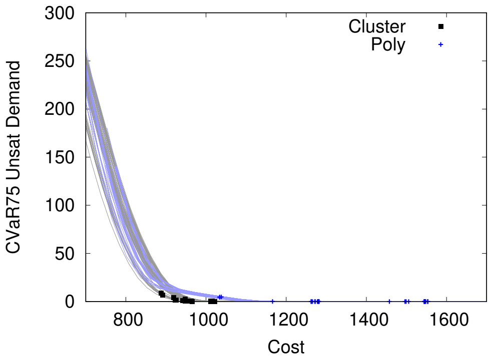
Compare this performance to the right-hand column of Figure 5.4, where the per-



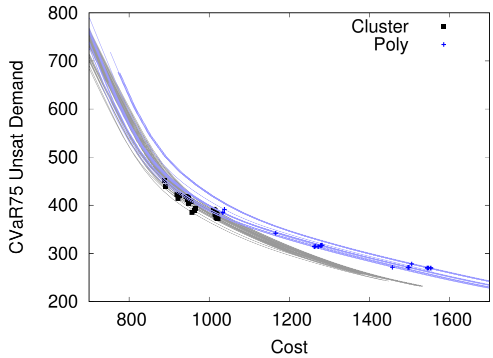
(a) Training set, average values.



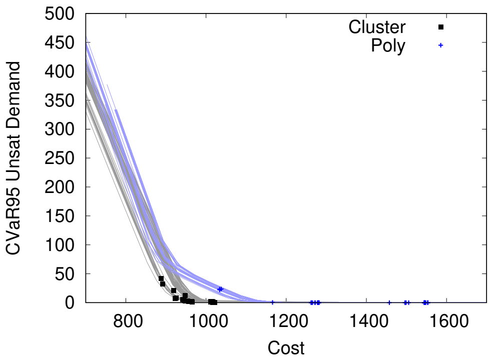
(b) Month 08, average values.



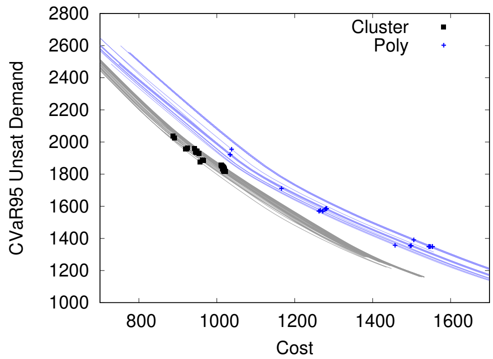
(c) Training set, $\text{CVaR}_{0.75}$ values.



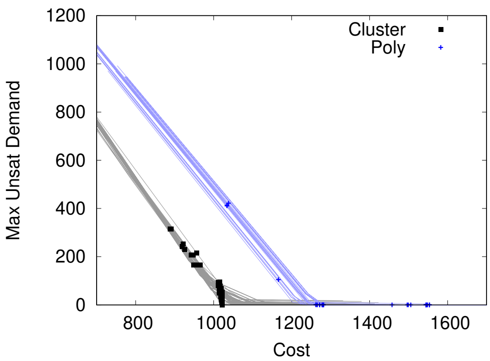
(d) Month 08, $\text{CVaR}_{0.75}$ values.



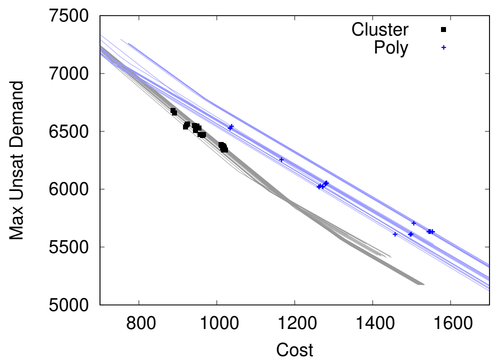
(e) Training set, $\text{CVaR}_{0.95}$ values.



(f) Month 08, $\text{CVaR}_{0.95}$ values.



(g) Training set, maximum values.



(h) Month 08, maximum values.

Figure 5.4: Results for training set and month 08.

formance on month 08 is presented. The order of magnitude of unsatisfied demand has increased for each type of solution: whereas in Figure 5.4g we can reach zero unsatisfied demand, the same solutions have between five and seven thousand units of maximum unmet demand in Figure 5.4h. But the relative performance between the solution types is similar. Whereas solutions based on polyhedral uncertainty generally have a higher degree of robustness at higher investment costs, it is possible to scale solutions based on clustered data up to reach solutions with a similar degree of robustness at lower costs. This is particularly visible for the high risk-adverse measures in Figures 5.4f and 5.4h, whereas these performance differences are less clear-cut for the less risk-adverse measures in Figures 5.4b and 5.4d.

Figure 5.6 in Appendix 5.5 shows the results for months 05 and 06, where the same observations apply as for month 08.

In terms of solution quality, i.e., trade-off between investment costs and unsatisfied demand, we thus find the following result: Solutions based on discrete uncertainty found by clustering outperform solutions based on polyhedral uncertainty found by supervised learning when using high risk-adverse performance metrics such as maximum or $\mathrm{CVaR}_{0.95}$ of unsatisfied demand. This is less clear for less risk-adverse metrics such as expected value or $\mathrm{CVaR}_{0.75}$ of unsatisfied demand, but the clustering approach is still superior in most cases.

We now consider the time required to solve the corresponding robust optimization models. Figure 5.5 shows the Cplex solution time for discrete and for polyhedral uncertainty, which depends on the size of the uncertainty set (note the two different horizontal axes and the logarithmic vertical axis).

It can be seen that even the largest discrete model (that uses all training scenarios directly) is still easier to solve than the smallest polyhedral model (using five hyperplanes in addition to the lower and upper bounds). So this experiment reveals that using discrete uncertainty sets not only results in a better solution quality, they are also easier to solve.

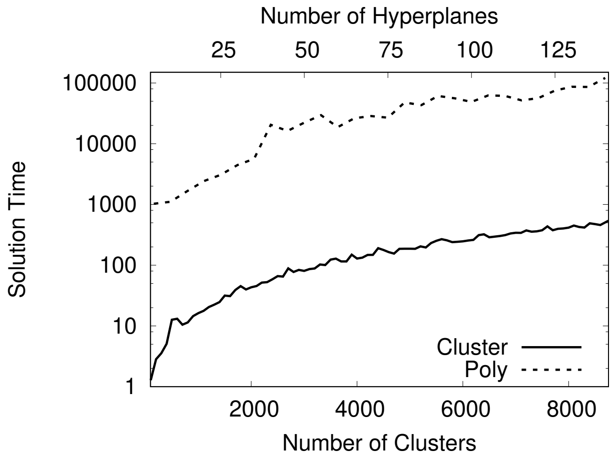


Figure 5.5: Solution times for discrete and polyhedral uncertainty.

5.5 Conclusions

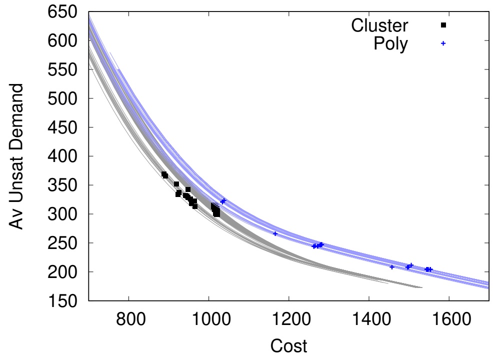
In the robust optimization literature, frequently both discrete and polyhedral uncertainty sets are being used. In this paper we compared the resulting solutions using real-world data for a network expansion problem. We describe how to construct uncertainty sets based on clustering the training data using a well-known data mining technique, and by separating training data from noise using a well-known machine learning method. In our computational study we found that solutions based on discrete uncertainty models outperform solutions based on polyhedral models in most performance metrics, and are also easier to compute. The strong performance of discrete uncertainty sets is in line with evidence from the experiment on shortest path data performed in [41]. This also indicates that the current network design literature, which has a strong focus on polyhedral models, may benefit from considering simple discrete models more.

One reason for this observation may be that the raw data itself does not have a convex shape, and thus an approximation by a convex polyhedron is inadequate. Potentially, a robust optimization approach can use an uncertainty set $\mathcal{U}$ that is the union of multiple polyhedra. While for one-stage min-max problems it holds that optimizing with respect to $\mathcal{U}$ or its convex hull is equivalent, this is not the case for two-stage problems. Two-stage network design with an uncertainty set that is the union of polyhedra may therefore have the potential to reach better solutions than by using a single polyhedron as model for the uncertainty. However, such an approach will come at additional computational cost.

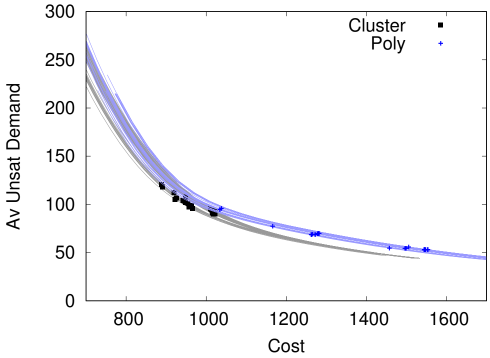
..the knowledge of the Holy One is understanding.
Prov. 9:10

Appendix

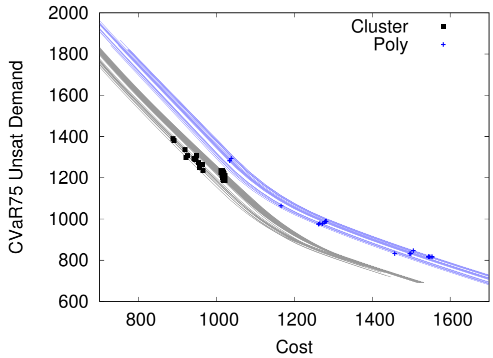
Additional Experimental Results



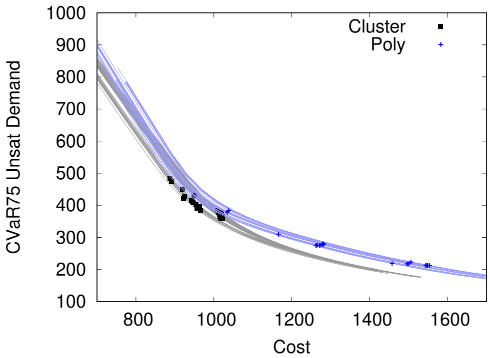
(a) Month 05, average values.



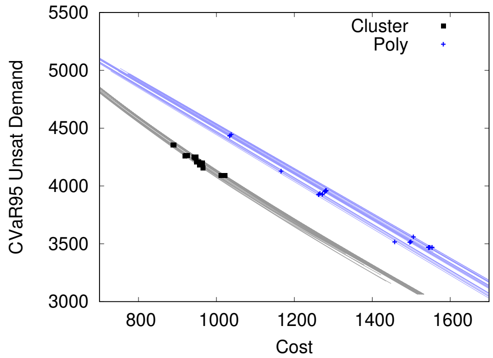
(b) Month 06, average values.



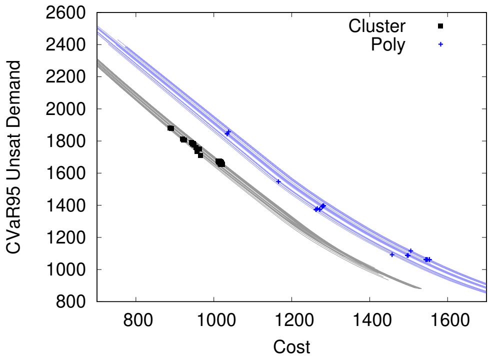
(c) Month 05, $\text{CVaR}_{0.75}$ values.



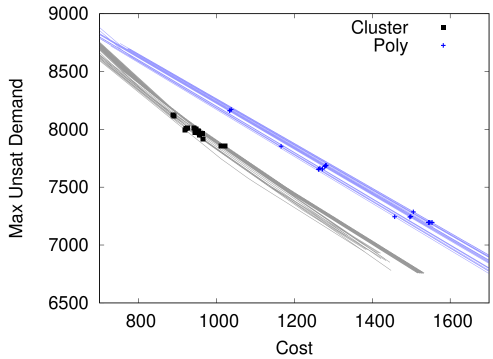
(d) Month 06, $\text{CVaR}_{0.75}$ values.



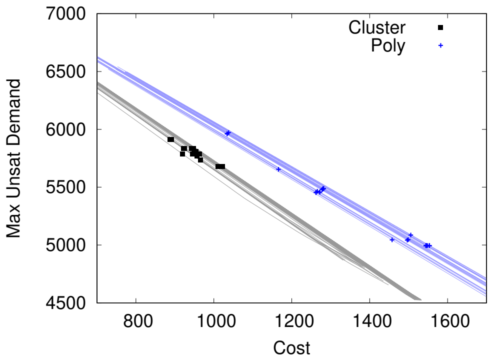
(e) Month 05, $\text{CVaR}_{0.95}$ values.



(f) Month 06, $\text{CVaR}_{0.95}$ values.



(g) Month 05, maximum values.



(h) Month 06, maximum values.

Figure 5.6: Results for months 05 and 06.

The LORD by wisdom, founded the earth...
Prov 3:19

Chapter 6

An Efficient Approach to Distributionally Robust Network Capacity Planning

Abstract. In this paper, we consider a network capacity expansion problem in the context of telecommunication networks, where there is uncertainty associated with the expected traffic demand. We employ a distributionally robust stochastic optimization (DRSO) framework where the ambiguity set of the uncertain demand distribution is constructed using the moments information, the mean and variance. The resulting DRSO problem is formulated as a bilevel optimization problem. We develop an efficient solution algorithm for this problem by characterizing the resulting worst-case two-point distribution, which allows us to reformulate the original problem as a convex optimization problem. In computational experiments the performance of this approach is compared to that of the robust optimization approach with a discrete uncertainty set. The results show that solutions from the DRSO model outperform the robust optimization approach on highly risk-averse performance metrics, whereas the robust solution is better on the less risk-averse metric.

Keywords: network design; robust optimization; optimization in telecommunications; distributionally robust stochastic optimization

6.1 Introduction

Uncertainty has been recognized as a reality of our day-to-day living where choices are often made under partial or unknown information. Hence mitigating against uncertainty in decision making has always been a key business driver. In operations research, frameworks have been developed that help to address decision making under uncertainty in two broad areas, namely the stochastic and the robust optimization approaches.

In the stochastic approach, we assign probabilities to the random variables by assuming that the probability distribution of these variables or uncertain data is known or can be accurately estimated from historic data [45]. A drawback of this approach is that in real life, the probabilities are often not available or correctly estimated. Robust optimization on the other hand addresses the problem of data uncertainty by assuming that the data lie within a closed set [17]. It provides an uncertainty immune solution for the worst case of the uncertain data set. Whereas in the robust optimization approach, we optimize the worst-case objective, in the stochastic optimization approach we optimize relevant statistical measures, e.g. expectation, median, CVaR etc. Ignoring the probability information has been a main criticism of the robust approach which may produce an overly conservative solution [44]. Despite the limitations of these two approaches, network design problems under demand uncertainty using these approaches have been frequently considered. Network design has a strategic role within the planning function of most organizations. The task is to ensure the highest quality of design while efficiently balancing the requirement of just enough capacity with the capacity investment cost. [93, 100] provide a survey of the network design models as well as a unifying framework for many of such models. Network design has found application in many areas, such as transportation, supply chain, communications, and social networks.

[125] showed that stochastic just like robust optimization models are often NP-hard, and even the deterministic network design model itself may be difficult to solve for problems of industrial size. [106] investigated the heuristic methods based on Monte Carlo sampling techniques, stochastic approximation (SA) and sample average approximation (SAA), in their attempt to find a robust stochastic solution. On the other hand, [7] compared the result of the deterministic model to the stochastic model using a standard commercial solver while they proposed the use of heuristics or relaxation methods to solve large-scale problems. [124] determined the quality of the deterministic solutions for a stochastic multi-commodity network design problem and conclude that this solution can be used to find a good heuristic solution to the stochastic multi-commodity network design model. The deterministic solution is hence contained in the stochastic solution and using it as a skeleton improves results with as much as 97% of the initial loss recovered. The framework consists of solving a deterministic network design problem, extracting the discrete variables, fixing them in the stochastic model and then solving a stochastic linear problem.

[117] solved a supply chain network design problem using sample average approximation and combined this with an accelerated Benders decomposition algorithm to solve a problem with a large number of scenarios. [35] also study a supply chain network, which was solved using SAA combined with a Benders decomposition algorithm. [86] formulate a distribution network as a two-stage design problem and solve this using a MIP methodology called the TSMIP.

Literature abounds on the application of robust optimization to network planning and design following the seminal work of [121] and [14, 91, 90, 73, 17, 31, 32] among many others. Of particular importance is the work of [16] which introduced the adjustable robust framework that addresses two-stage decision problem where the network planning and design problem is situated. [4] applied this to network flow design problem under demand uncertainty, a two stage problem which allows the control of the conservatism of the solution via a parameterized budget uncertainty set, [107] studied the network capacity problem under both demand and travel time uncertainty for a multi-commodity flow problem with single source and sink per commodity while [109] introduced affine routing concept in their robust capacity planning model under demand uncertainty using path-based formulation and of recent, [69] addressed non-linearity of cost function in robust network capacity expansion problem. Also, there have been work towards exact solution with [98] being the forerunner while others are [132, 26, 111, 5] using different types of decomposition algorithms.

In this paper, to address the drawback of the two approaches, we leverage on their respective strengths and investigate a distributionally robust stochastic stochastic (DRSO) approach to a multi-commodity network design problem, where the probability distribution itself is affected by ambiguity. This combination of robust and stochastic optimization for a network capacity planning problem is then analyzed with respect to time of execution, resource usage and applicability in a real world setting

This approach has found increasing application in diverse areas/field since its introduction by [118] in his min-max solution of an inventory problem. [113] used this approach in the mean-covariance of the uncertain data distribution to derive a robust solution approach for a max-min and min-max stochastic problem without recourse. [58] in their work on distributionally robust stochastic programming develop a model that combines the distribution and moment of the uncertain data. They show that their model outperforms the naive approximated stochastic model proposed by [113]. However, a polynomial-time algorithm for the larger range of utility functions considered in [113] was beyond the scope of their work. [75] on the other hand develop a tractable approximation to a distributionally robust optimization problem. Unlike most min-max stochastic programs, the expectation of recourse variable was included in their model.

[95] applied the distributionally robust framework to solve an electric vehicle bat-

tery swapping station location problem. They also validate the accuracy of this solution against that of the SAA and were able to show that it provides a good approximation to the SAA. [47] proposed a new reformulation and approximation for solving the distribution robust shortest path problem building on their earlier work [48]. A unifying framework was proposed by [129] for modeling and solving distributionally robust optimization problems based on standardized ambiguity sets which encompasses many sets from the literature. They identify conditions under which this framework is tractable and develop a tractable conservative approximation for problems that violate these conditions. [33] developed a framework for solving adaptive distributionally robust linear optimization problem.

This paper presents the following contributions: We consider an uncertain network capacity expansion problem and develop an efficient algorithm to its distributionally robust counterpart by characterizing the resulting two point distribution using Richter-Rogonski's theorem [115, 116]. We compare the quality of solutions from this algorithm to the solutions obtained from a discrete robust approach. It is observed from our numerical experiments that solutions found by the DRSO algorithm outperform solutions from the robust optimization approach on highly risk-averse performance metrics.

The model from the literature that is the most similar to ours, but with a flow cost, can be found in [105]. While their approach resulted in a computationally challenging solution approach (including the discretization of continuous values), the model presented here is considerably easier to solve.

The rest of this paper is organized as follows. In Section 6.2, we describe the general distributionally robust stochastic optimization concept. Section 6.3 explains the robust network design problem we consider. In Section 6.4, our efficient formulation of the distributionally robust problem is developed. We first focus on the single-commodity case and then extend results to the multi-commodity case. In Section 6.5, the experimental setup and computational result are discussed. Finally, Section 6.6 concludes our work and points out future research directions.

6.2 Distributionally Robust Stochastic Optimization

Distributionally robust stochastic optimization is a data-driven modeling methodology for optimization under uncertainty. It encompasses aspects from both robust and stochastic optimization, frameworks that are complementary to each other though differing in their approaches to addressing the uncertainty [15].

Robust optimization provides a framework to immunize against uncertainty that is believed to lie within a closed and bounded set known as the uncertainty set, while the stochastic optimization framework assumes that the probability distribution of the parameter uncertainty is known. However, in real world applications, "true" distribution knowledge is never completely known but at most can only be estimated from available data [119]. On the other hand, one of the major attractiveness which has resulted in the explosion of research into the robust framework is its tractability to a wide range of challenging problems. Nevertheless, this methodology also faced criticism for its inability to factor in the distribution knowledge of the underlying uncertain data, leading to overly conservative solutions [129, 44].

Uncertainty can be viewed as *risk* when the probability distribution is known or as an *ambiguity* otherwise [33]. Neither of these two approaches is suitable to deal with ambiguity from the perspective of decision theory. However, combining the uncertainty set of the robust approach with the probability distribution from the stochastic approach produces a more potent methodology that is able to handle both risk and ambiguity. Hence, under the distributionally robust stochastic framework, the probability distribution is also subjected to uncertainty. The aim is to find a decision such that for any possible probability distribution from the ambiguity set, the stochastic constraints of the model are satisfied.

In the era of growing data-driven applications, moments that constitute the ambiguity are estimated in the face of limited historical data. [118] was the first to apply this methodology in his min-max proposal to the newsvendor problem where the "true" distribution, though not known completely, is only characterized by its mean and standard deviation and belongs to a class of probability distributions with the same mean and standard deviation. The distributionally robust stochastic approach hence seeks to maximize the expectation by considering the worst case distribution in this probability distributions class also known as the ambiguity set.

6.3 Problem Description

Planning for capacities to be added in networks is a major strategic problem in most telecommunication organizations. Usually, this decision is made under uncertainty of the future traffic demand. As with most strategic roles, this often involves large capital expenditure investment. Hence, in this paper, we consider a multi-commodity network demand flow problem, where additional capacities are added to accommodate uncertain traffic demands while minimizing the total cost involved subject to design constraints.

6.3.1 Basic Network Expansion Problem

The problem can be represented by a directed graph $G = (\mathcal{V}, \mathcal{A})$ which denotes the network of interest. Each of the arcs $a \in \mathcal{A}$ has an original capacity u_a which can be upgraded at a cost c_a per incremental unit of capacity. A set of commodities $\mathcal{K} = \{1, \ldots, K\}$ need to be routed across the network with each commodity $k \in \mathcal{K}$ consisting of a demand $d^k \geq 0$, a source node $s^k \in \mathcal{V}$, and a sink node $t^k \in \mathcal{V}$. Additionally, let ϕ be the cost of not satisfying one unit of demand over the planning horizon (i.e., by outsourcing it). Under complete demand certainty, the nominal network capacity expansion problem can then be formulated as:

$$\min \sum_{a\in\mathcal{A}} c_a x_a + \phi \sum_{k\in\mathcal{K}} \left[d^k - \sum_{a\in\delta^-(t^k)} f_a^k + \sum_{a\in\delta^+(t^k)} f_a^k \right]_+ \tag{6.1}$$

$$\text{s.t.} \sum_{a\in\delta^-(v)} f_a^k - \sum_{a\in\delta^+(v)} f_a^k \geq 0 \qquad \forall k\in\mathcal{K}, d\in\mathcal{U}, v\in\mathcal{V}\setminus\{s^k,t^k\} \tag{6.2}$$

$$\sum_{k\in\mathcal{K}} f_a^k \leq u_a + x_a \qquad \forall a\in\mathcal{A} \tag{6.3}$$

$$f_a^k \geq 0 \qquad \forall k\in\mathcal{K}, d\in\mathcal{U}, a\in\mathcal{A} \tag{6.4}$$

$$x_a \geq 0 \qquad \forall a\in\mathcal{A} \tag{6.5}$$

Here, $[y]_+$ denotes the positive part $\max\{0,y\}$, while $\delta^+(v)$ and $\delta^-(v)$ are the sets of the outgoing and incoming arc at node $v\in\mathcal{V}$, respectively. Variables f_a^k denote the flow of commodity $k\in\mathcal{K}$ along edge $a\in\mathcal{A}$, while x_a models the amount of capacity being added to arc a. The objective function (6.1) is to minimize the sum of capacity expansion cost and outsourcing costs. Constraints (6.2) are a variant of flow constraints, where we allow an arbitrary amount of flow to leave the source node s^k. Through the objective, only the flow arriving in t^k is counted. Finally, Constraints (6.3) which model the capacity on each edge ensure that amount of flow does not exceed the sum of initial and added capacity.

6.3.2 Robust Problem Formulation

Since the actual demand values $\boldsymbol{d} = (d^1,\ldots,d^K) \in \mathbb{R}_+^K$ are uncertain, we assume here that they can take any value in a predetermined uncertainty set $\mathcal{U}$, which can be represented as $\mathcal{U} = \{\boldsymbol{d}^1,\ldots,\boldsymbol{d}^N\}$. The network capacity expansion problem using the robust optimization framework is to minimize the cost of capacity investment and the worst-case costs of outsourced (unsatisfied) demand while satisfying the network constraints. The robust network design for uncertainty set $\mathcal{U}$ is hence:

$$\min \sum_{a\in\mathcal{A}} c_a x_a + \max_{\boldsymbol{d}\in\mathcal{U}} \phi \sum_{k\in\mathcal{K}} \left[d^k - \sum_{a\in\delta^-(t^k)} f_a^k(\boldsymbol{d}) + \sum_{a\in\delta^+(t^k)} f_a^k(\boldsymbol{d}) \right]_+ \tag{6.6}$$

$$\text{s.t.} \sum_{a\in\delta^-(v)} f_a^k(\boldsymbol{d}) - \sum_{a\in\delta^+(v)} f_a^k(\boldsymbol{d}) \geq 0 \qquad \forall k \in \mathcal{K}, \boldsymbol{d} \in \mathcal{U}, v \in \mathcal{V} \setminus \{s^k, t^k\} \tag{6.7}$$

$$\sum_{k\in\mathcal{K}} f_a^k(\boldsymbol{d}) \leq u_a + x_a \qquad \forall \boldsymbol{d} \in \mathcal{U}, a \in \mathcal{A} \tag{6.8}$$

$$f_a^k(\boldsymbol{d}) \geq 0 \qquad \forall k \in \mathcal{K}, \boldsymbol{d} \in \mathcal{U}, a \in \mathcal{A} \tag{6.9}$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{6.10}$$

with a scenario $\boldsymbol{d}$ being the demand vector over all commodities. As before, ϕ is a penalty parameter for uncovered demand (e.g., outsourcing costs). We can send as much flow as we like, but flow cannot appear outside the source, and sending insufficient flow creates a penalty. The constraints (6.1)-(6.5) have been updated to take uncertainty into account, hence the worst case is considered in constraints (6.6). The positive part $[\cdot]_+$ in the objective can be easily linearized using additional variables $\tau^k \geq 0$ and constraints $\tau^k \geq d^k - \sum_{a\in\delta^-(t^k)} f_a^k(\boldsymbol{d}) + \sum_{a\in\delta^+(t^k)} f_a^k(\boldsymbol{d})$ for all $k \in \mathcal{K}$. The inner maximum can be linearized in an analogous way.

6.4 Distributionally Robust Stochastic Problem Formulation

6.4.1 Single-Commodity Case

The single commodity network design problem with outsourcing can be modeled as below. Notice that we drop the subscript k from d^k, s^k and t^k, because we assume a single commodity problem, i.e., $K = 1$;

$$\min \sum_{a \in \mathcal{A}} c_a x_a + \phi \max_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[d - \tilde{d} \right]_+ \tag{6.11}$$

$$\text{s.t.} \sum_{a \in \delta^-(v)} f_a - \sum_{a \in \delta^+(v)} f_a = 0 \qquad \forall v \in \mathcal{V} \backslash \{s, t\} \tag{6.12}$$

$$\sum_{a \in \delta^-(t)} f_a - \sum_{a \in \delta^+(t)} f_a = \tilde{d} \tag{6.13}$$

$$\sum_{a \in \delta^-(s)} f_a - \sum_{a \in \delta^+(s)} f_a = -\tilde{d} \tag{6.14}$$

$$f_a \leq u_a + x_a \qquad \forall a \in \mathcal{A} \tag{6.15}$$

$$f_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{6.16}$$

$$\tilde{d} \geq 0 \tag{6.17}$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{6.18}$$

In this formulation, $\tilde{d}$ is the amount of demand that we intend to satisfy. We consider the problem as a bilevel optimization problem, where first the network owner makes his decision of the amount of demand he wishes to satisfy, and then nature chooses a probability distribution for the demand which maximizes the expected unsatisfied demand. So the second level problem (nature's problem) is

$$\max_{\mathbb{P}} \quad \mathbb{E}_{\mathbb{P}} \left[d - \tilde{d} \right]_+ \tag{6.19}$$

$$\text{s.t.} \quad \mathbb{E}_{\mathbb{P}} \left[d \right] = \mu \tag{6.20}$$

$$\mathbb{E}_{\mathbb{P}}\big[d-\mu\big]^2 = \sigma^2 \tag{6.21}$$

This means we consider all probability distributions $\mathbb{P}$ over d that have the same mean μ and variance σ^2. We denote this set as $\mathcal{P}$. We can thus rewrite the DRSO problem as

$$\min_{(\boldsymbol{x},\boldsymbol{f},\tilde{d})\in X} \sum_{a\in\mathcal{A}} c_a x_a + \phi N(\tilde{d})$$

where $N(\tilde{d})$ denotes the value of the inner nature's problem, and X the set of feasible solutions with respect to $\boldsymbol{x}$, $\boldsymbol{f}$ and $\tilde{d}$.

6.4.2 Model Reformulation

Lemma 6.1. *Let some first-stage solution $(\boldsymbol{x},\boldsymbol{f},\tilde{d})$ be fixed. The optimal objective value of nature's problem can then be written as*

$$N(\tilde{d}) = \begin{cases} 1/2\left(\mu - \tilde{d} + \sqrt{(\tilde{d}-\mu)^2+\sigma^2}\right) & \textit{if } \tilde{d} > \frac{\mu^2+\sigma^2}{2\mu} \\ \mu - \tilde{d}\frac{\mu^2}{\mu^2+\sigma^2} & \textit{if } \tilde{d} \le \frac{\mu^2+\sigma^2}{2\mu} \end{cases} \tag{6.22}$$

Proof. A proof of the result can be found in [92] . Recall that $N(\tilde{d}) = \max_{\mathbb{P}\in\mathcal{P}} \mathbb{E}_{\mathbb{P}}[\max(d-\tilde{d},0)]$. It can be shown that there is a worst-case distribution that is a two-point distribution, which follows from the Richter-Rogonski theorem [115, 116]. For the sake of completeness, we present a proof in Appendix 6.6. $\square$

An example for the shape of function $N(\tilde{d})$ is presented in Figure 6.1.

Lemma 6.2. *The function $N(\tilde{d})$ is convex.*

Proof. The first derivative of N with respect to $\tilde{d}$ is

$$\frac{\partial N}{\partial \tilde{d}} = \begin{cases} \frac{1}{2}\left[\frac{\tilde{d}-\mu}{\sqrt{(\tilde{d}-\mu)^2+\sigma^2}} - 1\right] & \text{if } \tilde{d} > \frac{\mu^2+\sigma^2}{2\mu} \\ -\frac{\mu^2}{\mu^2+\sigma^2} & \text{if } \tilde{d} \le \frac{\mu^2+\sigma^2}{2\mu} \end{cases}$$

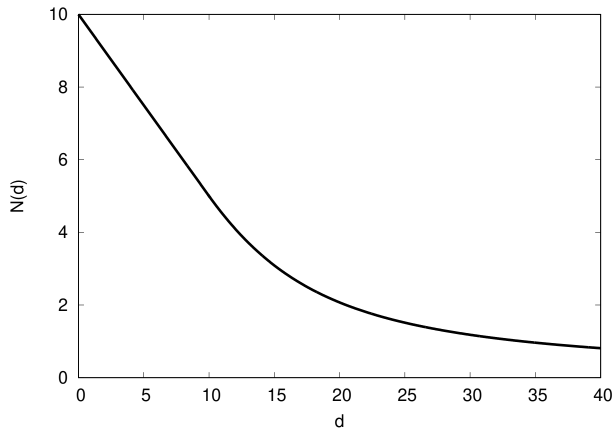


Figure 6.1: Example shape of $N(\tilde{d})$ with $\mu = 10$ and $\sigma^2 = 100$. In this case, $(\mu^2 + \sigma^2)/2\mu = 10$.

and the second derivative is

$$\frac{\partial^2 N}{\partial \tilde{d}^2} = \begin{cases} \frac{\sigma^2}{2((\tilde{d}-\mu)^2+\sigma^2)^{\frac{3}{2}}} & \text{if } \tilde{d} > \frac{\mu^2+\sigma^2}{2\mu} \\ 0 & \text{if } \tilde{d} \leq \frac{\mu^2+\sigma^2}{2\mu} \end{cases}$$

Note that the first derivative is continuous, as

$$\frac{1}{2}\left[\frac{\frac{\mu^2+\sigma^2}{2\mu} - \mu}{\sqrt{(\frac{\mu^2+\sigma^2}{2\mu} - \mu)^2 + \sigma^2}} - 1\right] = \frac{1}{2}\left[\frac{\frac{\sigma^2-\mu^2}{2\mu}}{\sqrt{(\frac{\mu^2+\sigma^2}{2\mu})^2 - (\mu^2+\sigma^2) + \mu^2 + \sigma^2}} - 1\right]$$
$$= \frac{1}{2}\left[\frac{\frac{\sigma^2-\mu^2}{2\mu}}{\sqrt{(\frac{\mu^2+\sigma^2}{2\mu})^2}} - 1\right] = \frac{1}{2}\left[\frac{\frac{\sigma^2-\mu^2}{2\mu}}{\frac{\mu^2+\sigma^2}{2\mu}} - 1\right] = \frac{1}{2}\left[\frac{\sigma^2-\mu^2}{\mu^2+\sigma^2} - 1\right] = \frac{1}{2}\left[-\frac{2\mu^2}{\mu^2+\sigma^2}\right] = -\frac{\mu^2}{\mu^2+\sigma^2},$$

and it is non-decreasing. Hence, the function is convex. □

We now introduce some additional notation. Let $F(\tilde{d})$ be the objective value of the DRSO problem for fixed value of $\tilde{d}$. Then,

$$F(\tilde{d}) = \min_{(\boldsymbol{x},\boldsymbol{f}) \in X(\tilde{d})} \sum_{a \in \mathcal{A}} c_a x_a + \phi N(\tilde{d})$$

where as before, $N(\tilde{d})$ is the objective of the adversary problem, and $X(\tilde{d})$ is the set of vectors $\boldsymbol{x}$ and $\boldsymbol{f}$ that give a flow value $\tilde{d}$. We rewrite this to

$$F(\tilde{d}) = \phi N(\tilde{d}) + G(\tilde{d})$$

where $G(\tilde{d}) = \min_{(\boldsymbol{x},\boldsymbol{f})\in X(\tilde{d})} \sum_{a\in\mathcal{A}} c_a x_a$.

Theorem 6.1. *The function $F(\tilde{d})$ is convex.*

Proof. We show that $G(\tilde{d})$ is a convex function in $\tilde{d}$. This, together with the fact that $N(\tilde{d})$ is convex due to Lemma 6.2, implies that their sum, $F(\tilde{d})$ is also convex.

Consider the function $G'(\tilde{d})$, where

$$G'(\tilde{d}) = \min \sum_{a\in\mathcal{A}} c_a x_a + \psi \cdot \left| \tilde{d} - \sum_{a\in\delta^-(t)} f_a + \sum_{a\in\delta^+(t)} f_a \right| \tag{6.23}$$

$$\text{s.t.} \sum_{a\in\delta^-(v)} f_a - \sum_{a\in\delta^+(v)} f_a = 0 \qquad \forall v \in \mathcal{V}\backslash\{s,t\} \tag{6.24}$$

$$f_a \leq u_a + x_a \qquad \forall a \in \mathcal{A} \tag{6.25}$$

$$f_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{6.26}$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{6.27}$$

for a large value $\psi \geq \sum_{a\in\mathcal{A}} c_a$. Let $(\boldsymbol{x}', \boldsymbol{f}')$ be an optimal solution to $G'(\tilde{d})$ and assume that $\Delta := |\tilde{d} - \sum_{a\in\delta^-(t)} f'_a + \sum_{a\in\delta^+(t)} f'_a| > 0$. Then we can increase each x'_a by Δ to find a new solution where there is sufficient capacity to outsource no demand at all. As increasing the capacity this way increases the costs by $\Delta \sum_{a\in\mathcal{A}} c_a$ and $\psi > \sum_{a\in\mathcal{A}} c_a$, we have constructed a new solution that is feasible and has no higher objective value that $(\boldsymbol{x}', \boldsymbol{f}')$. Hence there is an optimal solution to $G'(\tilde{d})$ that meet exactly a demand of $\tilde{d}$. Therefore, $G'(\tilde{d}) = G(\tilde{d})$.

Recall that if a function $f_1(x, y)$ is convex in (x, y) and C is a convex set, then

$$f_2(x) = \inf_{y\in C} f_1(x, y)$$

is convex as well [37]. Therefore, $G(\tilde{d}) = G'(\tilde{d}) = \min_{(\boldsymbol{x},\boldsymbol{f},\tilde{d})\in X} \sum_{a\in\mathcal{A}} c_a x_a + \psi \cdot |\tilde{d} - \sum_{a\in\delta^-(t)} f_a + \sum_{a\in\delta^+(t)} f_a|$ with X represented by constraints (6.24-6.27) is convex.

$\square$

We can use Theorem 6.1 to solve the single-commodity DRSO problem efficiently. For a fixed value $\tilde{d}$, we evaluate $F(\tilde{d})$ by solving $G(\tilde{d})$ as a linear program and $N(\tilde{d})$ using the formula provided in Equation 6.22. We can now apply standard convex optimization methods (in our experiments we use the Nelder-Mead method) to solve $\min_{\tilde{d}} F(\tilde{d})$ to optimality.

6.4.3 Extension to the Multi-Commodity Case

In this setting, we assume that the demand at each of the origin-destination pairs (s^k, t^k) is affected by a different distribution. In particular, we have mean μ_k and variance σ_k^2 for each $k \in \mathcal{K}$. The nature problem in this case becomes:

$$\max \quad \mathbb{E}_{\mathbb{P}}\left[\sum_{k\in\mathcal{K}} [d^k - \tilde{d}^k]_+\right] \tag{6.28}$$

$$s.t. \quad \mathbb{E}_{\mathbb{P}}\left[d^k\right] = \mu_k \qquad \forall k \in \mathcal{K} \tag{6.29}$$

$$\mathbb{E}_{\mathbb{P}}\left[d^k - \mu_k\right]^2 = \sigma_k^2 \qquad \forall k \in \mathcal{K} \tag{6.30}$$

The DRSO problem can now be written as

$$\min \sum_{a\in\mathcal{A}} c_a x_a + \phi \cdot N(\tilde{\boldsymbol{d}}) \tag{6.31}$$

$$\text{s.t.} \sum_{a\in\delta^-(v)} f_a^k - \sum_{a\in\delta^+(v)} f_a^k = 0 \qquad \forall k \in \mathcal{K}, v \in \mathcal{V}\backslash\{s^k, t^k\} \tag{6.32}$$

$$\sum_{a\in\delta^-(t^k)} f_a^k - \sum_{a\in\delta^+(t^k)} f_a^k = \tilde{d}^k \qquad \forall k \in \mathcal{K} \tag{6.33}$$

$$\sum_{a\in\delta^-(s^k)} f_a^k - \sum_{a\in\delta^+(s^k)} f_a^k = -\tilde{d}^k \qquad \forall k \in \mathcal{K} \tag{6.34}$$

$$\sum_{k \in \mathcal{K}} f_a^k \leq u_a + x_a \qquad \forall k \in \mathcal{K}, a \in \mathcal{A} \tag{6.35}$$

$$f_a^k \geq 0 \qquad \forall k \in \mathcal{K}, a \in \mathcal{A} \tag{6.36}$$

$$x_a \geq 0 \qquad \forall a \in \mathcal{A} \tag{6.37}$$

where $N(\tilde{\boldsymbol{d}})$ denotes the multi-dimensional version of nature's problem as defined by equations (6.28)-(6.30).

Corollary 6.1. *The optimal solution for nature's problem defined by equations* (6.28)-(6.30) *can be expressed as*

$$N(\tilde{\boldsymbol{d}}) = \sum_{k \in \mathcal{K}} N(\tilde{d}^k) \tag{6.38}$$

Proof. This extends lemma 6.1 using the linearity of the expectation on the one hand, and the decomposability of nature's problem on the other hand. We have

$$\begin{aligned}
N(\tilde{\boldsymbol{d}}) &= \max_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[\sum_{k \in \mathcal{K}} [d^k - \tilde{d}^k]_+ \right] \\
&= \max_{\mathbb{P} \in \mathcal{P}} \sum_{k \in \mathcal{K}} \mathbb{E}_{\mathbb{P}} \left[[d^k - \tilde{d}^k]_+ \right] \\
&= \sum_{k \in \mathcal{K}} \max_{\mathbb{P} \in \mathcal{P}} \mathbb{E}_{\mathbb{P}} \left[[d^k - \tilde{d}^k]_+ \right] = \sum_{k \in \mathcal{K}} N(\tilde{d}^k)
\end{aligned}$$

which proves the result. □

Moreover, it can be easily verified that $N(\tilde{\boldsymbol{d}})$ is convex, as it is a sum of convex functions. However, this extension, constraints (6.31)-(6.37), is not quite exact and should be seen as an approximation.

6.5 Experiments

6.5.1 Setup

Our models were implemented using a real-world network from the SNDLib library (see [108]), the Nobel-US network with 14 nodes and 42 arcs. A sample of $10,000$ demand scenarios was generated to form our reference set, sampling i.i.d from a gamma distribution with shape 4 and scale 5 while discarding negative demands and any demand above 50. Three instances of 20 source-sink node pairs were randomly generated for the test network; commodity set A, commodity set B and commodity set C. We then sample training sets for the optimization models. Commodities A and B use the same set of 60 scenarios. For commodity C, we sample a separate set of 60 scenarios. These are used as discrete uncertainty sets for the robust model, and to compute the empirical mean and the variance for the DRSO model.

The solutions found by the two optimization models are then evaluated with demands from the reference set using 5,000 scenarios. The whole experimental setup is repeated 21 times for commodities A and B, and 20 times for commodity C. We provide an overview on the experimental setup in Table 6.1.

The cost of capacity allocation to the arc is randomly generated using a normal distribution with mean 40 and variance 36, while the penalty of unsatisfied or outsourced demand was set to 130 using $10(N-1)$, where N is the number of arcs.

Table 6.1: Experimental setup.

Experiment	Commodity	# Repetitions	# Evaluation per Solution	Total # Evaluation
DRSO Model	A, B	21	5,000	210,000
Robust Model	A, B	21	5,000	210,000
DRSO Model	C	20	5,000	100,000
Robust Model	C	20	5,000	100,000

The results of the two models are recorded as the in-sample results where the first-stage investment cost of the objective value (Cap. Inv) is the cost of deploying capacity and O/S demand is the outsourced (unsatisfied) demand, which when multiplied by the unit penalty cost (ϕ) gives the second-stage outsourcing costs of the objective value. The performance of the evaluation model is reported in terms of mean outsourced demand (E[O/S]), expected maximum outsourced demand(E[max O/S]), average outsourced demand over the worst 5% values (CVaR95), average outsourced demand over the worst 25% values (CVaR75) and the mean satisfied demand E[$\tilde{d}$]. The first of these metrics, the E[O/S], is a low risk measure while the rest three are high risk measures. The two models were implemented using Julia and Gurobi version 7.5 on a Lenovo desktop machine with 8GB RAM and Intel Core i5-65 CPU with 2.50GHz using Windows 10 (64-bit) OS.

The results of the two models, DRSO and Robust, are evaluated in-sample and out-sample using the below model. This seeks to calculate an optimal flow for a fixed scenario $\boldsymbol{d}$ while minimizing the expected outsourced demand τ^k in each commodity $k \in \mathcal{K}$, due to lack of adequate capacity. As the first-stage investment is already fixed, we fix the $\boldsymbol{x}$ solution in this model.

$$
\begin{aligned}
\min \; & \sum_{k \in \mathcal{K}} \tau^k && \\
\text{s.t.} \; & \sum_{a \in \delta^-(v)} f_a^k - \sum_{a \in \delta^+(v)} f_a^k = 0 && \forall k \in \mathcal{K}, v \in \mathcal{V} \backslash \{s^k, t^k\} \\
& \sum_{a \in \delta^-(t^k)} f_a^k - \sum_{a \in \delta^+(t^k)} f_a^k = \tilde{d}^k && \forall k \in \mathcal{K} \\
& \sum_{a \in \delta^-(s^k)} f_a^k - \sum_{a \in \delta^+(s^k)} f_a^k = -\tilde{d}^k && \forall k \in \mathcal{K} \\
& \tau^k \geq d^k - \tilde{d}^k && \forall k \in \mathcal{K} \\
& \sum_{k \in \mathcal{K}} f_a^k \leq u_a + x_a && \forall k \in \mathcal{K}, a \in \mathcal{A} \\
& f_a^k \geq 0 && \forall k \in \mathcal{K}, a \in \mathcal{A} \\
& \tilde{d}^k \geq 0 && \forall k \in \mathcal{K} \\
& \tau^k \geq 0 && \forall k \in \mathcal{K}
\end{aligned}
$$

For each $\boldsymbol{x}$ solution, $5,000$ evaluations are done and the outsourced demand together with satisfied demand are recorded for each evaluation.

6.5.2 Computational Results

Table 6.2 presents a high level summary of the experimental results. Recall that three instances of 20 S-T pairs were used, which are denoted as commodities A, B and C in the table. Each row of results under commodities A and B in the table is the average of 21 instances while the ones under commodity C is the average of 20 instances of different 60 demand samples. The first three columns results are the in-sample optimization result while the next two are the out-of-sample evaluation result. The average (E[O/S]), the maximum (E[Max O/S]), average of the largest 5% (CVaR95) and average of largest 25% (CVaR75) are calculated from the $5,000$ evaluation results for the outsourced demand as the out-of-sample results. While for the satisfied demand, only the average (E[$\tilde{d}$]) is calculated.

Table 6.2: Comparing the two models under two commodities.

Commodity A	Tot. Inv	Cap. Inv	O/S Dem	E[O/S]	E[$\tilde{d}$]	Cap Add	Unit Cost
Robust	44,679.05	32,641.54	92.60	70.91	372.02	842.09	38.76
DRSO	41,830.77	18.671.69	178.15	159.75	267.28	484.99	38.50
Commodity B							
Robust	43,548.63	28,685.06	114.33	86.22	355.32	730.25	39.29
DRSO	41,081.28	19,255.14	167.89	150.49	280.37	489.35	39.35
Commodity C							
Robust	50,878.38	29,226.48	166.55	132.37	281.68	705.39	41.43
DRSO	47,290.23	11,734.21	273.51	261.79	141.53	283.55	41.38

In the following, we focus on the evaluation for commodity type A. Results for commodity types B and C can be found in Appendix B 6.6 and Appendix C 6.6, respectively.

From Table 6.2, it is observed that DRSO solutions build less capacity compared to the robust solutions for all demand instances and hence a lower capital investment, both in terms of total investment and capacity investment. The capacity investment is the cost of adding capacity, the first term of the objective function, while the total investment is the objective value of the optimization problem. The observation seems valid irrespective of the commodity and data set used. For commodity A, the robust solution builds approximately 74% more capacity than the DRSO solution (50% and 150% more in case of commodity B and C, respectively). Though this result is the average over all demand instances, it is also true for each single demand instance, see Figure 6.5c.

Table 6.3: Robust model results for commodity type A.

	In Sample		Out of Sample					
Inst.	Cap. Inv.	O/S Demand	E[O/S]	E[Max O/S]	CVaR95	CVaR75	E[$\tilde{d}$]	CapAdd
1	32,589.64	97.48	72.91	562.76	407.99	231.40	378.76	832.84
2	34,600.41	77.58	59.55	525.62	369.93	199.51	385.59	894.49
3	33,855.56	78.89	65.59	558.41	402.72	219.72	375.00	873.30
4	34,166.13	75.71	60.50	525.91	378.00	204.94	376.94	885.85
5	34,445.25	88.87	64.67	545.86	391.37	215.37	370.99	888.71
6	33,818.93	90.58	65.47	542.46	387.26	215.31	387.39	870.78
7	32,527.15	95.14	75.24	561.43	411.77	239.97	364.20	838.04
8	31,603.34	106.24	74.96	561.52	407.47	232.43	372.81	809.27
9	32,173.50	103.65	77.82	561.57	405.88	231.07	370.11	834.15
10	29,153.00	99.19	91.97	585.70	437.79	264.62	355.20	751.48
11	29,642.85	118.30	82.57	578.90	423.41	250.96	350.37	768.80
12	29,848.79	124.09	86.77	579.68	423.98	245.93	354.20	773.85
13	31,146.65	106.54	75.19	561.82	410.03	239.60	363.83	805.23
14	32,925.94	80.73	67.86	551.43	396.82	222.27	363.21	853.06
15	33,431.34	90.17	67.58	555.29	403.51	228.90	374.53	858.36
16	33,160.00	90.34	68.89	544.42	390.52	216.68	369.76	854.61
17	30,910.42	93.55	75.41	579.02	424.54	246.26	352.29	799.72
18	35,969.65	74.46	52.75	524.08	368.39	190.02	395.79	925.42
19	33,415.01	84.78	67.08	540.52	387.26	212.55	384.07	860.78
20	30,853.02	93.20	78.69	574.63	425.13	248.80	358.67	795.68
21	35,235.78	75.03	57.68	515.23	363.10	200.72	408.72	909.52

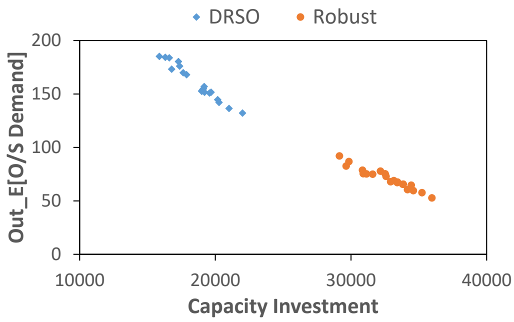
Table 6.3 and Table 6.4 present the results for each demand instance for the robust and DRSO solutions respectively. For the same demand instance, the DRSO solution invests less in capacity, which can be expected since it takes the distribution information of the random variable into consideration unlike the robust model. The DRSO solution is less conservative in this regards, whereas the robust solution plans for the worst observed realization of the random variable. The DRSO solution is therefore cost efficient, building only the needed capacity based on the distribution information

Table 6.4: DRSO model results for commodity type A.

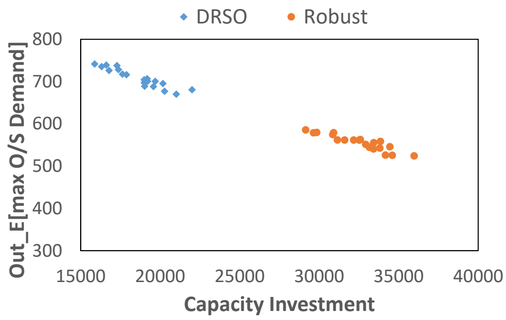
	In Sample			Out of Sample					
Inst.	Cap. Inv.	Nature	$\tilde{d}$	E[O/S]	E[Max O/S]	CVaR95	CVaR75	E[$\tilde{d}$]	CapAdd
1	19,066.17	176.23	300.70	154.02	698.87	543.18	356.25	278.11	495.16
2	19,577.97	171.08	297.69	150.88	688.22	532.53	350.41	279.93	506.00
3	16,785.07	185.68	267.71	173.22	726.25	570.56	383.89	254.52	437.08
4	20,276.51	162.84	307.05	142.09	677.09	521.40	339.98	286.83	529.02
5	17,372.72	192.30	271.26	176.06	728.31	572.62	385.63	246.74	452.77
6	17,882.24	183.66	283.42	168.07	716.15	560.46	373.48	261.94	463.40
7	19,125.03	170.55	292.95	154.98	704.27	548.58	361.60	271.78	495.25
8	17,633.05	197.36	275.90	169.89	717.59	561.90	376.29	258.67	459.89
9	17,282.82	195.63	261.69	180.33	737.88	582.19	395.31	237.93	442.80
10	19,688.38	161.49	292.87	151.62	700.52	544.83	359.57	274.50	507.90
11	19,001.74	172.63	298.09	152.80	697.06	541.37	354.93	274.36	495.37
12	21,012.66	162.36	317.59	136.45	670.18	514.49	332.53	292.79	547.64
13	16,316.88	202.74	264.19	184.32	735.38	579.69	392.82	246.14	424.59
14	18,998.62	172.22	291.80	152.49	704.37	548.68	361.77	268.54	497.03
15	20,174.18	166.03	304.22	144.64	695.35	539.66	352.67	282.83	523.39
16	22,007.06	144.70	318.79	132.10	680.78	525.08	338.10	298.47	569.06
17	19,207.57	172.70	296.34	151.61	700.80	545.11	358.26	269.02	503.15
18	19,184.25	176.21	292.80	156.95	706.77	551.08	364.10	261.23	500.53
19	19,023.42	173.25	296.49	153.29	688.88	533.19	352.03	276.72	493.69
20	15,879.57	198.35	257.87	185.17	741.70	586.01	399.03	243.59	412.61
21	16,609.68	203.07	260.61	183.82	738.96	583.26	396.28	248.14	428.43

which lowers the investment cost, while the robust solution seems to take the more pessimistic route. Comparing each demand instance, the penalty cost of outsourced demand is lower for the robust solution, which is attributed to the fact that it builds for the worst case.

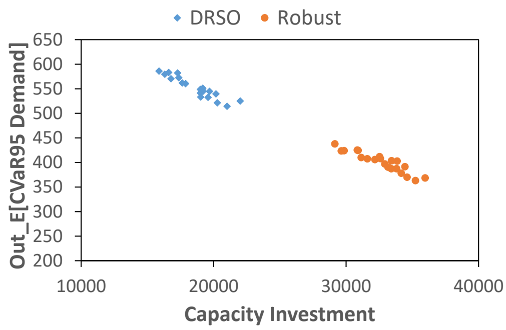
Evaluating the performance of the optimization solution, which is reported by the out-sample performance in Table 6.3 and Table 6.4, the expected unsatisfied demand and expected satisfied demand are lower compared to the DRSO solution. All other high-risk metrics are higher for the DRSO solution. The first four metrics are derived from the expected unsatisfied demand which explains this observation while the satisfied demand on the other hand is a function of the capacity already installed for which the DRSO solution is less conservative. However, this simple comparison may not fully explain the performance of the models, hence we will rely on the charts presented in Figure 6.2 to Figure 6.5. The capacity investment, on the horizontal axis, is compared with the four expected unsatisfied demand metrics in Figure 6.2a to Figure 6.2d.



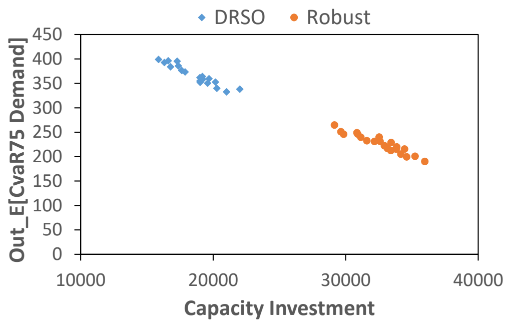
(a) Expected unsatisfied demand.



(b) Expected maximum unsatisfied demand.



(c) CVaR95 unsatisfied demand.

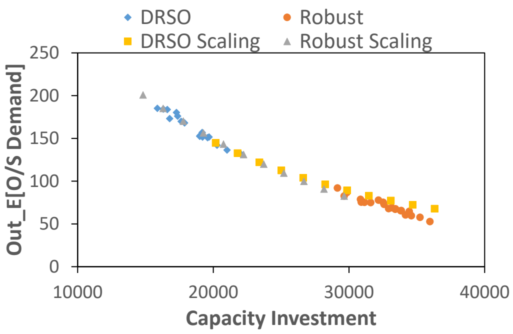


(d) CVaR75 unsatisfied demand.

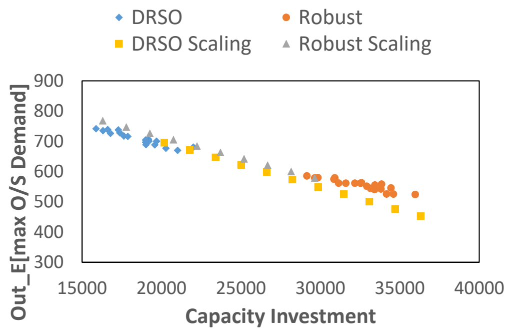
Figure 6.2: Expected unsatisfied demand mean and risk measures (commodity A).

Each of these four charts gives the same observation: Solutions based on the robust model,at higher investment cost, are revealed to be generally more robust than solutions based on DRSO model, which are at lower investment region. For an unexpected surge in traffic, the solution based on robust model will be more able to accommodate this surge compared to the DRSO solution, as its expected outsourced demand is lower. However, in order to allow for a fair comparison of these two models and to be sure this observation is consistent for both models in all the investment regions, we scale a robust solution up towards the DRSO solutions area while scaling down a DRSO solution towards the robust solutions area using expected unsatisfied demand that is presented in Figure 6.2a. This scaling, which was also used in [70], is carried out using a representative data point for each model. We scale the $\boldsymbol{x}$ solution in the direction of interest and re-evaluate with the out-of-sample set. In personal experience, this is also common in the industry, where an optimal solution is scaled up or down during planning iteration, thus allowing for a comprehensive comparison of the two models in all the investment regions. The DRSO $\boldsymbol{x}$ solution is multiplied by a factor of $\lambda = 1.0$ to $\lambda = 1.8$, where λ is the scaling factor, with a scale interval of 0.08, to scale up the solution towards the high investment area while the robust solution is multiplied by a factor of $\lambda = 1.0$ to $\lambda = 0.5$, with a scale interval of 0.05, to scale down the solution towards the low investment area. The result of this scaling is as shown in Figure 6.3 which to the contrary shows that for highly risk-averse metrics (maximum and CVaRs of expected unsatisfied demand), solutions based on the DRSO model are in fact better, having a higher degree of robustness with increasing capacity investment even for high investment. However, for the less risk-averse metric (expected unsatisfied demand), the robust model gives a better solution at higher investment region but with comparable performance for lower investment cost. This observation is consistent for the other two commodities B and C, see Figure 6.7 and Figure 6.10 in the appendix.

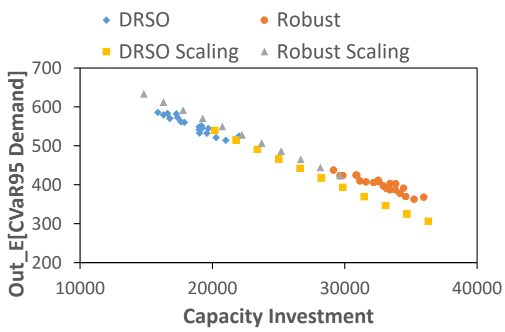
Next we compare the unsatisfied demand (in-sample) to the expected outsourced demand (out-of-sample) for these two models to see which of these gives a better estimate. The charts in Figure 6.4a to Figure 6.4e present these results and they show that the DRSO results produce a far better estimate and hence a better predictor of the input variables under data uncertainty. On the average the input/output ratio of the unsatisfied demand (in-sample) to the expected unsatisfied demand (out-of-sample) is around 10.3% (commodities B and C respectively are 10.3% and 4.29%) for the DRSO model, while for the robust model, this is as high as 23.43% (commodities B and C respectively are 24.59% and 20.52%). Also, regression analysis in Figure 6.4a shows that 93% variation in the estimate is explained by the in-sample result for the DRSO model while for the robust model this is approximately 71%, see Figure 6.4b. The result



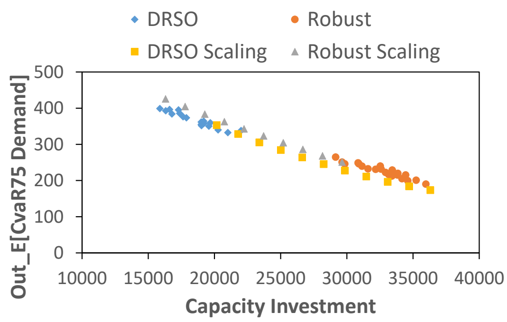
(a) Expected unsatisfied demand.



(b) Expected maximum unsatisfied demand.



(c) CVaR95 unsatisfied demand



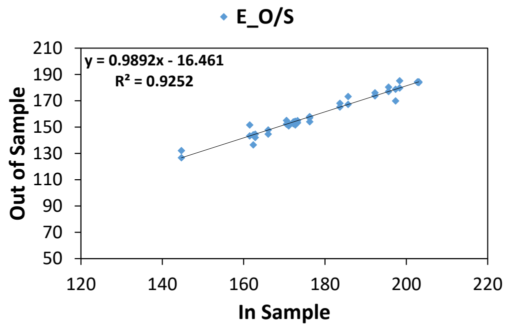
(d) CVaR75 unsatisfied demand.

Figure 6.3: Performance metric scaling (commodity A).

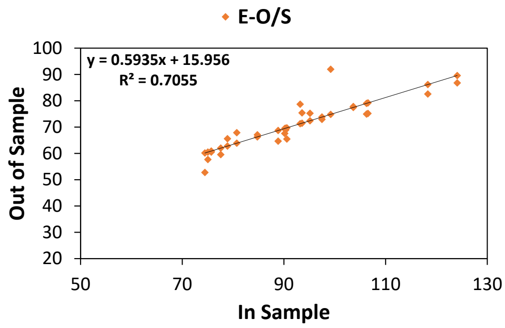
follows the same pattern for the maximum expected unsatisfied demand with respect to the unsatisfied demand in Figure 6.4c for DRSO with $R^2 = 0.8121$ and Figure 6.4d for the robust model with $R^2 = 0.6048$.

A similar trend is also observed for the satisfied demand ($\tilde{d}$) metric with an I/O gap of 7.35% (commodities B and C respectively are 7.23% and 4.20%) while the regression result in Figure 6.4e shows that 94.10% (commodities B and C respectively are 94.78% and 98.83%) variation in the expected satisfied demand (E[$\tilde{d}$]) is explained by the in-sample result, which means that 5.90% variation in the expected satisfied demand is not due to the in-sample satisfied demand.

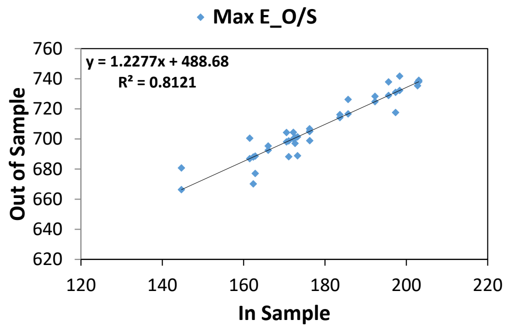
Although the robust solutions follow a pessimistic route and build more capacity



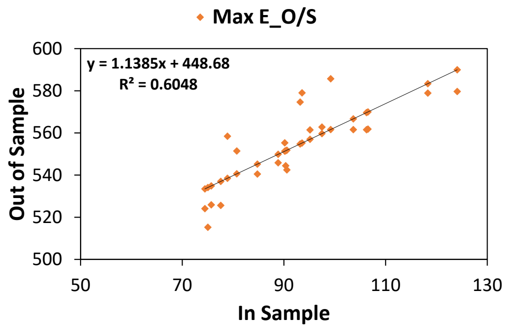
(a) DRSO solutions, expected demand gap.



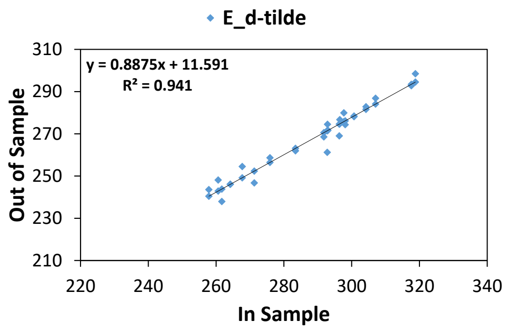
(b) Robust solutions, expected demand gap.



(c) DRSO solutions, expected max demand gap.



(d) Robust solutions, expected max demand gap.

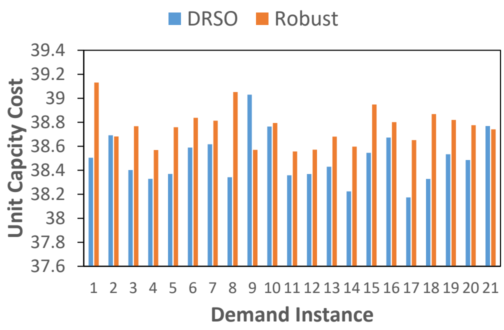


(e) DRSO solutions, expected satisfied demand gap.

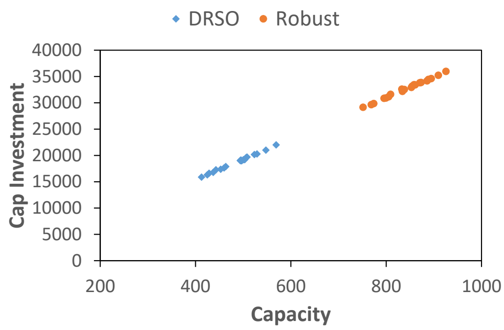
Figure 6.4: Results of out-of-sample prediction (commodity A).

for the same demand instances, there is no observed significant difference in the average unit cost of capacity for these two models irrespective of commodity type and data set, see Table 6.2. For commodity A, for instance, with average unit capacity cost of 38.50 for the DRSO solutions and 38.76 for the robust solutions.

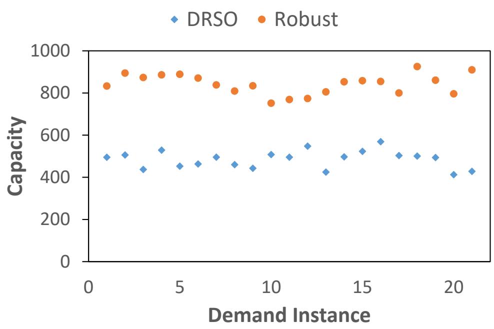
Additional insight on this is provided by Figure 6.5a, which compares the unit cost per instance and by Figure 6.5b, which shows similar linear relationship between capacity and investment for the two models. However, if the capacity installed is compared to the total investment cost, the unit cost of robust solutions becomes cheaper.



(a) Unit cost of capacity.



(b) Capacity investment.



(c) Comparison of capacity deployed.

Figure 6.5: Investment efficiency (commodity A).

6.6 Conclusions

In this paper an efficient approach to distributionally robust network capacity planning under demand ambiguity was proposed. In this approach, we formulate the problem as bilevel optimization, where the worst-case distribution can be characterized by a two-point distribution. This allows us to reformulate the problem as a convex optimization problem, where we need to search over the demand $\tilde{\boldsymbol{d}}$ we intent to satisfy. We then solve this new model using Nelder-Mead algorithm, a convex optimization method.

In order to evaluate the quality of our new approach, the resulting model was compared with the robust approach model on the Nobel-US network taken from the SNDlib[108] database on a number of performance metrics. Our computational result show that solutions from the DRSO model outperform those from the robust model on all high risk-averse performance metrics. Even in the area of solution robustness and quality where the later is generally of a higher robustness, the result scaling shows that solutions from DRSO outperform the robust model in this area on the high risk measures. The robust, however, performs better on the low risk-averse metric.

One interesting result which was also reported earlier by [105] using a different metric is the prediction accuracy of the DRSO model with over 90% expected result variability explained by model result whereas the Robust model cannot be relied upon having a prediction accuracy of approximately 57% and higher. It was also noted that despite the performance difference, the actual unit cost of capacity for this two model is not significantly different.

Moreover, the solutions based on the DRSO were found to be less conservative when compared to Robust model, irrespective of the observed demand instance, data set used and the commodity type, with lower total and capacity investment.

The LORD by wisdom, founded the earth...
Prov 3:19

Appendix

Appendix A: Proof of lemma 6.1

Proof. First suppose that $\tilde{d} \leq \frac{\mu^2+\sigma^2}{2\mu}$.

Here, Nature is characterized by a two point distribution defined by a one-sided Chebyshev inequality below;

$$T = \begin{cases} 0 & \text{w.p. } \frac{\sigma^2}{\sigma^2+\mu^2} \\ \frac{\sigma^2+\mu^2}{\mu} & \text{w.p. } \frac{\mu^2}{\sigma^2+\mu^2} \end{cases} \tag{6.39}$$

with mass $\frac{\sigma^2}{\sigma^2+\mu^2}$ at 0 and vice-versa. Hence, Nature becomes,

$$N(\tilde{d}, \chi_1) = (\chi_1 - \tilde{d}) \left(\frac{\mu^2}{\sigma^2 + \mu^2} \right)$$

where χ_1 is the upper support, hence;

$$N(\tilde{d}) = \left(\frac{\sigma^2 + \mu^2}{\mu} - \tilde{d} \right) \left(\frac{\mu^2}{\sigma^2 + \mu^2} \right)$$
$$N(\tilde{d}) = \mu - \tilde{d} \left(\frac{\mu^2}{\sigma^2 + \mu^2} \right)$$

Suppose now that $\tilde{d} > \frac{\mu^2+\sigma^2}{2\mu}$. Let χ_2 be the upper support point of nature's distribution. Nature objective function can be written as:

$$N(\tilde{d}, \chi_2) = (\chi_2 - \tilde{d})(1 - q), \tag{6.40}$$

where q is probability mass on lower support point. We can express q in terms of χ_2 using the fact that

$$\chi_2 = \mu + \sigma\sqrt{\frac{q}{1-q}},$$

which gives

$$q = \frac{(\chi_2 - \mu)^2}{\sigma^2 + (\chi_2 - \mu)^2}.$$

which can be derived from the following two equations, where α is the lower support point ;

$$p\alpha + (1-p)\chi_2 = \mu$$

$$p\alpha^2 + (1-p)\chi_2^2 = \mu^2 + \sigma^2$$

Then, Equation 6.40 becomes;

$$N(\tilde{d}, \chi_2) = (\chi_2 - \tilde{d})\left(1 - \frac{(\chi_2 - \mu)^2}{(\chi_2 - \mu)^2 + \sigma^2}\right) \tag{6.41}$$

The value of χ at the root maximizes the above function. To this end, the first derivative of N w.r.t χ is

$$\frac{\partial N}{\partial \chi_2} = -\frac{\sigma^2(\chi_2^2 - 2\tilde{d}\chi_2 + 2\mu\tilde{d} - \mu^2 - \sigma^2)}{(\chi_2^2 - 2\mu\chi_2 + \sigma^2 + \mu^2)^2},$$

Setting $\frac{\partial N}{\partial \chi_2} = 0$, produces a root at $\chi_2 = \tilde{d} + \sqrt{(\tilde{d} - \mu)^2 + \sigma^2}$ which when substituted in Equation 6.41 gives

$$N(\tilde{d}) = \left(\tilde{d} + \sqrt{(\tilde{d} - \mu)^2 + \sigma^2} - \tilde{d}\right)\left(1 - \frac{\left(\tilde{d} + \sqrt{(\tilde{d} - \mu)^2 + \sigma^2} - \mu\right)^2}{\left(\tilde{d} + \sqrt{(\tilde{d} - \mu)^2 + \sigma^2} - \mu\right)^2 + \sigma^2}\right)$$

Simplifying the above equation and re-arranging terms result in the below;

$$N(\tilde{d}) = 1/2\left(\mu - \tilde{d} + \sqrt{(\tilde{d}-\mu)^2 + \sigma^2}\right)$$

and this completes the proof of Equation 6.22.

$$N(\tilde{d}) = \begin{cases} 1/2\left(\mu - \tilde{d} + \sqrt{(\tilde{d}-\mu)^2 + \sigma^2}\right) & \text{when } \tilde{d} > \frac{\mu^2+\sigma^2}{2\mu} \\ \mu - \tilde{d}\left(\frac{\mu^2}{\mu^2+\sigma^2}\right) & \text{when } \tilde{d} \leq \frac{\mu^2+\sigma^2}{2\mu} \end{cases}$$

□

Appendix B: Results for commodity type B.

We present additional results for commodity type B in a similar way to the presentation of results for commodity type A in the main text. Table 6.5 and Table 6.6 show key metrics for the 21 repetitions using the robust and the DRSO model, respectively.

Figure 6.6, Figure 6.7 and Figure 6.8 correspond to Figure 6.2, Figure 6.3 and Figure 6.4 using commodity type B instead of A.

Results indicate an overall similarity to the results reported in Section 6.5 on pages 120-128 with the same conclusion as commodity type A.

Appendix C: Results for commodity type C.

We present additional results for commodity type C in a similar way to the presentation of results for commodity type A in the main text. Table 6.7 and Table 6.8 show key metrics for the 20 repetitions using the robust and the DRSO model, respectively. Results of commodity C also aligns with that of commodities A and B.

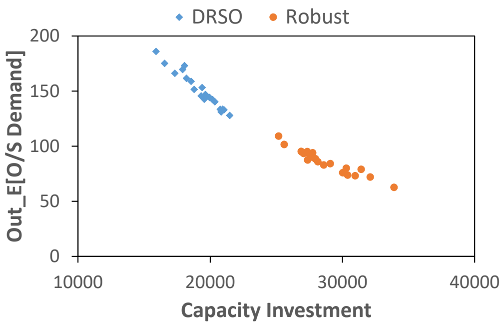
Figure 6.9, Figure 6.10 and Figure 6.11 correspond to Figure 6.2, Figure 6.3 and Figure 6.4 using commodity type B instead of A.

Table 6.5: Robust model results for commodity type B.

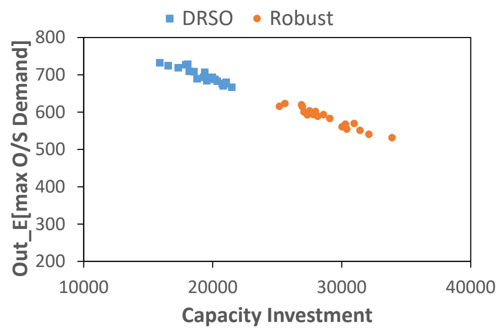
	In Sample		Out of Sample					
Inst.	Cap. Inv.	O/S Demand	E[O/S]	E[Max O/S]	CVaR95	CVaR75	E[$\tilde{d}$]	CapAdd
1	26,955.23	138.21	94.47	614.68	458.99	272.58	342.28	682.99
2	27,340.83	122.58	95.06	592.60	436.90	259.98	358.40	690.01
3	27,966.59	109.42	88.49	601.61	445.92	262.78	349.14	698.86
4	30,966.21	96.20	73.17	569.72	414.03	232.73	356.59	787.31
5	27,081.86	135.60	93.40	600.58	444.89	262.61	342.51	698.33
6	32,108.77	98.27	71.99	540.65	384.96	212.76	384.38	824.14
7	25,174.15	144.43	109.13	615.47	459.78	280.33	340.72	627.89
8	28,144.28	125.35	85.90	588.84	433.15	250.75	363.18	717.50
9	33,905.19	84.59	62.63	531.42	375.74	206.53	394.00	871.49
10	25,598.99	127.12	101.55	622.91	467.21	284.64	329.08	656.61
11	30,395.00	100.40	73.78	554.45	399.09	223.45	379.72	774.90
12	29,084.16	112.90	84.14	583.00	427.31	244.61	358.90	743.51
13	30,291.89	109.15	80.03	567.86	412.16	235.45	352.00	764.24
14	27,374.82	111.42	87.46	593.87	438.18	254.00	356.60	701.42
15	28,590.93	112.90	82.91	593.14	437.45	251.73	351.84	731.89
16	30,027.87	95.84	75.90	560.51	404.82	226.01	367.25	764.01
17	27,506.54	111.69	92.84	603.45	447.76	266.75	337.79	703.11
18	27,813.21	121.74	89.64	593.74	438.04	253.97	354.84	709.47
19	31,422.87	100.67	78.99	550.96	395.27	218.36	357.05	800.68
20	26,886.16	118.71	95.24	619.57	463.88	278.46	341.75	685.73
21	27,750.82	123.86	93.98	594.73	439.04	260.02	343.73	701.20

Table 6.6: DRSO model results for commodity type B.

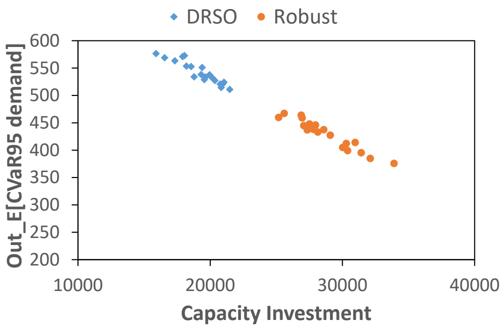
	In Sample			Out of Sample					
Inst.	Cap. Inv.	Nature	$\tilde{d}$	E[O/S]	E[Max O/S]	CVaR95	CVaR75	E[$\tilde{d}$]	CapAdd
1	17,917.30	197.00	272.84	169.65	726.73	571.04	384.06	252.57	450.22
2	20,356.05	156.43	316.57	140.34	683.00	527.31	340.33	293.09	513.17
3	19,387.54	164.77	292.85	153.23	706.72	551.03	364.05	273.34	491.20
4	18,059.27	191.98	271.09	173.10	728.48	572.79	386.11	241.03	460.61
5	19,949.98	166.55	306.07	144.27	693.50	537.81	350.83	279.57	515.17
6	21,042.16	153.28	319.69	132.98	679.88	524.19	337.21	289.33	538.45
7	20,756.93	149.48	323.17	133.46	676.40	520.71	333.73	299.24	527.19
8	19,641.00	171.67	310.38	146.93	689.19	533.50	346.52	292.60	494.15
9	19,625.72	161.37	310.06	143.92	689.51	533.82	346.83	291.45	494.48
10	19,318.96	153.30	305.69	145.59	693.88	538.19	351.21	283.81	493.99
11	18,554.09	179.70	291.15	158.88	708.42	552.73	366.19	263.01	471.08
12	15,903.68	203.29	267.32	186.02	732.25	576.56	389.58	253.73	401.86
13	16,550.21	197.09	275.00	175.14	724.57	568.88	382.16	258.62	420.55
14	17,326.99	183.18	280.58	166.19	718.99	563.29	376.65	254.37	448.43
15	18,206.14	177.24	290.24	161.64	709.33	553.64	366.66	268.19	469.62
16	20,826.06	139.73	329.30	131.07	670.27	514.58	327.60	303.62	529.05
17	20,943.80	152.76	323.21	133.57	676.36	520.66	333.68	301.89	526.64
18	21,479.18	146.50	332.86	127.91	666.71	511.02	324.06	309.97	542.32
19	19,553.38	159.83	314.98	142.59	684.59	528.89	342.11	296.98	498.68
20	20,173.40	156.14	312.15	142.32	687.42	531.73	344.75	287.68	510.07
21	18,786.10	164.48	309.88	151.53	689.69	534.00	347.02	293.73	479.52



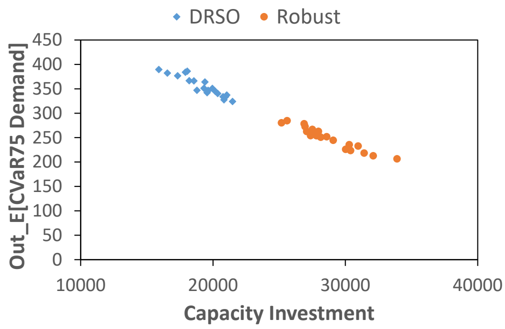
(a) Expected unsatisfied demand.



(b) Expected max unsatisfied demand.

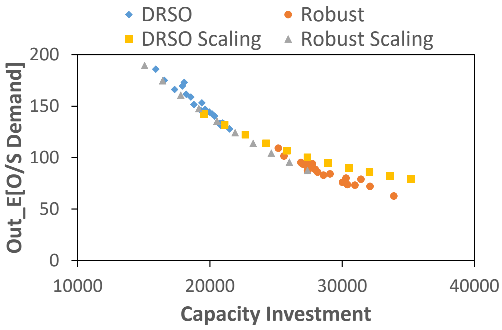


(c) CVaR95 unsatisfied demand.

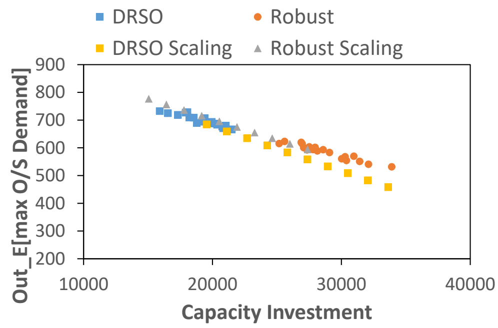


(d) CVaR75 unsatisfied demand.

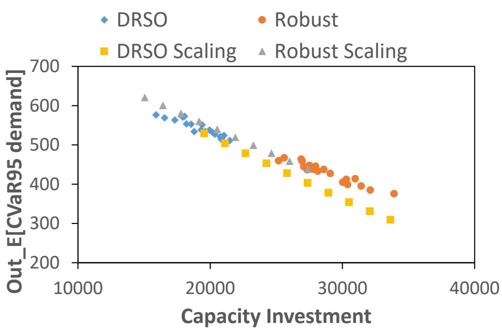
Figure 6.6: Expected unsatisfied demand mean and risk measures (commodity B).



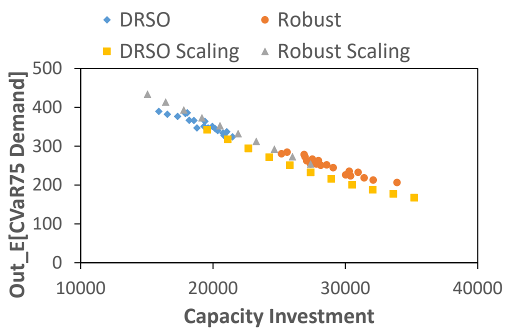
(a) Expected unsatisfied demand.



(b) Expected maximum unsatisfied demand.

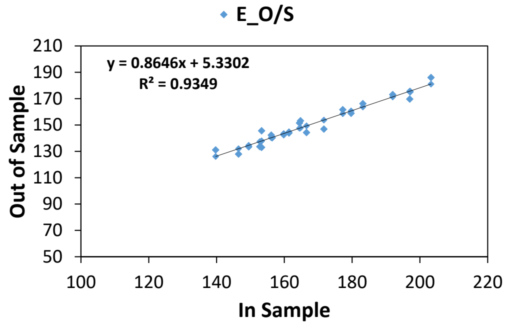


(c) CVaR95 unsatisfied demand

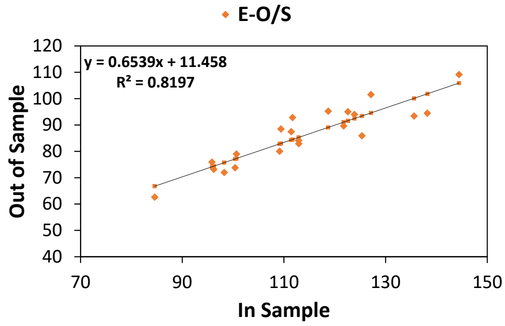


(d) CVaR75 unsatisfied demand.

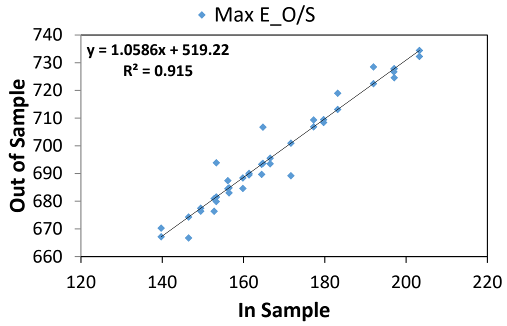
Figure 6.7: Performance metric scaling (commodity B).



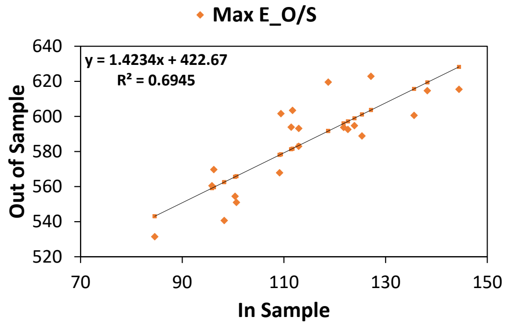
(a) DRSO solutions, expected demand gap.



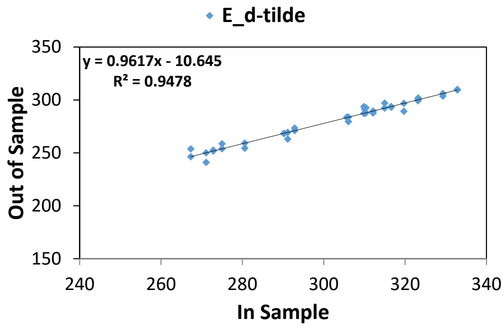
(b) Robust solutions, expected demand gap.



(c) DRSO solutions, expected max demand gap.



(d) Robust solutions, expected max demand gap.



(e) DRSO solutions, expected satisfied demand gap.

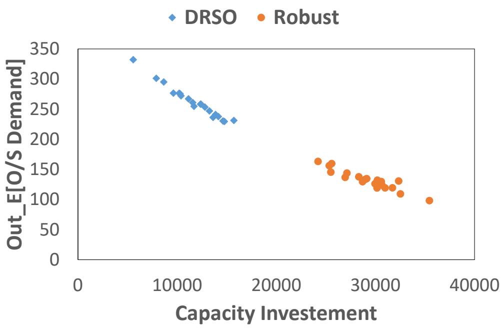
Figure 6.8: Results of out-of-sample prediction (commodity B).

Table 6.7: Robust model results for commodity type C.

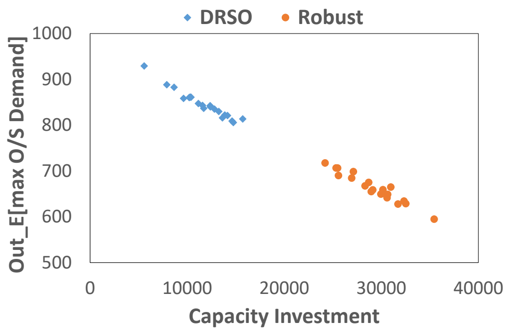
	In Sample		Out of Sample					
Inst.	Cap. Inv.	O/S Demand	E[O/S]	E[Max O/S]	CVaR95	CVaR75	E[$\tilde{d}$]	CapAdd
1	26,956.58	179.10	136.73	684.56	529.60	339.64	276.26	647.25
2	32,347.16	161.78	130.64	634.36	479.54	296.79	279.20	794.22
3	30,172.88	149.01	119.10	659.10	504.14	313.73	282.00	725.67
4	28,334.37	172.10	137.63	667.46	512.50	325.54	273.43	686.66
5	35,453.89	120.88	98.17	594.81	439.86	257.32	320.85	851.52
6	30,982.87	152.36	119.19	664.96	510.01	320.93	293.67	749.94
7	25,591.28	195.04	159.70	690.12	535.33	353.75	264.98	616.11
8	25,347.46	209.66	156.05	706.55	551.59	363.42	246.95	609.26
9	30,205.32	184.27	132.04	651.36	496.41	317.62	275.19	727.79
10	27,130.79	166.38	143.88	698.51	543.55	353.33	266.32	656.16
11	29,966.64	150.23	126.10	649.42	494.47	309.78	285.65	722.97
12	30,664.61	146.39	123.74	648.96	494.00	305.39	291.40	742.28
13	30,595.65	154.07	129.73	641.56	486.60	302.37	296.50	739.28
14	32,529.85	149.01	109.24	628.82	475.08	292.77	310.60	781.49
15	28,710.56	150.83	129.56	674.99	520.03	329.21	279.38	692.88
16	28,970.35	167.93	133.26	655.04	500.08	316.23	308.97	703.85
17	29,124.96	169.75	134.52	658.81	504.96	323.99	283.71	698.58
18	24,216.81	191.10	163.21	717.45	562.50	371.64	249.18	590.91
19	31,721.36	171.99	119.38	627.87	472.94	292.73	281.83	759.90
20	25,506.27	189.19	145.45	706.33	551.37	360.61	267.52	610.99

Table 6.8: DRSO model results for commodity type C.

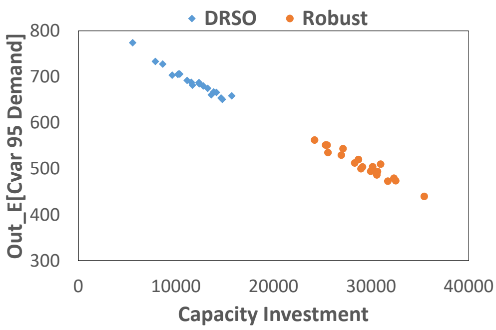
	In Sample			Out of Sample					
Inst.	Cap. Inv.	Nature	$\tilde{d}$	E[O/S]	E[Max O/S]	CVaR95	CVaR75	E[$\tilde{d}$]	CapAdd
1	12,363.99	268.29	152.98	258.35	842.53	687.58	496.72	146.54	294.04
2	5,571.67	345.44	59.95	332.11	929.18	774.23	583.37	65.39	136.01
3	13,259.41	259.92	165.58	247.00	829.93	674.97	484.12	157.41	321.55
4	7,897.34	309.18	105.22	301.23	888.37	733.42	542.56	101.94	186.84
5	12,396.05	285.00	146.38	258.36	839.87	684.91	494.06	142.81	299.13
6	11,576.99	263.34	151.77	261.03	843.09	688.14	497.28	147.41	283.31
7	11,152.00	270.66	148.23	267.12	847.28	692.33	501.47	136.14	266.95
8	14,153.71	249.28	174.53	237.78	820.98	666.03	475.17	162.31	344.49
9	9,627.59	282.76	134.52	276.63	858.54	703.58	512.73	129.57	234.24
10	10,393.15	287.02	126.55	272.23	861.08	706.13	515.27	126.03	249.93
11	10,305.45	284.76	131.37	275.49	861.32	706.36	515.51	130.22	252.57
12	8,650.82	311.90	105.36	295.17	882.67	727.72	536.86	106.36	214.13
13	14,760.43	242.42	189.52	229.55	805.94	650.98	460.13	178.94	355.27
14	10,207.56	295.88	132.09	276.41	860.33	705.37	514.52	125.09	248.92
15	15,725.45	240.47	176.62	231.30	813.68	658.73	467.87	173.52	381.58
16	12,809.80	261.85	160.46	253.39	835.05	680.10	489.24	154.27	312.61
17	14,626.78	243.12	186.55	230.40	808.96	654.00	463.15	171.96	353.18
18	13,876.73	245.18	173.35	241.13	822.12	667.17	476.31	159.20	336.01
19	13,628.69	262.00	175.23	236.30	815.98	661.03	470.17	165.27	327.33
20	11,700.59	261.69	158.25	254.83	836.88	681.93	491.07	150.23	272.91



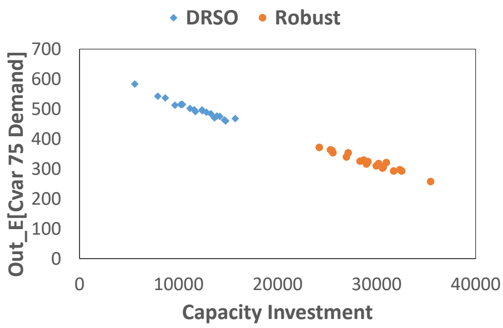
(a) Expected unsatisfied demand.



(b) Expected max unsatisfied demand.

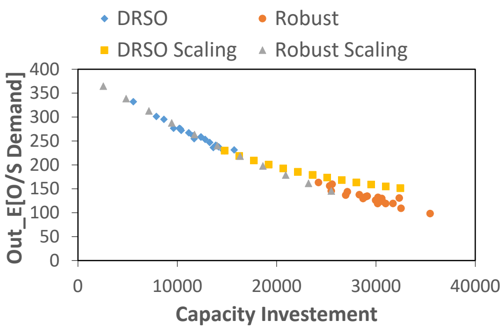


(c) CVaR95 unsatisfied demand.

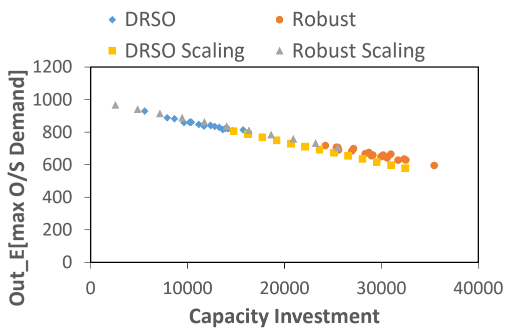


(d) CVaR75 unsatisfied demand.

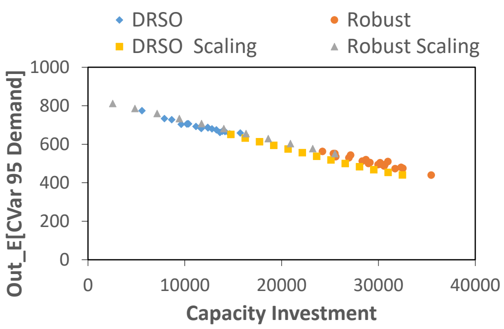
Figure 6.9: Expected unsatisfied demand mean and risk measures (commodity C).



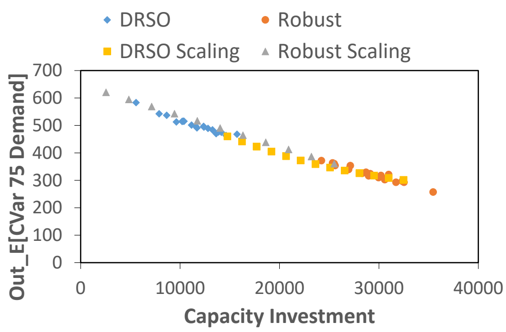
(a) Expected unsatisfied demand.



(b) Expected maximum unsatisfied demand.

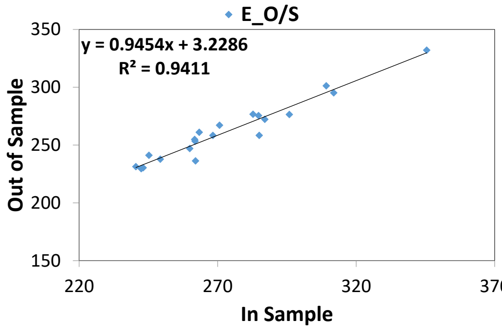


(c) CVaR95 unsatisfied demand

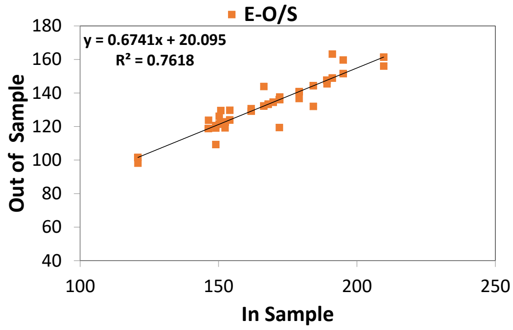


(d) CVaR75 Unsatisfied Demand.

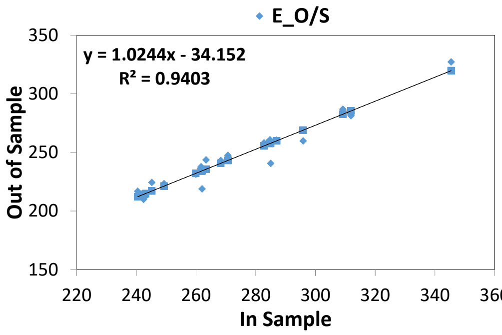
Figure 6.10: Performance metric scaling (commodity C).



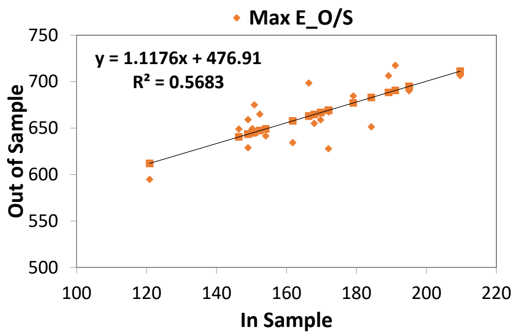
(a) DRSO solutions, expected demand gap.



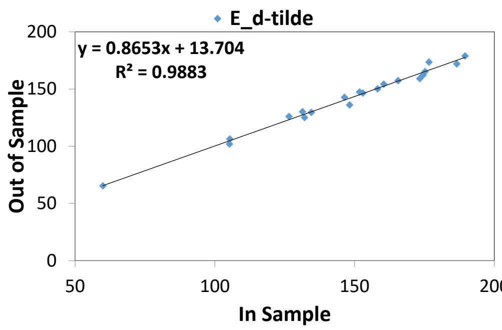
(b) Robust solutions, expected demand gap.



(c) DRSO solutions, expected max demand gap.



(d) Robust solutions, expected max demand gap.



(e) DRSO solutions, expected satisfied demand gap.

Figure 6.11: Results of out-of-sample prediction (commodity C).

Hear the conclusion of the whole matter: Fear God...
Eccl. 12:13

Chapter 7

Conclusion

Network design and capacity planning, strategic business decisions do require innovative tools to be able to deliver on the best business objective. Business decisions and especially that of strategic long term horizon are often made under varying degrees of uncertainty, from partial to complete/full uncertainty. It then becomes imperative that the tools to be used can incorporate uncertainty. Some of the tools that are available in the field of operations research have been investigated in the cause of this thesis and especially as it relates to the practicality of their use in the industry. These tools include stochastic programming, robust optimization and distributionally robust stochastic optimization with a couple of machine learning approaches in the context of a data-driven optimization approach. Two models of uncertainty set are considered along with these tools and these are scenario and polyhedral uncertainty sets.

Different models based on these approaches were developed and compared on several different problem types and objectives with performance comparison made on the result of numerical computation experiments. In Chapter 3, the nonlinear cost model to capacity planning was presented, this being the case in the real world as well in the industry, in contrast to the generally assumed linear cost function in most of the literature. This chapter discusses two types of this, the fixed charge cost model and the piecewise linear cost model. We show that a network of realistic sizes can be designed and solved in a practical amount of time.

In Chapter 4 the choice of uncertainty is addressed in the context of real-world usability from the wide array of uncertainty models proposed in the literature. This becomes necessary as an inappropriate choice does lead to models that may be too conservative or intractable. Robust models for network capacity expansion with outsourced demand are compared for two uncertainty types, discrete and polyhedral, as well as for a deterministic stochastic based model. The performance result shows that solutions based on the discrete uncertainty model performed the best on all metrics while the stochastic optimization model, which is relatively competitive, may fit more complex situations where the robust models are intractable.

Chapter 5 presents a machine learning extension of Chapter 4 where this is used for the generation of the two uncertainty sets under consideration. A discrete uncertainty set is generated using clustering, the K-means, while for a polyhedral uncertainty set, a supervised learning technique is employed. A path-based formulation is implemented for robust optimization, unlike the previous arc-based formulation. Comparing the performance of the resulting solutions from these two models on real-world data, the discrete uncertainty model still outperforms that of the polyhedral robust model in terms of solution quality and computational burden on all measured performance metrics.

Finally, Chapter 6 proposes a unifying approach, the distributionally robust stochastic optimization (DRSO), a data-driven modeling methodology, to the network capacity planning problem under demand ambiguity. The worst-case distribution is characterized by a two-point distribution where the problem is represented as a bi-level optimization problem. The solutions from this model when compared with that from the robust optimization model, show the DRSO to be the better model for the high risk-averse performance metrics. The robust on the other hand performed better on the low-risk averse metrics. Also, the prediction accuracy of the DRSO is found to be much better than the robust model.

In practice, traffic forecasting does precede any network capacity planning and design which entails strong collaboration between the marketing team and the network design/performance team. However, these two functions are treated in silos within the academic research community. It would be beneficial to practitioners, if future research could look at combining these two in a single study, breaking up the current silos, thereby looking at a possible hybrid model. In the era of big data, this becomes even more imperative where machine learning approach could be employed.

One of the findings in this work is the better performance of the discrete uncertainty set compared to the polyhedral set on actual world data. Apparently, literature before now already concluded the polyhedral uncertainty set as the better model. The question that comes to mind is, to what extent this new finding holds true, i.e., what are the possible bounds and limits of this new observation? Hence, it would be worth the while, if further research could be carried out using machine learning approach to see how these compare on a wide array of networks, settings and real world data sets. It would be fine also to know, to what extent is the shape of available data impacts on this performance.

In our robust model to piecewise-linear cost problem, it was observed that the solution times increase considerably with increasing number of scenarios, hence one could look at other algorithms, e.g Benders decomposition and hybrids or any other specialized algorithm, to speed up this computation time. In our piecewise-linear cost model, the Multiple Choice Model(MCM) MIP formulation was used based on the review of existing literatures. This model too could be compared with the two next promising ones, Disaggregated Convex Combination (DCC) and Logarithmic DCC (Dlog), and compare the performance before going the route of speed enhancing algorithms. Alternatively, other uncertainty types, polyhedral in particular, could be used with this non-linear cost model to see how this compares to the scenario uncertainty set.

The result of the DRSO study shows that this model performs better on the high risk metrics but not on the low risk one in comparison to the robust model. Though any decision maker would prefer a model that results in lower capital outlay for the same amount of expected traffic, which would ordinarily make DRSO the preference in practice, however the poor performance on low risk adverse metrics would need to be further investigated and where feasible addressed. What improvement can be made to DRSO model that will ensure it outperforms on all the metrics? Performance of different ambiguity sets could be compared as well as with the robust models also.

Future direction for the research study could also address the area of dynamic network resource allocation optimization. Resource allocation optimization in a mobile network has always generated interest for its network performance assurance role, hence optimization of network performance becomes a continuous process with its strategic inclination. This could still be combined with the Yield Management framework to maximize revenue.

Hear the conclusion of the whole matter: Fear God...
Eccl. 12:13

References

[1] Agarwal, Y. and Aneja, Y. (2016). Fixed charge multicommodity network design using p-partition facets. *European Journal of Operational Research, 258 (2016) 124-135.*

[2] Alekseev, V. B., Kozyrev, V. P., and Sapozhenko, A. A. (2011). Graph theory. *Encyclopedia of Mathematics.* `http://www.encyclopediaofmath.org/index.php?title=Graph_theory&oldid=15471`.

[3] Altın, A., Yaman, H., and Pınar, M. Ç. (2011). The robust network loading problem under hose demand uncertainty: Formulation, polyhedral analysis, and computations. *INFORMS Journal on Computing*, 23(1):75–89.

[4] Atamturk, A. and Zhang, M. (2007). Two-stage robust network flow and design under demand uncertainty. *Operations Research*, 55(4):662–673.

[5] Ayoub, J. and Poss, M. (2016). Decomposition for adjustable robust linear optimization subject to uncertainty polytope. *Computational Management Science*, 13(2):219–239.

[6] Babonneau, F., Vial, J.-P., Klopfenstein, O., and Ouorou, A. (2013). Robust capacity assignment solutions for telecommunications networks with uncertain demands. *Networks*, 62(4):255–272.

[7] Bai, R., Wallace, S. W., Li, J., and Chong, A. Y.-L. (2014). Stochastic service network design with rerouting. *Transportation Research Part B: Methodological*, 60:50–65.

[8] Balakrishnan, A. and Graves, S. C. (1989). A composite algorithm for a concave-cost network flow problem. *Networks*, 19(2):175–202.

[9] Beale, E. and Tomlin, J. A. (1970). Special facilities in a general mathematical programming system for non-convex problems using ordered sets of variables. *J. Lawrence (Ed.), Proc. 5th Int. Conf. Oper. Res.*, pages 447–454.

[10] Beale, E. M. L. (1955). On minimizing a convex function subject to linear inequalities. *Journal of the Royal Statistical Society. Series B*, 17(2):173–184.

[11] Bektas, T., Chouman, M., and Crainic, T. G. (2009). Lagrangean-based decomposition algorithms for multicommodity network design problems with penalized constraints. *Networks*, pages 171–180.

[12] Ben-Ameur, W. and Kerivin, H. (2005). Routing of uncertain traffic demands. *Optimization and Engineering*, 6(3):283–313.

[13] Ben-Ameur, W. and Żotkiewicz, M. (2010). Robust routing and optimal partitioning of a traffic demand polytope. *International Transactions in Operational Research*, 18(3):307–333.

[14] Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (1998). Robust semidefinite programming. *Handbook on Semidefinite Programming*, 27.

[15] Ben-Tal, A., El Ghaoui, L., and Nemirovski, A. (2009). *Robust Optimization*. Princeton University Press.

[16] Ben-Tal, A., Goryashko, A., Guslitzer, E., and Nemirovski, A. (2004). Adjustable robust solutions of uncertain linear programs. *Mathematical Programming*, 99(2):351–376.

[17] Ben-Tal, A. and Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations Research Letters*, 25(1):1–13.

[18] Ben-Tal, A. and Nemirovski, A. (2000). Robust solutions of linear programming problems contaminated with uncertain data. *Mathematical Programming*, 88(3):411–424.

[19] Ben-Tal, A. and Nemirovski, A. (2001). *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. Society for Industrial and Applied Mathematics.

[20] Ben-Tal, A. and Nemirovski, A. (2002). Robust optimization methodology and applications. *Mathematical Programming*, 92(3):453–480.

[21] Bernoulli, D. (1954). Exposition of a new theory on the measurement of risk. *Econometrica*, 22(1):23–36.

[22] Bertsekas, D. P. (1998). *Network Optimization: Continuous and Discrete Models*. Athena Scientific, Belmont, Massachusetts.

[23] Bertsimas, D., Brown, D. B., and Caramanis, C. (2011). Theory and applications of robust optimization. *SIAM Review*, 53(3):464–501.

[24] Bertsimas, D. and Caramanis, C. (2010). Finite adaptability in multistage linear optimization. *IEEE Transactions on Automatic Control*, 55(12):2751–2766.

[25] Bertsimas, D. and Dunning, I. (2016). Multistage robust mixed-integer optimization with adaptive partitions. *Operations Research*, 64(4):980–998.

[26] Bertsimas, D., Dunning, I., and Lubin, M. (2015). Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, 13(2):195–217.

[27] Bertsimas, D. and Georghiou, A. (2015). Design of near optimal decision rules in multistage adaptive mixed-integer optimization. *Operations Research. 2015 Apr 15;63(3):610-27.*

[28] Bertsimas, D. and Goyal, V. (2011). On the power and limitations of affine policies in two-stage adaptive optimization. *Mathematical Programming*, 134(2):491–531.

[29] Bertsimas, D., Gupta, V., and Kallus, N. (2017). Data-driven robust optimization. *Mathematical Programming*, 167(2):235–292.

[30] Bertsimas, D., Pachamanova, D., and Sim, M. (2004). Robust linear optimization under general norms. *Operations Research Letters*, 32(6):510–516.

[31] Bertsimas, D. and Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, 98(1-3):49–71.

[32] Bertsimas, D. and Sim, M. (2004). The price of robustness. *Operations Research*, 52(1):35–53.

[33] Bertsimas, D., Sim, M., and Zhang, M. (2018). Adaptive distributionally robust optimization. *Management Science*, 65(2):604–618.

[34] Bertsimas, D. and Thiele, A. (2006). A robust optimization approach to inventory theory. *Operations Research*, 54(1):150–168.

[35] Bidhandi, H. M. and Yusuff, R. M. (2011). Integrated supply chain planning under uncertainty using an improved stochastic approach. *Applied Mathematical Modelling*, 35(6):2618–2630.

[36] Birge, J. R. and Louveaux, F. (2011). *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer New York.

[37] Boyd, S. and Vandenberghe, L. (2004). *Convex optimization*. Cambridge university press.

[38] Brady, M. E. (2014). Adam smith on uncertainty, degrees of uncertainty and public policy impacts. *SSRN Electronic Journal*.

[39] Charnes, A. and Cooper, W. W. (1959). Chance-constrained programming. *Management Science*, 6(1):73–79.

[40] Charnes, A., Cooper, W. W., and Symonds, G. H. (1958). Cost horizons and certainty equivalents: an approach to stochastic programming of heating oil. *Management science*, 4(3):235–263.

[41] Chassein, A., Dokka, T., and Goerigk, M. (2019). Algorithms and uncertainty sets for data-driven robust shortest path problems. *European Journal of Operational Research*, 274(2):671–686.

[42] Chassein, A., Goerigk, M., Kasperski, A., and Zieliński, P. (2018). Approximating multiobjective combinatorial optimization problems with the OWA criterion. *arXiv preprint arXiv:1804.03594*.

[43] Chekuri, C., Shepherd, F., Oriolo, G., and Scutellá, M. (2007). Hardness of robust network design. *Networks*, 50(1):50–54.

[44] Chen, X., Sim, M., and Sun, P. (2007). A robust optimization perspective of stochastic programming. *Operations Research*, 55(6):1058–1071.

[45] Chen, X., Sim, M., Sun, P., and Zhang, J. (2008). A linear decision-based approximation approach to stochastic programming. *Operations Research*, 56(2):344–357.

[46] Chen, X. and Zhang, Y. (2009). Uncertain linear programs: Extended affinely adjustable robust counterparts. *Operations Research*, 57(6):1469–1482.

[47] Cheng, J., Leung, J., and Lisser, A. (2016). New reformulations of distributionally robust shortest path problem. *Computers & Operations Research*, 74:196–204.

[48] Cheng, J., Lisser, A., and Letournel, M. (2013). Distributionally robust stochastic shortest path problem. *Electronic Notes in Discrete Mathematics*, 41:511–518.

[49] Chouman, M., Crainic, T. G., and Gendron, B. (2009). A cutting-plane algorithm for multicommodity capacitated fixed-charge network design. *CIRRELT*.

[50] Chouman, M., Crainic, T. G., and Gendron, B. (2017). Commodity representations and cut-set-based inequalities for multicommodity capacitated fixed-charge network design. *Transportation Science*, 51(2):650–667.

[51] Chung, B. D., Yao, T., Friesz, T. L., and Liu, H. (2012). Dynamic congestion pricing with demand uncertainty: A robust optimization approach. *Transportation Research Part B: Methodological*, 46(10):1504–1518.

[52] Cisco (2017). Cisco visual networking index: Global mobile data traffic forecast update, 20162021 white paper. Technical Report 1454457600805266, Cisco.

[53] Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network and design problems. *Computers and Operations Research*, 32(6):1429–1450.

[54] Croxton, K. L., Gendron, B., and Magnanti, T. L. (2003). A comparison of mixed-integer programming models for nonconvex piecewise linear cost minimization problems. *Management Science*, 49(9):1268–1273.

[55] Croxton, K. L., Gendron, B., and Magnanti, T. L. (2007). Variable disaggregation in network flow problems with piecewise linear costs. *Operations Research*, 55(1):146–157.

[56] Dantzig, G. B. (1955). Linear programming under uncertainty. *Management Science*, 1(3-4):197–206.

[57] Dantzig, G. B. (1960). On the significance of solving linear programming problems with some integer variables. *Econometrica*, 28(1):30–44.

[58] Delage, E. and Ye, Y. (2010). Distributionally robust optimization under moment uncertainty with application to data-driven problems. *Operations Research*, 58(3):595–612. INFORMS.

[59] Dimitris Bertsimas, Xuan Vinh Doan, K. N. C.-P. T. (2010). Models for minimax stochastic linear optimization problems with risk aversion. *Mathematics of Operations Research*, 35(3):580–602.

[60] Dokka, T., Garuba, F., Goerigk, M., and Jacko, P. (2020). An efficient approach to distributionally robust network capacity planning. *arXiv preprint arXiv:2004.04493v1*.

[61] Duffield, N. G., Goyal, P., Greenberg, A., Mishra, P., Ramakrishnan, K. K., and van der Merive, J. E. (1999). A flexible model for resource management in virtual private networks. *ACM SIGCOMM Computer Communication Review*, 29(4):95–108.

[62] Elias, P., Feinstein, A., and Shannon, C. (1956). A note on the maximum flow through a network. *IEEE Transactions on Information Theory*, 2(4):117–119.

[63] Ericsson (2017). Ericsson mobility report (2017). Technical report, Ericsson Mobility, SE-164 80 Stockholm, Sweden.

[64] Fingerhut, J., Suri, S., and Turner, J. S. (1997). Designing least-cost nonblocking broadband networks. *Journal of Algorithms*, 24(2):287–309.

[65] Ford, L. R. and Fulkerson, D. R. (1956). Maximal flow through a network. *Journal canadien de mathématiques*, 8(0):399–404.

[66] Fulkerson, D. R. and Dantzig, G. B. (1955). Computation of maximal flows in networks. *Naval Research Logistics Quarterly*, 2(4):277–283.

[67] Gabrel, V., Murat, C., and Thiele, A. (2014). Recent advances in robust optimization: An overview. *European Journal of Operational Research*, 235(3):471–483.

[68] Garuba, F., Goerigk, M., and Jacko, P. (2019a). A comparison of models for uncertain network design. *arXiv preprint arXiv:1901.03586v1*.

[69] Garuba, F., Goerigk, M., and Jacko, P. (2019b). Robust network capacity expansion with non-linear costs. In Cacchiani, V. and Marchetti-Spaccamela, A., editors, *19th Symposium on Algorithmic Approaches for Transportation Modelling, Optimization, and Systems, ATMOS 2019, September 12-13, 2019, Munich, Germany*, volume 75 of *OASICS*, pages 5:1–5:13. Schloss Dagstuhl - Leibniz-Zentrum für Informatik.

[70] Garuba, F., Jacko, P., and Goerigk, M. (2019c). A comparison of data-driven uncertainty sets for robust network design. *arXiv preprint arXiv:2003.10507*. In Publication.

[71] Gendron, B. (2002). A note on dual-ascent approach to the fixed-charge capacitated network design problem. *European Journal of Operational Research*, 138:671–675.

[72] Gendron, B., Crainic, T. G., and Frangioni, A. (1999). Multicommodity capacitated network design. In *Telecommunications Network Planning*, pages 1–19. Springer US.

[73] Ghaoui, L. E., Oustry, F., and Lebret, H. (1998). Robust solutions to uncertain semidefinite programs. *SIAM Journal on Optimization*, 9(1):33–52.

[74] Goerigk, M. and Schöbel, A. (2016). Algorithm engineering in robust optimization. In *Algorithm engineering*, pages 245–279. Springer International Publishing.

[75] Goh, J. and Sim, M. (2010). Distributionally robust optimization and its tractable approximations. *Operations Research*, 58(4-part-1):902–917.

[76] González, P. H., Simonetti, L. G., de Jesus Martinhon, C. A., Santos, E., and Michelon, P. Y. P. (2015). A relax and fix approach to solve the fixed charge network design problem with user-optimal flow. In Pinson, E., Valente, F., and Vitoriano, B., editors, *Operations Research and Enterprise Systems: Third International Conference, ICORES 2014, Angers, France, March 6-8, 2014, Revised Selected Papers*, volume 509, pages 173–185. Springer, Cham.

[77] Gorissen, B. L., Yanıkoğlu, İ., and den Hertog, D. (2015). A practical guide to robust optimization. *Omega*, 53:124–137.

[78] Guisewite, G. M. and Pardalos, P. M. (1990). Minimum concave-cost network flow problems: Applications, complexity, and algorithms. *Annals of Operations Research*, 25(1):75–99.

[79] Hannah, L. A. (2015). Stochastic optimization. In *International Encyclopedia of the Social & Behavioral Sciences*, pages 473–481. Elsevier.

[80] Harary, F. (1969). *Graph Theory*. Addison-Wesley Publishing Company, Ontario.

[81] Herrmann, J. W., Ioannou, G., Minis, I., and Proth, J. M. (1996). A dual ascent approach to the fixed-charge capacitated network design problem. *European Journal of Operational Research*, 95(3):476–490.

[82] Higle, J. L. (2005). Stochastic programming: Optimization when uncertainty matters. In *Emerging Theory, Methods, and Applications*, pages 30–53. INFORMS.

[83] Ibaraki, T. (1976). Integer programming formulation of combinatorial optimization problems. *Discrete Mathematics*, 16(1):39–52.

[84] Kall, P. and Mayer, J. (2011). *Stochastic Linear Programming*. Springer US.

[85] Keynes, J. M. (1921). *A Treatise on Probability*. Macmillan and Co, London.

[86] Kılıç, Y. E. and Tuzkaya, U. R. (2014). A two-stage stochastic mixed-integer programming approach to physical distribution network design. *International Journal of Production Research*, 53(4):1291–1306.

[87] Kirchhoff, G. (1847). Ueber die auflsung der gleichungen, auf welche man bei der untersuchung der linearen vertheilung galvanischer strme gefhrt wird. *Annalen der Physik und Chemie*, 148(12):497–508.

[88] Knight, F. H. (1921). *Risk, Uncertainty, and Profit*. Hart, Schaffner & Marx, Boston.

[89] Koster, A. M., Kutschka, M., and Raack, C. (2013). Robust network design: Formulations, valid inequalities, and computations. *Networks*, 61(2):128–149.

[90] Kouvelis, P. and Yu, G. (1997). *Robust Discrete Optimization and Its Applications*. Springer US.

[91] Laguna, M. (1998). Applying robust optimization to capacity expansion of one location in telecommunications with demand uncertainty. *Management Science*, 44(11-part-2):S101–S110.

[92] Lo, A. W. (1987). Semi-parametric upper bounds for option prices and expected payoffs. *Journal of Financial Economics*, 19(2):373–387.

[93] Magnanti, T. L. and Wong, R. T. (1984). Network design and transportation planning: Models and algorithms. *Transportation Science*, 18(1):1–55.

[94] Mahey, P., Benchakroun, A., and Boyer, F. (2001). Capacity and flow assignment of data networks by generalized Benders decomposition. *Journal of Global Optimization*, 20(2):169–189.

[95] Mak, H.-Y., Rong, Y., and Shen, Z.-J. M. (2013). Infrastructure planning for electric vehicles with battery swapping. *Management Science*, 59(7):1557–1575.

[96] Mak, H.-Y., Rong, Y., and Zhang, J. (2015). Appointment scheduling with limited distributional information. *Management Science*, 61(2):316–334.

[97] Markowitz, H. M. and Manne, A. S. (1957). On the solution of discrete programming problems. *Econometrica*, 25(1):84–110.

[98] Mattia, S. (2012). The robust network loading problem with dynamic routing. *Computational Optimization and Applications*, 54(3):619–643.

[99] Meyer, R. (1976). Mixed integer minimization models for piecewise-linear functions of a single variable. *Discrete Mathematics*, 16(2):163–171.

[100] Minoux, M. (1989). Networks synthesis and optimum network design problems: Models, solution methods and applications. *Networks*, 19(3):313–360.

[101] Minoux, M. (2010). Robust network optimization under polyhedral demand uncertainty. *Discrete Applied Mathematics*, 158(5):597–603.

[102] Minoux, M. (2011). On 2-stage robust LP with RHS uncertainty: complexity results and applications. *Journal of Global Optimization*, 49(3):521–537.

[103] Mudchanatongsuk, S., Ordóñez, F., and Liu, J. (2008). Robust solutions for network design under transportation cost and demand uncertainty. *Journal of the Operational Research Society*, 59(5):652–662.

[104] Nahapetyan, A. and Pardalos, P. (2008). Adaptive dynamic cost updating procedure for solving fixed charge network flow problems. *Computational Optimization and Applications*, 39(1):37–50.

[105] Nakao, H., Shen, S., and Chen, Z. (2017). Network design in scarce data environment using moment-based distributionally robust optimization. *Computers & Operations Research*, 88:44–57.

[106] Nemirovski, A., Juditsky, A., Lan, G., and Shapiro, A. (2009). Robust stochastic approximation approach to stochastic programming. *SIAM Journal on Optimization*, 19(4):1574–1609.

[107] Ordóñez, F. and Zhao, J. (2007). Robust capacity expansion of network flows. *Networks*, 50(2):136–145.

[108] Orlowski, S., Wessäly, R., Pióro, M., and Tomaszewski, A. (2010). Sndlib 1.0-Survivable Network Design Library. *Networks.*, 55(3):276–286.

[109] Ouorou, A. and Vial, J.-P. (2007). A model for robust capacity planning for telecommunications networks under demand uncertainty. *IEEE*, page 1.

[110] Pascal, B. (1958). *Pascal's Penses*. Everyman's library. E. P. Dutton & Co., Inc, New York, USA, a dutton paperback reprint. edition.

[111] Pessoa, A. A. and Poss, M. (2015). Robust network design with uncertain outsourcing cost. *INFORMS Journal on Computing*, 27(3):507–524.

[112] Petrou, G., Lemaréchal, C., and Ouorou, A. (2007). An approach to robust network design in telecommunications. *RAIRO - Operations Research*, 41(4):411–426.

[113] Popescu, I. (2007). Robust mean-covariance solutions for stochastic optimization. *Operations Research*, 55(1):98–112.

[114] Poss, M. and Raack, C. (2012). Affine recourse for the robust network design problem: Between static and dynamic routing. *Networks*, 61(2):180–198.

[115] Richter, H. (1957). Parameterfreie Abschätzung und Realisierung von Erwartungswerten. *Bl. Deut. Ges. Vers.-Math*, 3(2):147–162.

[116] Rogosinski, W. W. (1958). Moments of non-negative mass. *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, 245(1240):1–27.

[117] Santoso, T., Ahmed, S., Goetschalckx, M., and Shapiro, A. (2005). A stochastic programming approach for supply chain network design under uncertainty. *European Journal of Operational Research*, 167(1):96–115.

[118] Scarf, H. E. (1957). A min-max solution of an inventory problem. *RAND Corporation*.

[119] Shapiro, A. and Kleywegt, A. (2002). Minimax analysis of stochastic problems. *Optimization Methods and Software*, 17(3):523–542.

[120] Smith, A. (1776). An inquiry into the nature and causes of the wealth of nations.

[121] Soyster, A. L. (1973). Technical note—convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, 21(5):1154–1157.

[122] Sridhar, S., Linderoth, J., and Luedtke, J. (2013). Locally ideal formulations for piecewise linear functions with indicator variables. *Operations Research Letters*, 41(6):627–632.

[123] Sridhar, V. and Park, J. S. (2000). Benders-and-cut algorithm for fixed-charge capacitated network design problem. *European Journal of Operational Research*, 125(3):622–632.

[124] Sun, C., Wallace, S. W., and Luo, L. (2017). Stochastic multi-commodity network design: The quality of deterministic solutions. *Operations Research Letters*, 45(3):266–268.

[125] Thapalia, B. K., Crainic, T. G., Kaut, M., and Wallace, S. W. (2012). Single-commodity network design with random edge capacities. *European Journal of Operational Research*, 220(2):394–403.

[126] Todd, B., Ibigbami, A., and Doucette, J. (2014). Survivable network design and optimization with network families. *Journal of Computer Networks and Communications*, 2014:1–12.

[127] Vielma, J. P., Ahmed, S., and Nemhauser, G. (2010). Mixed-integer models for nonseparable piecewise-linear optimization: Unifying framework and extensions. *Operations Research*, 58(2):303–315.

[128] Vielma, J. P. and Nemhauser, G. L. (2008). Modeling disjunctive constraints with a logarithmic number of binary variables and constraints. In Lodi, A., Panconesi, A., and Rinaldi, G., editors, *Integer Programming and Combinatorial Optimization: 13th International Conference, IPCO 2008 Bertinoro, Italy, May 26-28, 2008 Proceedings*, pages 199–213, Berlin, Heidelberg. Springer Berlin Heidelberg.

[129] Wiesemann, W., Kuhn, D., and Sim, M. (2014). Distributionally robust convex optimization. *Operations Research*, 62(6):1358–1376.

[130] Yue, J., Chen, B., and Wang, M.-C. (2006). Expected value of distribution information for the newsvendor problem. *Operations Research*, 54(6):1128–1136.

[131] Žáčková, J. (1966). On minimax solutions of stochastic linear programming problems. *Časopis pro pěstování matematiky*, 91(4):423–430.

[132] Zeng, B. and Zhao, L. (2013). Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5):457–461.