May 2020

A Mean-Risk Mixed Integer Nonlinear Program for Network Protection

Amy Elizabeth Burton
Clemson University, aburto2@g.clemson.edu

A Mean-risk Mixed Integer Nonlinear Program for Network Protection

---

A Thesis
Presented to
the Graduate School of
Clemson University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science
Mathematical Science

---

by
Amy Elizabeth Burton
May 2020

Accepted by:
Dr. Akshay Gupte, Committee Chair
Dr. Cole Smith
Dr. Matthew Saltzman

Abstract

Many of the infrastructure sectors that are considered to be crucial by the Department of Homeland Security include networked systems (physical and temporal) that function to move some commodity like electricity, people, or even communication from one location of importance to another. The costs associated with these flows make up the price of the network's normal functionality. These networks have limited capacities, which cause the marginal cost of a unit of flow across an edge to increase as congestion builds. In order to limit the expense of a network's normal demand we aim to increase the resilience of the system and specifically the resilience of the arc capacities.

Divisions of critical infrastructure have faced difficulties in recent years as inadequate resources have been available for needed upgrades and repairs. Without being able to determine future factors that cause damage both minor and extreme to the networks, officials must decide how to best allocate the limited funds now so that these essential systems can withstand the heavy weight of society's reliance.

We model these resource allocation decisions using a two-stage stochastic program (SP) for the purpose of network protection. Starting with a general form for a basic two-stage SP, we enforce assumptions that specify characteristics key to this type of decision model. The second stage objective—which represents the price of the network's routine functionality—is nonlinear, as it reflects the increasing marginal cost per unit of additional flow across an arc. After the model has been designed properly to reflect the network protection problem, we are left with a nonconvex, nonlinear, nonseparable risk-neutral program.

This research focuses on key reformulation techniques that transform the problematic model into one that is convex, separable, and much more solvable. Our approach focuses on using perspective functions to convexify the feasibility set of the second stage and second order conic constraints to represent nonlinear constraints in a form that better allows the use of computational solvers. Once these methods have been applied to the risk-neutral model we introduce a risk measure into the first stage that allows us to control the balance between an efficient, solvable model and the need to hedge against extreme events. Using Benders cuts that exploit linear separability, we give a decomposition and solution algorithm for the general network model. The innovations included in this formulation are then implemented on a transportation network with given flow demand.

Acknowledgements

This work would not have been possible without the support and guidance of a number of important individuals. First, I would like to thank my research advisor, Dr. Akshay Gupte. His support of my growth not only as a student but as a professional is beyond measure. He has helped and guided me throughout the entire process of developing this work and enabled me to develop relationships that will stay with me for life. Second, I would like to thank the other two members of my committee, Dr. Cole Smith and Dr. Matthew Saltzman. Both of these individuals have been supportive of my work and helped me through the many challenges I have faced through my course work and research. They have served as role models and I am thankful for the lessons they have taught me.

I would also like to thank the many friends and peers that I have met and worked with while at Clemson University. They have served as a source of support, advice, and friendship and have helped me though the peaks and valleys of graduate school.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Critical infrastructure systems such as energy grids, telecommunication networks, and food distribution systems are essential to the function of all modern societies. Within the 16 infrastructure sectors deemed vital by the Department of Homeland Security, a countless number of systems are in need of refurbishment, repair, or replacement [Directive, 2013]. How to address this problem with the inadequate amount of resources available has become a popular topic of debate and discussion amongst decision makers in our government. Critical infrastructure such as transportation and cyber systems can be modeled as networks, making these difficult decisions network protection problems. There are large amounts of uncertainty and risk in finding the best way to allocate limited resources, especially taking into account possible future events that can affect the network's condition and performance.

Stochastic programming (SP) is a commonly used method for making decisions under uncertainty [Wallace and Ziemba, 2005]. In a general sense, a two-stage stochastic program finds the decision that can be made now that minimizes the expected cost of all future scenarios. Two-stage SPs are often applied to network protection problems [Barbarosoğlu and Arda, 2004, Liu et al., 2009, Lu et al., 2017]. For this type of network protection problem, our first stage deals with how we decide to allocate resources. For each possible scenario, the second stage minimizes costs associated with the network functioning under these conditions.

This research focuses on networks depicted as directed graphs $G(N, A)$ whose functionality involves moving units of flow between locations of significance (nodes). To study the networks properly we must be given necessary information including edge capacities, normal flow demand between pairs of nodes in the graph, and a way to measure the cost of pushing flow across each link. Within these networks we distinguish a subset of edges to be critical. Critical edges are in need of repair or improvement and thus vulnerable to possible capacity loss subject to future conditions. We wish to allocate our limited resources across the critical links in a way that best hedges against future damage and increases the resilience of the system. Similar research has been done on this type of problem, including mean-risk SPs [Liu et al., 2009, Lu et al., 2016, 2017]. This study aims to make improvements on the current methods by establishing a stronger, more tractable formulation second stage problem in network protection models.

The main objective of this research is to build a risk-averse two stage stochastic program that can be used to make the best decision for our network. We start with the basic two stage SP and enforce certain conditions that shape the type of problems we are trying to solve. To avoid any assumptions that might limit the applicability of the model, we use the sum of univariate polynomials as a general second stage function. We then reformulate this program to eliminate issues that arise in efficiently finding the optimal solution. The remainder of this work is organized in the following way: Chapter 2 introduces the general form of a two-stage stochastic program; Chapter 3 demonstrates the convexification and reformulation of the second stage; Chapter 4 makes arguments for a mean-risk model as well as demonstrates how to implement a risk measure into the objective of our model; Chapter 5 outlines the decomposition of the mean-risk model and provides a solution algorithm; Chapter 6 provides an application of the discussed methods for a transportation network.

Chapter 2

Two-stage Stochastic Programs

To build a risk-averse decision model for our network protection problem, we begin with a general two-stage stochastic program. The first stage decision includes making m different decisions. For each decision $i = 1, \ldots, m$ there are n_i possible strategies that can be chosen, each with some cost $c_{i,j}$ for $j = 1, \ldots, n_i$. The first stage decision variable is a binary vector denoted $x = (x^1, \ldots, x^m) \in \{0,1\}^n$ (where $n = \sum_{i=1}^{m} n_i$). For each of decision i, we must make exactly one choice j. For $i = 1, \ldots, m$

$$x^i \in \{0,1\}^{n_i}, \qquad x^i_j = \begin{cases} 1 & \text{If we choose strategy } j \text{ for decision } i \\ 0 & \text{If we do not choose strategy } j \text{ for decision } i \end{cases} \qquad \forall j = 1, \ldots, n_i$$

Since we can only make one choice for each decision i, there can only be one nonzero element in each subvector x^i. This type of vector is referred to as a special ordered set of type 1, denoted $x_i \in$ SOS-1. Since x_i is binary, it is equivalent to say $\sum_{j=1}^{n_i} x^i_j = 1$. After the realization of some uncertainty set $\omega \in \Omega$, (a future unknown event occurs) we hope that our first decision x results in minimal cost $Q(x, \omega)$. The general two-stage SP allows us to solve for the first stage decision x that minimizes the total expected cost incurred on the network.

2.1 Formulation

Our research is focused on stochastic programming problems of the form

$$\min \ \mathbb{E}\left[F(x,\omega)\right] \tag{2.1a}$$

$$\text{s.t. } x = (x^1, x^2, \ldots, x^m) \in X \tag{2.1b}$$

$$x^i \in \{0,1\}^{n_i} \qquad \forall i = 1, \ldots, m \tag{2.1c}$$

$$\sum_{j=1}^{n_i} x^i_j = 1 \qquad \forall i = 1, \ldots, m \tag{2.1d}$$

where $x \in \{0,1\}^n$ is a vector of binary first stage decision variables; $X \subseteq \{0,1\}^n$ is the set of feasible solutions; $(\Omega, \mathcal{F}, P)$ is a probability space with elements ω; and $F : \{0,1\}^n \times \Omega \mapsto \mathbb{R}$ is a cost function with $F(x,\cdot)$ being $\mathcal{F}$-measurable and P-integrable for all $x \in \{0,1\}^n$. The set X contains solutions that satisfy constraints associated with the particular type of problem. The mapping $\mathbb{E} : \mathcal{F} \to \mathbb{R}$ denotes the expected value, where $\mathcal{F}$ is the space of all real random cost variables $F(x,\cdot) : \Omega \mapsto \mathbb{R}$ with finite expectation [Birge and Louveaux, 2011, Kall and Wallace, 1994]. Specifically we study a class of two-stage stochastic programming problems with

$$F(x,\omega) = c^T x + Q(x,\omega),$$

under the following assumptions Shapiro et al. [2009]:

(A1) The set of feasible first stage decision variables is nonempty, that is $X \neq \emptyset$.

(A2) For all $x \in X$ and almost every $\omega \in \Omega$, $F(x, \omega) < \infty$.

(A3) The outcomes of the random variable $\omega \in \Omega$ can be approximated by a finite discrete set of scenarios $s \in S$, each with probability p_s.

Together assumptions (A1) and (A2) guarantee that an optimal solution does exist with assumption (A2) additionally requiring that the recourse function $Q(x, \omega)$ be relatively complete. The recourse function $Q(x, \omega)$ is relatively complete if, for all feasible first stage decisions $x \in X$, the set of second stage feasible solutions is nonempty for almost every $\omega \in \Omega$. Assumption (A3) is required for the model to be tractable and allows us to discretize the expectation of the cost function. Therefore the general two-stage objective (2.1a) can be written as:

$$\begin{aligned}
\min_x \mathbb{E}\left[F(x,\omega)\right] &= \min_x \ \sum_{s\in S} p_s F^s(x) \\
&= \min_x \sum_{s\in S} p_s \left[c^T x + Q^s(x)\right] \\
&= \min_x \ c^T x + \sum_{s\in S} p_s Q^s(x)
\end{aligned}$$

where $F^s(x) = c^T x + Q^s(x)$ is the total cost function for the s-th scenario with $Q^s(x)$ being the optimal value for the second stage cost, given the first stage decision vector x. The recourse function for each scenario s is defined as

$$\begin{aligned}
Q^s(x) \ = \ \min_{y,z} \ & f^s(x, y) \\
\text{s.t. } & B^s y + C^s z \geq d^s \\
& z \geq 0 \\
& y \in \mathbb{R}^m_+.
\end{aligned}$$

where the superscript s on the function $f^s : \{0,1\}^n \times \mathbb{R}^m_+ \mapsto \mathbb{R}$, vector d^s and matrices B^s and C^s represents how the realization s of the probability space $(\Omega, \mathcal{F}, P)$ has an effect on the recourse function. In the study of this general model, we do not have an explicit description for how $Q^s(x)$ changes based on specific $s \in S$. The following assumptions and subsequent formulation should hold for all $s \in S$. For that reason we, when able, denote the recourse function as

$$\begin{aligned} Q(x) \;=\; \min_{y,z} \;& f(x,y) \\ \text{s.t. } & By + Cz \geq d \\ & z \geq 0 \\ & y \in \mathbb{R}^m_+. \end{aligned} \tag{2.2}$$

though we do acknowledge and respect how the discretized outcomes of the random variable $\omega \in \Omega$ can greatly affect the second stage of a stochastic model.

For this research, we consider four additional assumptions about the second stage problem $Q(x)$. These assumptions specify the type of recourse functions we are studying while still allowing the model to be applied to a wide variety of problems

(A4) Problem (2.2) has a finite optimal value for all $x \in X$.

(A5) The objective of $Q(x)$, $f(x,\cdot)$, is a convex function in $\mathbb{R}^m$ for all $x \in \{0,1\}^n$.

(A6) For all $x \in \{0,1\}^n$, $f(x,y) = \sum_{i=1}^{m} f_i(x^i, y_i)$ where $f_i : \{0,1\}^{n_i} \times \mathbb{R}_+ \mapsto \mathbb{R}$ is a univariate function that depends only on $x^i \in \{0,1\}^{n_i} \cap$ SOS-1. Since x^i has n_i possible values, we denote the function $f_i(x^i, y_i) = f_i(e_j, y_i) = g_{i,j}(y_i)$ when x^i is equal to e_j, the j-th standard basis vector in $\mathbb{R}^{n_i}$, meaning the only nonzero element in x^i is a 1 in the j-th position. Thus,

$$f(x, y) = \sum_{i=1}^{m} f_i(x^i, y_i) = \sum_{i=1}^{m}\sum_{j=1}^{n_i} x^i_j g_{i,j}(y_i).$$

(A7) For all $i = 1, \ldots, m$ and $j = 1, \ldots, n_i$ the function $g_{i,j}(y_i)$ is a univaraite polynomial of degree d_i.

$$g_{i,j}(y_i) = a^i_1(y_i) + a^i_2(y_i)^2 + \cdots + a^i_{d_i}(y_i)^{d_i},$$

where $a^{i,j}_0, a^{i,j}_1, a^{i,j}_2, \ldots, a^{i,j}_{d_i} \in \mathbb{R}$ are constant coefficients. The polynomial $g_{i,j}(y_i)$ does not contain any terms of degree zero as those costs would be solely determined by the first stage variable x^i_j and therefore are included as part of the first stage.

By assumption (A4), every first stage solution $x \in X$ has a feasible completion in the second stage [Birge and Louveaux, 2011]. Therefore, the model has complete recourse. The convexity enforced by assumption (A5) implies that the coefficients of $g_{i,j}(y_i)$ must be nonnegative thus, $a^{i,j}_k \geq 0$ for all $i = 1, \ldots, m$ $j = 1, \ldots, n_i$ and $k = 1, \ldots, d_i$. Together assumptions (A6) and (A7) gives that the objective of (2.2) can be written as

$$\begin{aligned} f(x, y) &= \sum_{i=1}^{m}\sum_{j=1}^{n_i} x^i_j g_{i,j}(y_i) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n_i} x^i_j \left(a^{i,j}_1(y_i) + a^{i,j}_2(y_i)^2 + \cdots + a^{i,j}_{d_i}(y_i)^{d_i}\right) \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n_i} x^i_j \left[\sum_{k=1}^{d_i} a^{i,j}_k (y_i)^k\right] \end{aligned}$$

Thus, for a given feasible first stage decision vector $x \in X$ the recourse function $Q(x)$ is:

$$\begin{aligned} Q(x) &= \min_{y,z} \sum_{i=1}^{m} \sum_{j=1}^{n_i} x_j^i \left[\sum_{k=1}^{d_i} a_k^{i,j} (y_i)^k \right] \\ &\quad \text{s.t. } By + Cz \geq d \\ &\quad z \geq 0 \\ &\quad y \in \mathbb{R}_+^m. \end{aligned}$$

Therefore the current formulation of our two-stage stochastic program is:

$$\begin{aligned} &\min_x \; c^T x + \sum_{s \in S} p(s) Q^s(x) && (2.3) \\ &\text{s.t. } x = (x^1, x^2, \ldots, x^m) \in X \\ &\quad x^i \in \{0,1\}^{n_i} && \forall i = 1, \ldots, m \\ &\quad \sum_{j=1}^{n_i} x_j^i = 1 && \forall i = 1, \ldots, m \end{aligned}$$

where for all scenarios $s \in S$

$$\begin{aligned} Q^s(x) &= \min_{y_s, z_s} \sum_{i=1}^{m} \sum_{j=1}^{n_i} x_j^i \left[\sum_{k=1}^{d_i} a_{s,k}^{i,j} (y_{s,i})^k \right] && (2.4) \\ &\quad \text{s.t. } B^s y_s + C^s z_s \geq d^s \\ &\quad z_s \geq 0 \\ &\quad y_s \in \mathbb{R}_+^m. \end{aligned}$$

The optimal solutions y_s^*, z_s^* represent optimal values for y and z given $(a_{s,k}^{i,j}, B^s, C^s, d^s)$. For a different scenario $s' \in S$ any of the given information in the second stage problem may fluctuate. Therefore y_s^*, z_s^* does not necessarily hold for other $s' \in S$. One of the goals in this study is to formulate a model that preforms well even as the size of S increases. Very large sets S model the randomness in the second stage much better than small sets and therefore return optimal solutions better suited for practical application.

Based on the above assumptions the second stage problem $Q^s(x)$ is a nonlinear optimization problem for all $s \in S$. The objective function of (2.4) includes the product of first stage variable x and second stage variable y, thus the first and second stages are nonseparable in this non-convex formulation of the recourse function. Though separability of first and second stage variables is not required to implement Benders Decomposition (BD) Fischetti et al. [2016], Floudas [1995], it along with convexity are properties that can be exploited by the BD algorithm [Benders, 2005, Geoffrion, 1972]. Methods for decomposition and solution of the model will be discussed in Chapter 5. First we must derive a reformulation of $Q^s(x)$ that is a linear function of first stage decision x, convex, and separable in first and second stage variables.

Chapter 3

Convexification and Reformulation of Recourse Function

The specific information constraining each scenario is information given for a particular model. Thus, we focus our reformulation on the general recourse problem

$$
\begin{aligned}
Q(x) \;=\; \min_{y,z} \; & \sum_{i=1}^{m} \sum_{j=1}^{n_i} x_j^i \left[\sum_{k=1}^{d_i} a_k^{i,j} (y_i)^k \right] \\
\text{s.t. } & By + Cz \geq d \\
& z \geq 0 \\
& y \in \mathbb{R}_+^m .
\end{aligned}
$$

as the following reformulation techniques hold for all $s \in S$, regardless of $(a_{s,k}^{i,j}, B^s, C^s, d^s)$ given. For $i = 1, \ldots, m$ and $k = 1, \ldots, d_i$ the objective term

$$
\sum_{j=1}^{n_i} x_j^i \left[\sum_{k=1}^{d_i} a_k^{i,j} (y_i)^k \right]
$$

is non-separable in first (x) and second stage (y) variables and contains nonlinear terms for $k > 1$. Though assumption (A5) requires that the objective of $Q^s(x)$ be convex for all $x \in \{0,1\}^n$ this does not gives us that $Q^s(\cdot)$ is convex in general. To better handle these problems, we move the complicating pieces out of the objective and into the constraints. For every $i = 1, \ldots, m$ we introduce an auxiliary second stage nonnegative continuous variable w_i constrained by

$$w_i \geq \sum_{j=1}^{n_i} x_j^i \left[\sum_{k=1}^{d_i} a_k^{i,j} (y_i)^k \right], \qquad w_i \geq 0 \qquad\qquad \forall i = 1, \ldots, m$$

Then the recourse function can be written with a linear objective as

$$\begin{aligned} Q^s(x) \;=\; & \min_{w,y,z} \sum_{i=1}^{m} w_i && (3.1a) \\ & \text{s.t. } By + Cz \geq d && (3.1b) \\ & z \geq 0 && (3.1c) \\ & y \in \mathbb{R}_+^m && (3.1d) \\ & w_i \geq \sum_{j=1}^{n_i} x_j^i \left[\sum_{k=1}^{d_i} a_k^{i,j} (y_i)^k \right] \qquad \forall i = 1, \ldots, m && (3.1e) \end{aligned}$$

3.1 Convexifying the Union of Disjunctive Sets

For $x \in X$, since $x^i \in \{0,1\}^{n_i} \cap \text{SOS-1}$, constraint (3.1e) causes a disjunction in the set of feasible solutions. For all $i = 1, \ldots, m$ we would like formulate the closed convex hull of the union of these disjunctive convex sets each corresponding to a different possible value of x^i. We now present an efficient approach to dealing with this disjunction.

Remark. For the simplicity of notation, we consider the single union of convex disjunctive sets for some $1 \leq i \leq m$. We drop the indices i and k (other than when i differentiates n from n_i), and consider a single instance of taking the convex hull over the set of disjunctive constraints in the second stage as well as the binary constraints of the first stage decision vector.

For $i = 1, \ldots, m$ we now formulate convex constraints for the following convex set:

$$\Pi_i = \text{conv}\left\{ (x^i, y_i, w_i) \ : \ \begin{array}{cc} \sum_{j=1}^{n_i} x_j^i \left[\sum_{k=1}^{d_i} a_k^{i,j} (y_i)^k \right] \leq w_i; & x^i \in \{0,1\}^{n_i}; \\ y_i \geq 0; & \sum_{j=1}^{n_i} x_j^i = 1 \end{array} \right\}. \tag{3.2}$$

3.1.1 Perspective Reformulation

Ceria and Soares, Ceria and Soares [1999] give a method for approximating the union of convex sets based on a projection of the convex hull into a space of some higher-dimensional set. This projection is due to the perspective mapping of a function. This process has been used successfully to deal with disjunctions caused by indicator variables [Aktürk et al., 2009, Günlük and Linderoth, 2008].

We derive the closure of (3.2) from the union of convex sets corresponding to the indicator variables of this mixed integer program. $\Pi_i = \text{conv}\left(\bigcup_{j=1}^{n_i} P_{i,j} \right)$ where we define

$$P_{i,j} := \left\{ (x^i, y_i, w_i) \in \{0,1\}^{n_i} \times \mathbb{R}_+ \times \mathbb{R}_+ : \ \sum_{k=1}^{d_i} a_k^{i,j} (y_i)^k \leq w_i; \ x^i = e_j \right\}$$

where e_j is the j-th standard basis vector in $\mathbb{R}^{n_i}$ having only one nonzero element, 1 in the j-th row. Given a fixed first stage feasible solution $x^i = e_j$, $P_{i,j}$ is the set of feasible solutions to second stage constraints (3.1d) and (3.1e). It should be noted as the intersection of two hyperplanes and a convex set, $P_{i,j}$ is a convex, bounded set in $\mathbb{R}^{n_i} \times \mathbb{R}_+ \times \mathbb{R}_+$. The first inequality constraint of $P_{i,j}$, $\sum_{k=1}^{d_i} a_k^{i,j}(y_i)^k \leq w_i$, can be rewritten as

$$p_{i,j}(y_i, w_i) \leq 0 \;\text{ where }\; p_{i,j} : \mathbb{R}_+^2 \mapsto \mathbb{R},\; p_{i,j}(y_i, w_i) = \sum_{k=1}^{d_i} a_k^{i,j}(y_i)^k - w_i$$

giving way to an equivalent formulation of $P_{i,j}$:

$$P_{i,j} = \left\{ (x^i, y_i, w_i) \in \{0,1\}^{n_i} \times \mathbb{R}_+ \times \mathbb{R}_+ \;:\; \begin{array}{c} p_{i,j}(y_i, w_i) \leq 0; \\ x^i = e_j \end{array} \right\}.$$

For a general $f : \mathbb{R}^n \to \mathbb{R}$, the perspective function of f is the function $\tilde{f} : \mathbb{R}^{n+1} \to \mathbb{R}$ defined as

$$\tilde{f}(\lambda, x) = \begin{cases} \lambda f(x/\lambda) & \text{if } \lambda > 0 \\ 0 & \text{if } \lambda = 0 \\ \infty & \text{otherwise} \end{cases} . \qquad \text{(Perspective Function)}$$

Proposition 3.1. *If the function $f : \mathbb{R}^n \to \mathbb{R}$ is convex, then the perspective function $\tilde{f} : \mathbb{R}^{n+1} \to \mathbb{R}$ is also convex.*

Readers may visit Günlük and Linderoth [2012] for a proof of Proposition 3.1. The perspective function of $p_{i,j}$ is

$$\tilde{p_{i,j}}(\lambda_j, y_i, w_i) = \begin{cases} \lambda_j p_{i,j}(\frac{y_i}{\lambda_j}, \frac{w_i}{\lambda_j}) = \lambda_j \left[\sum_{k=1}^{d_i} a_k^{i,j} \left(\frac{y_i}{\lambda_j}\right)^k - \frac{w_i}{\lambda_j} \right] = \sum_{k=1}^{d_i} \frac{a_k^{i,j}(y_i)^k}{(\lambda_j)^{k-1}} - w_i & \text{if } \lambda_j > 0 \\ 0 & \text{if } \lambda_j = 0 \\ \infty & \text{otherwise} \end{cases} .$$

Proposition 3.2 (Ceria and Soares [1999]). *For $j = 1, \ldots, n_i$, $P_{i,j}$ is convex thus, $(x^i, y_i, w_i) \in \Pi_i = \text{conv}\left(\bigcup_{j=1}^{n_i} P_{i,j}\right)$ if and only if the following system is feasible :*

$$y_i = \sum_{j=1}^{n_i} v_{i,j};\ w_i = \sum_{j=1}^{n_i} u_{i,j}; \sum_{j=1}^{n_i} \lambda_j = 1;$$

$$\tilde{p_{i,j}}(\lambda_j, v_{i,j}, u_{i,j}) = \sum_{k=1}^{d_i} \frac{a_k^{i,j}(v_{i,j})^k}{(\lambda_j)^{k-1}} - u_{i,j} \leq 0\ \forall j = 1, \ldots, n_i$$

$$v_{i,j}, u_{i,j}, \lambda_j \geq 0\ \forall j = 1, \ldots, n_i$$

Then $(x^i, y_i, w_i) \in \Pi_i$ if and only if there exists $\left(\{v_{i,j}\}_{j=1}^{n_i}, \{u_{i,j}\}_{j=1}^{n_i}\right)$ such that:

$$\Pi_i = \left\{ \left(x^i, y_i, \{v_{i,j}\}_{j=1}^{n_i}, w_i, \{u_{i,j}\}_{j=1}^{n_i},\right) : \begin{array}{r} y_i = \sum_{j=1}^{n_i} v_{i,j},\ w_i = \sum_{j=1}^{n_i} u_{i,j}, \sum_{j=1}^{n_i} x_j^i = 1; \\ \forall j = 1, \ldots, n_i\ \sum_{k=1}^{d_i} \frac{a_k^{i,j}(v_{i,j})^k}{(x_j^i)^{k-1}} \leq u_{i,j}, \\ \forall j = 1, \ldots, n_i\ x_j^i \geq 0,\ v_{i,j} \geq 0,\ u_{i,j} \geq 0 \end{array} \right\}.$$

Therefore the disjunctive constraint (3.1e) can be replaced by the system of inequalities that define Π_i for $i = 1, \ldots, m$. The second stage problem $Q(x)$ takes input of a given first stage decision x. Since $x^i \in \{0,1\}^{n_i} \cap \text{SOS-1}$ for $i = 1, \ldots, m$ in any feasible first stage decision, the constraint $\sum_{j=1}^{n_i} x_j^i = 1$ of Π_i should not be included in $Q(x)$. Using the rest of the constraints from Π_i, we can form a continuous relaxation of $Q(x)$ with the addition of $2n_i$ auxiliary second stage nonnegative variables for each

$i = 1, \ldots, m$.

$$Q(x) = \min_{u,v,w,y,z} \sum_{i=1}^{m} w_i \tag{3.3a}$$

$$\text{s.t. } By + Cz \geq d \tag{3.3b}$$

$$z \geq 0 \tag{3.3c}$$

$$y \in \mathbb{R}_+^m \tag{3.3d}$$

$$y_i = \sum_{j=1}^{n_i} v_{i,j}, \; w_i = \sum_{j=1}^{n_i} u_{i,j} \qquad \forall i = 1, \ldots, m \tag{3.3e}$$

$$\sum_{k=1}^{d_i} \frac{a_k^{i,j}(v_{i,j})^k}{(x_j^i)^{k-1}} \leq u_{i,j} \qquad \forall i = 1, \ldots, m \; j = 1, \ldots, n_i \tag{3.3f}$$

$$v_{i,j}, u_{i,j} \geq 0 \qquad \forall i = 1, \ldots, m \; j = 1, \ldots, n_i \tag{3.3g}$$

3.2 Second Order Cone Programming

The current formulation of the recourse function $Q(x)$ is a convex minimization problem with linear objective. All of the constraints on the second stage are also linear except for constraints (3.3f). Nonlinear constraints pose a computational problem as nonlinear optimization solvers are not as efficient as linear solvers. Many techniques used to solve mixed integer linear programs such as cutting planes cannot be applied directly to mixed integer problems with nonlinear constraints.

Second order cone programs (SOCP) are convex nonlinear optimization problems that minimize a linear objective over the intersection of an affine linear space and the Cartesian product of second order (Lorentz) cones. Figure 3.1 illustrates the 3-dimensional Lorentz cone. The n-dimensional Lorentz (second order) cone is defined by the set:

$$\mathbf{L}^n := \left\{ x = (x_1, x_2, \ldots, x_n) \in \mathbb{R}^n \ : \ x_n \geq \sqrt{x_1^2 + x_2^2 + \ldots x_{n-1}^2} \right\}, \quad n \geq 2.$$

An SOCP is a conic problem of the form

$$\min_x \left\{ c^T x \mid Ax - b \geq_{\mathbf{K}} 0 \right\}$$

where the cone $\mathbf{K} = \mathbf{L}^{n_1} \times \mathbf{L}^{n_2} \times \cdots \times \mathbf{L}^{n_k}$ is a direct product of several Lorentz cones. Second order cone programming falls in between linear programming and semidefinite programming and can be solved using interior point methods in polynomial time, a far improvement to generic mixed integer nonlinear problems. SOCPs have been formulated for use in many fields including structural optimization, Tchebychev approximation, antenna array design, and portfolio optimization [Aktürk et al., 2009, Alizedeh and Goldfarb, 2003, Lobo et al., 1998]. Second order cone programming opens opportunity to solve previously difficult problems in a computationally

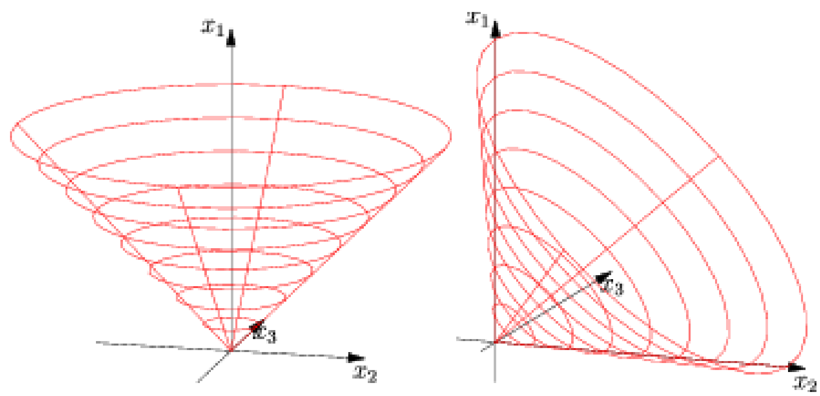

Figure 3.1: Boundary of the quadratic (second order, Lorentz) cone $x_1 \geq \sqrt{x_2^2 + x_3^2}$ (left) and rotated quadratic cone $2x_1x_2 \geq x_3^2$,$x_1, x_2 \geq 0$ (right) [MOSEK, 2018].

tractable way. For these reasons we choose to address the nonlinear constraints in the second stage by formulating them as SOCPs.

SOCPs, also known as conic quadratic problems, can be written as

$$\begin{aligned} \min_x \ & c^T x \\ \text{s.t. } & A_i x - b_i \geq_{\mathbf{L}^{n_i}} 0, \qquad i = 1, \ldots, m \end{aligned}$$

where the inequality constraints means that the last entry in the vector formed by $A_i x - b_i$ is greater than or equal to the Euclidean norm of vector made up of the first $n_i - 1$ entries in $A_i x - b_i$. Constraints such as this can also be written as:

$$\|A_i' x - b_i'\|_2 \leq p_i^T x - q_i$$

where A_i' and b_i' are submatrices of A_i and b_i respectively, consisting on the first $n_i - 1$ rows, while p_i^T and q_i represents the last rows of A_i and b_i respectively. To transform a general nonlinear problem into a conic quadratic problem we must explicitly represent the set of feasible solutions using a finite number of conic quadratic inequalities (CQIs).

Let $X \subset \mathbb{R}^n$ be the set of feasible solutions for some general minimization problem. Then X is categorized as *conic quadratic representable* (CQr), meaning we can represent it using a finite number of CQIs, if there exists a finite system of inequalities, denoted Λ, of the form

$$A_i \binom{x}{\tau} - b_i \geq_{\mathbf{L}^{n_i}} 0 \tag{3.4}$$

where $x \in \mathbb{R}^n$ are the original variables and additional design variables τ are added in such a way that X is the projection of the solution set of Λ onto the x-space. The system Λ is referred to as a *conic quadratic representation* (CQR) of the set X. A function is said to be CQr if its epigraph is a CQr set. Simple examples of CQr functions include constant functions, affine maps, and naturally the Euclidean norm. Operations that preserve the conic quadratic representability of sets include the intersection of CQr sets, projection of a set to an affine image, and affine parameterization.

Lemma 3.3 (Ben-Tal and Nemirovski [2001]). *Any set $X \subset \mathbb{R}^n$ that is CQr is also convex. Since the epigraph of a CQr function is a CQr set and thus convex, it is a necessary condition that a CQr function be convex.*

Remark. Once again we simplify notation in an effort to clarify the reformulation process. Constraint (3.3f) needs reformulation for $i = 1, \ldots, m$, $j = 1, \ldots, n_i$. We focus on one instance with $1 \leq i \leq m$ and $1 \leq j \leq n_i$ and remove the indices i and j from the representation. The simplified form of the constraints (3.3f) and (3.3g) is

$$\sum_{k=1}^{d} \frac{a_k (v)^k}{(x)^{k-1}} \leq u, \qquad x, v, u \geq 0. \tag{3.5}$$

We derive a conic quadratic representation (CQR) for the set defined by (3.5).

An instance of (x, v, u) satisfies the inequality of (3.5) if and only if it belongs to epigraph of the function $f_+ : \mathbb{R}^2_+ \mapsto \mathbb{R}$ defined by

$$f_+(x, v) = \sum_{k=1}^{d} \frac{a_k(v)^k}{(x)^{k-1}}.$$

The epigraph of f is defined as

$$\text{Epi}\{f_+\} = \left\{(x, v, u) \in \mathbb{R}^2_+ \times \mathbb{R}_+ \mid (x, v) \in \text{dom}(f_+),\ f_+(x, v) \leq u\right\}.$$

The domain of f_+ is defined in the nonnegativity constraints of (3.5), and therefore can be written as such. Since the set defined by (3.5) is contained in the first quadrant of $\mathbb{R}^3$, we relax the domain of f_+ to the entirety of $\mathbb{R}^2$ and denote the relaxed function $f : \mathbb{R}^2 \mapsto \mathbb{R}$. A point (x, v, u) satisifies (3.5) if and only if it belongs to the intersection of $Epi\{f\}$ and $\mathbb{R}^3_+$, which we denote $Epi_+\{f\}$.

$$\text{Epi}_+\{f\} = \left\{(x, v, u) \in \mathbb{R}^2 \times \mathbb{R} \mid x, v, u \geq 0,\ f(x, v) \leq u\right\}. \tag{3.6}$$

Ben-Tal and Nemirovski present a wide range of functions whose epigraphs are CQr, as well as operations that preserve the conic quadratic representability of a set [Ben-Tal and Nemirovski, 2001]. We now prove the set (3.6) can be formulated as a system of conic quadratic inequalities using methods asserted by Ben-Tal and Nemirovski [Ben-Tal and Nemirovski, 2001]. Once we have proven that a CQr exists for (3.6), we use the binary tree method described by Alizedeh and Goldfarb [2003] to derive the CQR. Ben-Tal and Nemirovski give a different approach by which to derive the CQR of sets such as (3.6); however, the method requires the addition of far more design variables and inequalities and is therefore not favorable for studies such as this.

3.2.1 Proof of Conic Quadratic-Representability

Proposition 3.4. *The set* (3.6) *is conic quadratic representable, i.e., there exists a finite system* Λ *of vector inequalities of the form* (3.4) *such that* $(x, v, u) \in Epi_+\{f\}$ *if and only if* (x, v, u) *can be extended to a solution* (x, v, u, ξ) *of* Λ.

Proof. To begin, Ben-Tal and Nemirovski have proven that the intersection of CQr sets is CQr as well as the half-spaces defined by $x \geq 0$, $v \geq 0$ and $u \geq 0$ Ben-Tal and Nemirovski [2001]. Therefore if the function $f(x, v)$ is CQr, the set $Epi_+\{f\}$ is the intersection of CQr sets and thus CQr. It should be noted that due to the convexity assumption (A5) in Section 2.1, $a_k \geq 0$ for all $k = 1, \ldots, d$. Hence the function $f(x, v)$ can be written as the summation of d functions $f_1, \ldots, f_d$ with nonnegative weights $a_1, \ldots, a_k$.

$$
\begin{aligned}
f(x, v) &= \sum_{k=1}^{d} \frac{a_k (v)^k}{(x)^{k-1}} \\
&= a_1 v + \frac{a_2 (v)^2}{x} + \cdots + \frac{a_{d-1}(v)^{d-1}}{(x)^{d-2}} + \frac{a_d (v)^d}{(x)^{d-1}} \\
&= a_1 f_1(x, v) + a_2 f_2(x, v) + \cdots + a_{d-1} f_{d-1}(x, v) + a_d f_d(x, v)
\end{aligned}
$$

where the functions $f_k(x, v) : \mathbb{R}^2_+ \mapsto \mathbb{R}$ are defined as

$$
f_k(x, v) = \frac{(v)^k}{(x)^{k-1}} \qquad \text{for } k = 1, \ldots, d.
$$

If the functions $f_k(x, v)$ are CQr for all $k = 1, \ldots, d$, then the summation with nonnegative weights, $f(x, v)$ is also CQr. (Proven in [Ben-Tal and Nemirovski, 2001].) Therefore it suffices for us to prove that the function $f_k(x, v) = \frac{(v)^k}{(x)^{k-1}}$ is CQr (or equivalently that the epigraph of f_k is a conic quadratic representable set) for all $k = 1, \ldots, d$. For the case $k = 1$, $f_1(x, v) = v$ which is a linear function and therefore

CQr. No further reformulation is necessary to derive a CQI for $k = 1$ since linear inequalities are also conic quadratic inequalities. For $k > 1$, we can rewrite the epigraph of f_k as:

$$\begin{aligned}
\text{Epi}\{f_k\} &= \left\{(x, v, u) \in \mathbb{R}^2_+ \times \mathbb{R}_+ \mid \frac{(v)^k}{(x)^{k-1}} \le u\right\} \\
&= \left\{(x, v, u) \in \mathbb{R}^2_+ \times \mathbb{R}_+ \mid (v)^k \le u(x)^{k-1}\right\} \\
&= \left\{(x, v, u) \in \mathbb{R}^2_+ \times \mathbb{R}_+ \mid v \le (u)^{\frac{1}{k}}(x)^{\frac{k-1}{k}}\right\} \\
&= \text{Hypo}\{g\}
\end{aligned}$$

where Hypo$\{g\}$ is the hypograph of the function $g : \mathbb{R}^2_+ \mapsto \mathbb{R}_+$ defined by

$$g(x, u) = (u)^{\frac{1}{k}}(x)^{\frac{k-1}{k}}.$$

The function g is a concave monomial with positive rational powers that sum to 1. Therefore the hypograph of $g(x, u)$ is CQr [Alizedeh and Goldfarb, 2003, Ben-Tal and Nemirovski, 2001]. This proves that the functions $f_k(x, v)$ are CQr for all $k = 1, \ldots, d$ and thus Epi$\{f\}$ can be represented by a system of conic quadratic inequalities. □

Since Epi$_+\{f\}$ requires that a weighted summation of $f_1(x, v), \ldots, f_d(x, v)$ is less than u, we add d nonnegative auxiliary variables $t_1, \ldots, t_d$ such that

$$f_k(x, v) \le t_k \ \forall k = 1, \ldots, d; \ \text{ and } \ a_1t_1 + a_2t_2 + \cdots + a_{d-1}t_{d-1} + a_dt_d \le u.$$

If $a_k = 0$ for any $k = 1, \ldots, d$ then there is no need to add the auxiliary variable t_k or derive a CQR for $f_k(x, v)$. For this general model we assume that $a_k > 0$ for all k. Let $\Lambda_k(x, v, t_k, \tau_k)$ denote a set of conic quadratic inequalities that represents $(x, v, t_k) \in \text{Epi}\{f_k\}$ where x, v, t_k are variables in $\mathbb{R}$ defined previously and τ_k is a column vector in $\mathbb{R}^{\eta_k}$ of η_k nonnegative design variables required to obtain the CQR of Epi$\{f_k\}$. (We derive the actual CQIs in Section 3.2.2.) Then the conic quadratic representation of Epi$\{f\}$ is the system

$$\Lambda(x, v, u, t_1, \tau_1, \ldots, t_d, \tau_d) = \left\{\sum_{k=1}^{d} a_k t_k \leq u\right\} \ \& \ \bigcup_{k=1}^{d} \Lambda_k(x, v, t_k, \tau_k). \tag{3.7}$$

This means that $(x, v, u) \in$ Epi$\{f\}$ if and only if there exists $(t_1, \tau_1, \ldots, t_d, \tau_d)$ such that the system $\Lambda(x, v, u, t_1, \tau_1, \ldots, t_d, \tau_d)$ is satisfied. Therefore the CQR of Epi$_+\{f\}$ = Epi$\{f\} \cap \{x \geq 0\} \cap \{v \geq 0\} \cap \{u \geq 0\}$ is the system

$$\{x \geq 0\} \ \& \ \{v \geq 0\} \ \& \ \{u \geq 0\} \ \& \ \Lambda(x, v, u, t_1, \tau_1, \ldots, t_d, \tau_d) \tag{3.8}$$

3.2.2 Deriving CQIs Using the Binary Tree Method

Our derivation exploits the following characteristic of hyperbolic inequalities. Observation 3.5 stems directly from the CQR of the half cone

$$K_+^2 = \left\{(a, b, c) \in \mathbb{R}^3 : b, c \geq 0, 0 \leq a \leq \sqrt{bc}\right\}.$$

Observation 3.5. Any hyperbolic inequality in $\mathbb{R}^3$ of the form $a^2 \leq bc$ where $a, b, c \in \mathbb{R}_+$ can be formulated as the following second-order cone constraint:

$$\left\| \begin{pmatrix} 2a \\ b - c \end{pmatrix} \right\|_2 \leq b + c.$$

with $a \geq 0$ still enforced. Figure (3.1) illustrates how these two types of cones are related.

Proposition 3.6 (Aktürk et al. [2009]). *. For integral $k \geq 0$, there is an equivalent formulation to the inequalities*

$$(v)^k \leq t_k(x)^{k-1}, \qquad x, v, t_k \geq 0 \tag{3.9}$$

using $\mathcal{O}(\log_2 k)$ variables and $\mathcal{O}(\log_2 k)$ conic quadratic constraints.

Based on the work of Alizedeh and Goldfarb [2003] Aktürk et al. [2009] give an efficient representation of inequalities like (3.9) in $\mathbb{R}^3_+$ using a polynomial number of conic quadratic constraints. The method published by Alizedeh and Goldfarb [2003] is very different than the method discussed by Ben-Tal and Nemirovski [2001]; though both are built from the hypograph for a concave monomial and exploit the use of hyperbolic inequalities. Let $L = \min\{L \in \mathbb{Z}_+ : 2^L \geq k\}$. Alizedeh and Goldfarb [2003] demonstrate how to build the CQR of inequalities with rational powers using a binary tree with exactly $L+1$ levels. If $r = 2^L - k > 0$, multiplying each side of inequality (3.9) by $(v)^r$ results in

$$(v)^{2^L} \leq t_k(x)^{k-1}(v)^r = t_k(x)^{k-1}(v)^{2^L-k}. \tag{3.10}$$

This inequality is equivalent to the hypograph of the geometric mean of 2^L variables. There are exactly $L+1$ leaf nodes and thus at most $2L+2$ non-leaf nodes, each requiring the addition of 1 new design variable τ (except for the root node). Each non-leaf node represents a hyperbolic inequality formed between its corresponding design variable and the variables corresponding to the children nodes. The following procedure builds a full, inverted binary tree and converts the nodes into hyperbolic inequalities and then second order cone constraints in $\mathbb{R}^3$. To start we have an empty binary tree with $L+1$ levels. Leaf nodes are added to levels 0 through $L-1$. We construct the root node in level L. This node is associated with the variable v and corresponds to the left-hand side of (3.10), v^{2^L}.

3.2.2.1 Creating Leaf Nodes

The leaf nodes of the binary tree correspond to the variables in the right hand side of (3.10). Through the process of creating the leaf nodes we break the exponents on x and v into sums of powers of 2.

For each of the levels $\ell = 0, 1, \ldots, L-1$, add exactly one leaf node corresponding to either x^{2^ℓ} or v^{2^ℓ}.

Step 1: In order to decide which levels $\ell = 1, \ldots, L-1$ should have a node associated with x and which should have one associated with v we create a partition (J_x, J_v) of the set $\{0, 1, \ldots, L-2\}$ such that if $\ell \in J_x$, level ℓ contains a node corresponding to x^{2^ℓ} and similarly for J_v.

Step 2: To begin, let $J_x = J_v = \emptyset$. Define ℓ_1^x and ℓ_1^v as

$$\ell_1^x = \max\{\ell \in \mathbb{Z}_+ : 2^\ell \leq k-1\} = \lfloor \log_2(k-1) \rfloor,$$
$$\ell_1^v = \max\{\ell \in \mathbb{Z}_+ : 2^\ell \leq 2^L - k\} = \lfloor \log_2(2^L - k) \rfloor,$$

and update J_x and J_v such that $J_x = J_x \cup \{\ell_1^x\}$ and $J_v = J_v \cup \{\ell_1^v\}$. Add leaf nodes to level ℓ_1^x and ℓ_1^v corresponding to $x^{2^{\ell_1^x}}$ and $v^{2^{\ell_1^v}}$ respectively.

Step 3: Define ℓ_2^x and ℓ_2^v as

$$\ell_2^x = \max\left\{\ell \in \mathbb{Z}_+ : 2^\ell \leq k - 1 - \sum_{j \in J_x} 2^j\right\} = \max\left\{\ell \in \mathbb{Z}_+ : 2^\ell \leq k - 1 - 2^{\ell_1^x}\right\},$$

$$\ell_2^v = \max\left\{\ell \in \mathbb{Z}_+ : 2^\ell \leq 2^L - k - \sum_{j \in J_v} 2^j\right\} = \max\left\{\ell \in \mathbb{Z}_+ : 2^\ell \leq 2^L - k - 2^{\ell_1^v}\right\},$$

and again let $J_x = J_x \cup \{\ell_2^x\}$ and $J_v = J_v \cup \{\ell_2^v\}$. Add leaf nodes to levels ℓ_2^x and ℓ_2^v corresponding to $x^{2^{\ell_2^x}}$ and $v^{2^{\ell_2^v}}$ respectively.

Step 4: Continue this process of finding ℓ_i^x and ℓ_i^v as

$$\ell_i^x = \max\left\{\ell \in \mathbb{Z}_+ : 2^\ell \leq k - 1 - \sum_{j \in J_x} 2^j\right\},$$

$$\ell_i^v = \max\left\{\ell \in \mathbb{Z}_+ : 2^\ell \leq 2^L - k - \sum_{j \in J_v} 2^j\right\},$$

updating J_x and J_v, and adding nodes at the appropriate level until either ℓ_i^x or ℓ_i^v is equal to zero. (If k is even, ℓ_i^x will equal zero first, while ℓ_i^v will equal zero first if k is odd.)

Step 5: At this point, whichever set J_x or J_v containing zero is complete. The building process can continue for other set, or we can simply add the remaining elements of $\{0, 1, \ldots, L-1\}$ to the active set and add appropriate leaf nodes to the binary tree.

Step 6: Lastly, create a leaf node at level 0 corresponding to t_k.

Proposition 3.7. *Given $k > 0$ and $L = \max\{\ell \in \mathbb{Z}_+ : 2^\ell \geq k\}$ the above process* *forms a unique partition* (J_x, J_v) *of the set* $\{0, 1, \ldots, L-1\}$ *such that*

$$\sum_{\ell \in J_x} 2^\ell = k - 1 \qquad \textit{and} \qquad \sum_{\ell \in J_v} 2^\ell = 2^L - k. \tag{3.11}$$

Thus for all $k > 0$ *there is a unique binary tree that can be used to derive the CQR of* (3.9) *and at levels* $\ell = 0, \ldots, L-1$ *there is exactly one leaf node corresponding to either* x^{2^ℓ} *or* v^{2^ℓ}.

The proof for Proposition (3.7) is included in Appendix A.

Remark. The leaf node added to level $\ell = L-1$ must be associated with the variable x. Otherwise we have that $\ell = L-1 \in J_v$ which implies that $2^{L-1} \leq 2^L - k$. If this is true then $2^{L-1} \geq k$ and $L \neq \min\{L \in \mathbb{Z}_+ : 2^L \geq k\}$ which is a contradiction.

Figure 3.2 illustrates the leaf nodes created by the above procedure for the case $k = 3$.

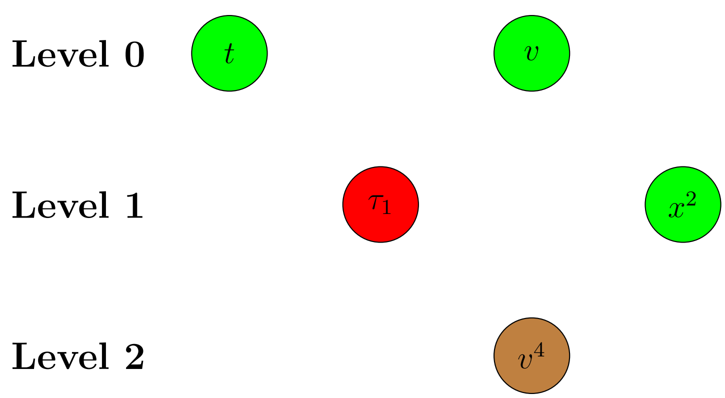

Figure 3.2: The Binary Tree Leaf Nodes case $k = 3$.

3.2.2.2 Connecting the Binary Tree

Before we start connecting the binary tree there are exactly 2 leaf nodes at level 0 and 1 leaf node at levels $\ell = 1, 2, \ldots, L-1$.

Starting at level 0, there are exactly two leaf nodes: one representing t_k and the other representing either x^1 if k is even or v^1 if k is odd. Connect the leaf-nodes via a parent node added to level 1. This non-leaf node represents the addition of a design variable $\tau_{k,1} \geq 0$ such that

$$\tau_{k,1}^2 \leq \begin{cases} t_k x & \text{if } k \text{ is even} \\ t_k v & \text{if } k \text{ is odd} \end{cases}.$$

Step 1: Move to level 1 where there are now two nodes: one non-leaf node representing $\tau_{k,1}$ and one leaf node corresponding to either x^2 or v^2. Connect these nodes by adding a parent node in level 2 along with the design variable $\tau_{k,2} \geq 0$ such that

$$\tau_{k,2}^2 \leq \begin{cases} \tau_{k,1} x & \text{if } \ell = 1 \in J_x \\ \tau_{k,1} v & \text{if } \ell = 1 \in J_v \end{cases}.$$

Step 2: Continue this process throughout the remaining $L-2$ layers adding one design variable $\tau_{k,\ell}$ to level ℓ at each step.

Step 3: At level $\ell = L-1$ there are exactly two nodes: one corresponding to the design variable $\tau_{k,L-1}$ and the other a leaf node representing $x^{2^{L-1}}$. Connect these two nodes to the root node which is associated with v^{2^L} such that

$$v^2 \leq \tau_{k,L-1} x.$$

Figure 3.3 gives an example of a binary tree built using the procedure above for the case $k=3$. Leaf nodes are green while the root node is brown and design variables are red.

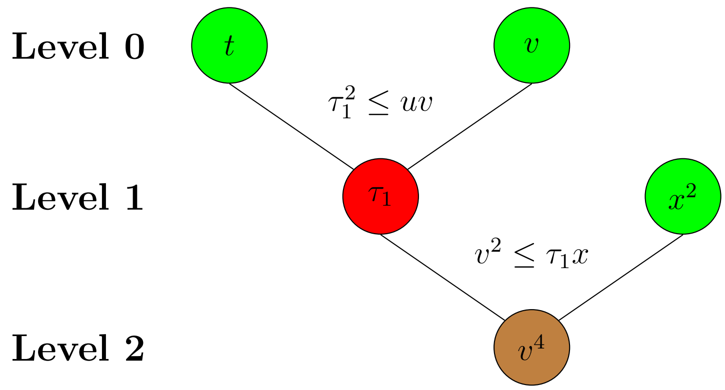

Figure 3.3: The Binary Tree Diagram used to derive CQR for the case $k = 3$.

3.2.2.3 Forming Conic Quadratic Inequalities

Once the binary tree has been completed, we can replace the hyperbolic inequalities currently in the binary tree with their conic quadratic equivalents as illustrated in Observation 3.5. By replacing $\Lambda_k(x, v, t_k, \tau_k)$ in the CQR system (3.7) with these conic quadratic inequalities we get the CQR of Epi$\{f_+\}$. For $k = 1, \ldots, d$ let $L_k = \min\{\ell \in \mathbb{Z}_+ : 2^\ell \geq k\}$. Then we can replace the system (3.8) with:

$$v \leq t_1 \qquad \text{(*The case when } k = 1\text{)} \tag{3.12a}$$

$$A_k \begin{pmatrix} x \\ v \\ t_k \\ \tau_k \end{pmatrix} \geq_{\mathbf{K}_k} 0 \qquad \forall k = 2, \ldots, d \tag{3.12b}$$

$$t_k \geq 0 \qquad \forall k = 1, \ldots, d \tag{3.12c}$$

$$\tau_k \in \mathbb{R}^{L_k - 1} \qquad \forall k = 2, \ldots, d \tag{3.12d}$$

$$\tau_{k,\ell} \geq 0 \qquad \forall k = 2, \ldots, d \;\; \ell = 1, \ldots L_k - 1 \tag{3.12e}$$

$$\sum_{k=1}^{d} a_k t_k \leq u \tag{3.12f}$$

$$x, v, u \geq 0 \tag{3.12g}$$

The cone $\mathbf{K}_k$ in constraint (3.12b) is the direct product of L_k Lorentz cones $\mathbf{L}^3$. The matrix A_k is an extraordinarily sparse $3L_k \times L_k + 2$ matrix for all $k = 2, \ldots, d$. There are only $5L_k$ nonzero elements in A_k while there are $3L_k^2 + 6L$ zeros. This matrix also holds for each k, regardless of i or j. A detailed description of the columns of A_k is included in Appendix B.

Now that we have a conic quadratic representation for the constraints (3.5) we expand the system (3.12a)–(3.12g) for all $i = 1, \ldots, m$ and $j = 1, \ldots, n_i$. The indices of t_k, and $\tau_{k,1}, \ldots, \tau_{k,L_k-1}$ are updated to $t_{i,j,k}$, and $\tau_{i,j,k,1}, \ldots, \tau_{i,j,k,L_k-1}$ for $i = 1, \ldots, m$, $j = 1, \ldots, n_i$ and $k = 1, \ldots, d_i$. Since L_k and A_k are determined solely by k and independent of i and j their indices remain the same. We also leave $\mathbf{K}_k$ indexed only by k as it is the direct product of L_k Lorentz cones $\mathbf{L}^3$, and therefore independent of i and j. We can now replace the nonlinear constraints in the recourse function (3.3f) with the following:

$$
\begin{aligned}
& v_{i,j} \leq t_{i,j,1} && \forall i = 1, \ldots, m,\ j = 1, \ldots, n_1,\ k = 1 \\
& A_k \begin{pmatrix} x_j^i \\ v_{i,j} \\ t_{i,j,k} \\ \tau_{i,j,k} \end{pmatrix} \geq_{\mathbf{K}_k} 0 && \forall i = 1, \ldots, m,\ j = 1, \ldots, n_1,\ k = 2, \ldots, d_i \\
& \sum_{k=1}^{d_i} a_k^{i,j} t_{i,j,k} \leq u_{i,j} && \forall i = 1, \ldots, m,\ j = 1, \ldots, n_i \\
& t_{i,j,k} \geq 0 && \forall i = 1, \ldots, m,\ j = 1, \ldots, n_i,\ k = 1, \ldots, d_i \\
& \tau_{i,j,k} \in \mathbb{R}_{\geq 0}^{L_k - 1} && \forall i = 1, \ldots, m,\ j = 1, \ldots, n_i,\ k = 2, \ldots, d_i
\end{aligned}
$$

Therefore the second stage problem $Q(x)$ can be written:

$$
\begin{aligned}
Q(x) = \min_{\substack{\tau,t,u,v,\\ w,y,z}} & \sum_{i=1}^{m} w_i \\
\text{s.t.} & By + Cz \geq d \\
& z \geq 0 \\
& y \in \mathbb{R}^m_+ \\
& y_i = \sum_{j=1}^{n_i} v_{i,j}, && 1 \leq i \leq m \\
& w_i = \sum_{j=1}^{n_i} u_{i,j} && 1 \leq i \leq m \\
& v_{i,j} \leq t_{i,j,1} && 1 \leq i \leq m,\ 1 \leq j \leq n_i,\ k = 1 \\
& A_k \begin{pmatrix} x^i_j \\ v_{i,j} \\ t_{i,j,k} \\ \tau_{i,j,k} \end{pmatrix} \geq_{\mathbf{K}_k} 0 && 1 \leq i \leq m,\ 1 \leq j \leq n_i,\ 2 \leq k \leq d_i \\
& \sum_{k=1}^{d_i} a^{i,j}_k t_{i,j,k} \leq u_{i,j} && 1 \leq i \leq m,\ 1 \leq j \leq n_i \\
& t_{i,j,k} \geq 0 && 1 \leq i \leq m,\ 1 \leq j \leq n_i,\ 1 \leq k \leq d_i \\
& \tau_{i,j,k} \in \mathbb{R}^{L_k - 1}_{\geq 0} && 1 \leq i \leq m,\ 1 \leq j \leq n_i,\ 2 \leq k \leq d_i \\
& v_{i,j} \geq 0,\ u_{i,j} \geq 0, && 1 \leq i \leq m,\ 1 \leq j \leq n_i
\end{aligned}
\tag{3.13}
$$

3.3 Separability of First and Second Stage Variables

This formulation of $Q(x)$ minimizes a linear objective over a convex feasible region found by intersecting linear constraints with the Cartesian product of second-order cones. Though this is an improved formulation of $Q(x)$, the second order cone constraints (3.13) involve the product of first and second stage variables which makes the second stage nonseparable. For $i = 1, \ldots, m$, $j = 1, \ldots, n_i$, $k = 2, \ldots, d_i$ let $\left(J_x^k, J_v^k\right)$ be the unique partition of $\{0, 1, \ldots, L_k - 1\}$ found during the binary tree method where $L_k = \min\{\ell \in \mathbb{Z}_+ : 2^\ell \geq k\}$. Then the SOCP system (3.13) includes $|J_x^k|$ conic quadratic nonseparable inequalities for $i = 1, \ldots, m$, $j = 1, \ldots, n_i$, $k = 2, \ldots, d_i$.

$$\text{If } k \text{ is even, i.e. } \ell = 0 \in J_x^k, \qquad \left\| \begin{pmatrix} 2\tau_{i,j,k,1} \\ t_{i,j,k} - x_j^i \end{pmatrix} \right\|_2 \leq t_{i,j,k} + x_j^i$$

$$\text{for all } \ell \in J_x^k \setminus \{0, L_k - 1\}, \qquad \left\| \begin{pmatrix} 2\tau_{i,j,k,\ell+1} \\ \tau_{i,j,k,\ell+1} - x_j^i \end{pmatrix} \right\|_2 \leq \tau_{i,j,k,\ell+1} + x_j^i$$

$$\text{Since } \ell = L_k - 1 \in J_x^k, \qquad \left\| \begin{pmatrix} 2v_{i,j} \\ \tau_{i,j,k,L_k-1} - x_j^i \end{pmatrix} \right\|_2 \leq \tau_{i,j,k,L_k-1} + x_j^i$$

The hyperbolic inequality formulations of the above constraints shows clearly how the first and second stage variables are nonseparable.

$$\text{If } k \text{ is even, i.e. } \ell = 0 \in J_x^k, \qquad \tau_{i,j,k,1}^2 \leq t_{i,j,k} \cdot x_j^i$$

$$\text{for all } \ell \in J_x^k \setminus \{0, L_k - 1\}, \qquad \tau_{i,j,k,\ell+1}^2 \leq \tau_{i,j,k,\ell} \cdot x_j^i$$

$$\text{Since } \ell = L_k - 1 \in J_x^k, \qquad v_{i,j}^2 \leq \tau_{i,j,k,L_k-1} \cdot x_j^i$$

To fix this separability problem we add a nonnegative auxiliary variable $\chi_{i,j}$ for all $i = 1, \ldots, m,\ j = 1, \ldots, n_i$ such that

$$0 \le \chi_{i,j} \le x^i_j \qquad \forall i = 1, \ldots, m,\ j = 1, \ldots, n_i$$

and replace occurrences of x^i_j in the SOCP constraints (3.13) with $\chi_{i,j}$. Therefore we can write a final reformulation of $Q(x)$ that is linear in first stage variables x, convex in all second stage variables $\tau, \chi, t, u, v, w, y, z$, and does not contain the product of x with any second stage variable.

$$Q(x) = \min_{\substack{\tau,\chi,t,u,\\ v,w,y,z}} \sum_{i=1}^{m} w_i \tag{3.14a}$$

$$\text{s.t. } By + Cz \ge d \tag{3.14b}$$

$$z \ge 0 \tag{3.14c}$$

$$y \in \mathbb{R}^m_+ \tag{3.14d}$$

$$y_i = \sum_{j=1}^{n_i} v_{i,j}, \qquad 1 \le i \le m \tag{3.14e}$$

$$w_i = \sum_{j=1}^{n_i} u_{i,j} \qquad 1 \le i \le m \tag{3.14f}$$

$$v_{i,j} \le t_{i,j,1} \qquad 1 \le i \le m,\ 1 \le j \le n_i,\ k = 1 \tag{3.14g}$$

$$A_k \begin{pmatrix} \chi_{i,j} \\ v_{i,j} \\ t_{i,j,k} \\ \tau_{i,j,k} \end{pmatrix} \ge_{\mathbf{K}_k} 0 \qquad 1 \le i \le m,\ 1 \le j \le n_i,\ 2 \le k \le d_i \tag{3.14h}$$

$$\sum_{k=1}^{d_i} a^{i,j}_k t_{i,j,k} \le u_{i,j} \qquad 1 \le i \le m,\ 1 \le j \le n_i \tag{3.14i}$$

$$0 \le \chi_{i,j} \le x^i_j \qquad 1 \le i \le m,\ 1 \le j \le n_i \tag{3.14j}$$

$$t_{i,j,k} \ge 0 \qquad 1 \le i \le m,\ 1 \le j \le n_i,\ 1 \le k \le d_i \tag{3.14k}$$

$$\tau_{i,j,k} \in \mathbb{R}^{L_k - 1}_{\ge 0} \qquad 1 \le i \le m,\ 1 \le j \le n_i,\ 2 \le k \le d_i \tag{3.14l}$$

$$v_{i,j},\ u_{i,j} \ge 0 \qquad 1 \le i \le m,\ 1 \le j \le n_i \tag{3.14m}$$

Chapter 4

Mean-Risk Stochastic Programming Model

4.1 Risk-Neutral Stochastic Programming

Recall the general form of a two-stage SP (2.1) from Chapter 2.

$$\min_{x \in X} \mathbb{E}\left[F(x,\omega)\right]$$

In general, stochastic programming is risk neutral as it takes the expectation of cost over the set of all scenarios $\omega \in \Omega$ [Shapiro et al., 2009]. Consider the above minimization and let x^*, z^* be the optimal solution and values respectively. Then,

$$z^* = \min_{x \in X} \mathbb{E}\left[F(x,\omega)\right] = \mathbb{E}\left[F(x^*,\omega)\right]$$

It should be notes that $F(x^*,\omega)$ is a random variable representing cost. Since z^* is the expected value of this random variable, for a given realization of $F(x^*,\omega)$ the probability that $F(x^*, \omega) \geq z^*$ is 0.5. Furthermore,

$$P\left[F(x^*, \omega) \geq z^*\right] = P\left[F(x^*, \omega) \leq z^*\right] = 0.5.$$

To illustrate how the general two-stage SP returns a risk-neutral solution, let $\underline{F}, \bar{F}$ be the infimum and supremum respectively of $F(x^*, \omega)$ over $\omega \in \Omega$.

$$\underline{F} = \inf_{\omega \in \Omega} F(x^*, \omega) \qquad \bar{F} = \sup_{\omega \in \Omega} F(x^*, \omega).$$

Then the probability that a realization of the random cost variable $F(x^*, \omega)$ falls inside the interval $[\underline{F}, z^*]$ (respectively $[z^*, \bar{F}]$) is 0.5.

$$P\left[\underline{F} \leq F(x^*, \omega) \leq z^*\right] = P\left[z^* \leq F(x^*, \omega) \leq \bar{F}\right] = 0.5.$$

If the random cost variable $F(x^*, \omega)$ has a normal distribution, Figure (4.1) illustrates the relation between the behavior of the probability density function of $F(x^*, \omega)$, cost in the best case $\underline{F}$, cost in the worst case $\bar{F}$, and $z^* = \mathbb{E}\left[F(x^*, \omega)\right]$.

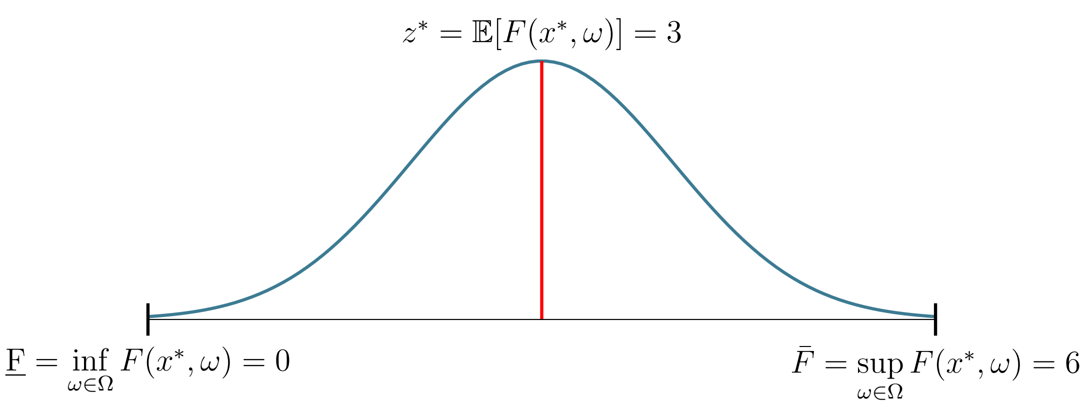

Figure 4.1: An example distribution for $F(x^*, \omega)$, $F(x^*, \omega) \sim Normal(3, 1)$.

On the other hand, this looks much different if the probability density function of $F(x^*, \omega)$ is instead skewed far right. In this case the worst case cost $\bar{F} = \sup_{\omega \in \Omega} F(x^*, \omega)$ is much higher than $\mathbb{E}[F(x^*, \omega)]$ and the probability $F(x^*, \omega)$ falls in the interval $[\bar{F} - \epsilon, \bar{F}]$ ($\epsilon > 0$, small) is much smaller than the probability $F(x^*, \omega)$ is within ϵ of $z^* = \mathbb{E}[F(x^*, \omega)]$. Realizations in the interval $[\bar{F} - \epsilon, \bar{F}]$ are considered extreme events. The expectation in general stochastic programming gives a solution x^* that preforms well in the long-term with different (more probable) realizations of $\omega \in \Omega$ but can result in severe costs in the case of an extreme event. Figure (4.2) illustrates an instance in which the probability density of $F(x^*, \omega)$ is highly skewed right and results in extreme events.

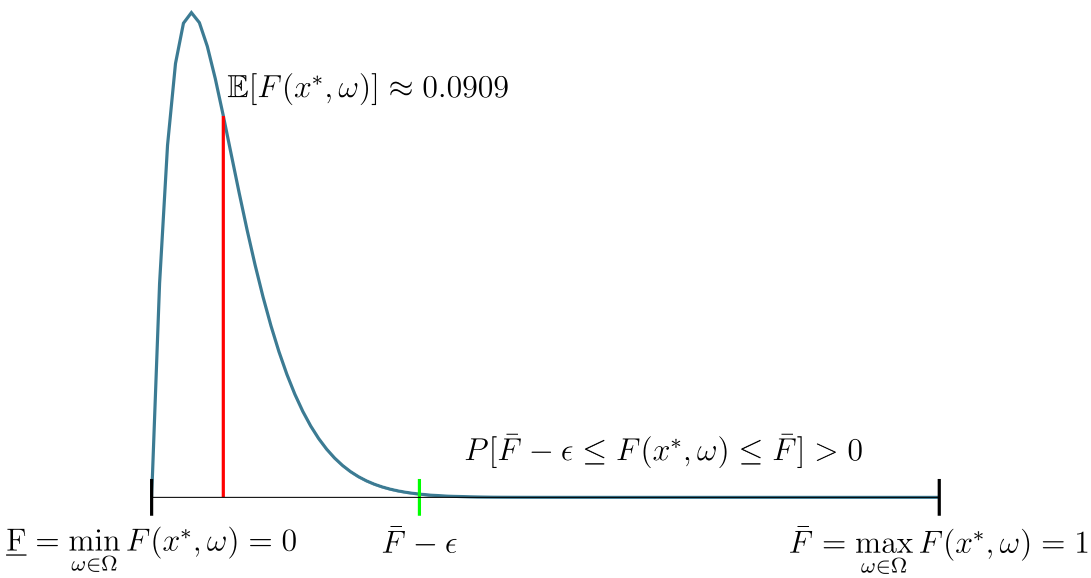

Figure 4.2: An example distribution for $F(x^*, \omega)$, $F(x^*, \omega) \sim Beta(2, 20)$.

Robust optimization is an alternative to risk neutral stochastic programming that considers the far less probable worst case scenario. This method results in conservative decisions that help hedge against the drastic impacts of extreme events and can also be useful when probability distributions are unknown. Since these events are rare, robust optimization makes more costly decisions to prepare for a realization that most likely will not come to fruition. Robust optimization also works well when distributions are unknown. To balance our need for economic solutions that also protect our network against extreme events we must consider how risk varies with different decisions [Ben-Tal et al., 2009].

4.2 Introducing Risk into Optimization Models

In order to increase risk averseness in the optimal solutions of stochastic programs risk measures can be incorporated that quantify aspects of the uncertainty involved in the random cost variable $F(x^*, \omega)$. Risk measures are real-valued functions whose domain is the set of random cost variables $F(x, \omega)$ feasible for $x \in X$ Cotton and Ntaimo [2015]. The risk measure $\rho : \{F(x, \omega)\}_{x \in X} \to \mathbb{R}$ can be included in the objective of the general two-stage SP to form a mean-risk stochastic program. A weight factor $\lambda > 0$ is used to control the trade-off between the risk measure ρ and the expectation originally included. Hence, the mean-risk two-stage SP can be written as

$$\min_{x \in X} \mathbb{E}\left[F(x, \omega)\right] + \lambda \rho \left[F(x, \omega)\right] \tag{4.1}$$

Larger values of λ give more consideration to the cost of ρ and resulting in solutions x^* associated with random cost variables $F(x^*, \omega)$ that have smaller variances and thus include less uncertainty. If $\lambda = 0$, (5.1) returns to the risk-neutral form.

It should be noted random variables $F(\bar{x}, \omega)$ are in the domain of ρ if and only if $F(\bar{x}, \omega) \in \{F(x, \omega) : x \in X\}$ which is equivalent to requiring $\bar{x} \in X$. Each element of $\{F(x, \omega) : x \in X\}$ corresponds to an instance of $x \in X$. Given the cost function of $F(x, \omega) : \mathbb{R}^m \times \Omega \to \mathbb{R}$, we can denote the risk measure $\rho(F(x, \omega))$ as $\rho(x)$ for all $x \in X$ due to the fact that $F(x, \cdot)$ is the random variable of interest Noyan [2012].

4.3 Commonly Used Risk Measures

Risk measures can be divided into two types: quantile measures that are defined using the a quantile of the probability distribution of $F(x^*, \omega)$ and deviation measures that compute the expected deviation of $F(x^*, \omega)$ from a given target. With each of these categories, some risk measures also fall into the class of coherent risk measures. Coherent risk measures posses certain properties that can be exploited for easier implementation and computing. These properties are as follows [Artzner et al., 1999]:

(C1) *Convexity:* For all $x, x' \in X$ and $\alpha \in [0, 1]$,

$$\rho\left(\alpha x + (1-\alpha)x'\right) \leq \alpha\rho(x) + (1-\alpha)\rho(x').$$

(C2) *Monotonicity:* If $x, x' \in X$ with $F(x, \cdot) : \Omega \to \mathbb{R}$, $F(x', \cdot) : \Omega \to \mathbb{R}$ such that $F(x, \cdot) \geq F(x', \cdot)$, then $\rho\left(F(x, \omega)\right) = \rho(x) \geq \rho(x') = \rho\left(F(x', \omega)\right)$.

(C3) *Translation Invariance:* For $\alpha \in \mathbb{R}$ and $x \in X$,

$$\rho\left(F(x, \omega) + \alpha\right) = \rho(x + \alpha) = \rho(x) + \alpha = \rho\left(F(x, \omega)\right) + \alpha$$

(C4) *Positive Homogeneity:* If $\alpha \in \mathbb{R}$ such that $\alpha \geq 0$ then

$$\rho\left(\alpha F(x, \omega)\right) = \rho\left(\alpha x\right) = \alpha\rho\left(x\right) = \alpha\rho\left(F(x, \omega)\right).$$

These properties are important in our ability to easily use risk measures as they do not disrupt the linearity and convexity of existing models. We now present examples of commonly used deviation and then quantile measures. For each risk measure we define the function ρ as well as make note of important characteristics.

4.3.1 Deviation Measures

Let $\eta \in \mathbb{R}$ be a target, acceptable value of $F(x, \omega)$. *expected excess* (EE, ρ_{EE_η}) measures the expected excess of $F(x, \omega)$ above the target η [Märkert and Schultz, 2005].

$$\rho_{EE_\eta}(x) := \mathbb{E}\left[\max\{F(x, \omega) - \eta, 0\}\right] \tag{EE}$$

Expected excess is a deviation measure commonly used in stochastic programs that model decisions made in the electricity production market [Carrion, 2008, Carrión et al., 2009, Schultz and Neise, 2006]. As electricity producers and retailers aim to control the balance of supply and demand in the market expected excess helps to estimate the amount of surplus given the targeted demand η.

If we set the target η to be the mean of $F(x, \omega)$ (i.e., $\eta = \mathbb{E}[F(x, \omega)$ then EE turns into another deviation measure, *absolute semideviation* (ASD, ρ_{ASD}) [Ogryczak and Ruszczynski, 2002].

$$\rho_{ASD}(x) := \mathbb{E}\left[\max\{F(x, \omega) - \mathbb{E}[F(x, \omega)], 0\}\right] \tag{ASD}$$

For a given $x \in X$, the absolute semideviation of the random cost variable $F(x, \omega)$ is its expected excess above its own mean.

4.3.2 Quantile Measures

Similar to expected excess, *excess probability* (EP, ρ_{EP_η}) [Schultz and Tiedemann, 2003] is a quantile measure of the probability $F(x, \omega)$ exceeds a target $\eta \in \mathbb{R}$.

$$\rho_{EP_\mu}(x) := P\left[\{\omega \in \Omega \; : \; F(x,\omega) > \eta\}\right]. \tag{EP}$$

For a given $x \in X$, the excess probability of $F(x,\omega)$ is the total probability of all realizations $\bar{\omega} \in \Omega$ such that $F(x,\bar{\omega}) > \eta$.

Given a confidence level $\alpha \in [0,1]$ *Value-at-risk* (VaR, ρ_{VaR_α}) [Artzner et al., 1999] is measured as

$$\mathrm{VaR}_\alpha(x) = \rho_{\mathrm{VaR}_\alpha}(x) := \min\left\{\eta \; : \; \mathbb{P}\left[F(x,\omega) \le \eta\right] > \alpha\right\}. \tag{VaR}$$

This value is the lower α-quantile, meaning the the probability of $F(x,\omega)$ taking a lower value is α, and the probability of a greater cost is $1-\alpha$. Value-at-Risk has often been criticized for its difficult use as it is nonconvex [Krokhmal et al., 2013].

Based on VaR, Conditional Value-at-Risk (CVaR, ρ_{CVaR_α}) [Pflug, 2000] measures the expected value of the $1-\alpha$ worst scenarios. In consideration of our attempt to minimize cost, this is the expected value of the $1-\alpha$ greatest costs. Given confidence level α, and setting η to be the value of $VaR_\alpha(F(x,\omega))$, CVaR is calculated as

$$\begin{aligned}\mathrm{CVaR}_\alpha(x) = \rho_{\mathrm{CVaR}_\alpha}(x) &:= \min\left\{\bar{\eta} \; : \; \frac{1}{1-\alpha}\mathbb{E}\left[\max\left\{0, F(x,\omega) - \bar{\eta}\right\}\right]\right\} \\ &= \eta + \frac{1}{1-\alpha}\mathbb{E}\left[\max\left\{0, F(x,\omega) - \eta\right\}\right]. \end{aligned} \tag{CVaR}$$

With the exception of Value-at-risk (which is a highly nonconvex function making it difficult to incorporate into models) all of the above measures are categorized as coherent risk measures.

4.4 Implementing Conditional Value-at-Risk

CVaR has been widely used in stochastic models beginning in financial portfolio models [Rockafellar and Uryasev, 2000] and eventually begin used in engineering applications such as disaster management [Noyan et al., 2017]. When incorporating CVaR into our current model with discretized probability distribution the first stage becomes

$$\begin{aligned}
&\min_x \; c^T x + \sum_{s\in S} p_s Q^s(x) + \lambda \rho_{CVaR_\alpha}\left[F(x,\omega)\right] \\
= &\min_x \; c^T x + \sum_{s\in S} p_s Q^s(x) + \lambda CVaR_\alpha\left[c^T x + \mathbb{E}[Q(x,\omega)]\right] \\
= &\min_x \; c^T x + \sum_{s\in S} p_s Q^s(x) + \lambda c^T x + \lambda\left(\eta + \frac{1}{1-\alpha}\sum_{s\in S} p_s \max\left\{Q^s(x)-\eta, 0\right\}\right) \\
= &\min_x \; (1+\lambda) c^T x + \sum_{s\in S} p_s Q^s(x) + \lambda\left(\eta + \frac{1}{1-\alpha}\sum_{s\in S} p_s \max\left\{Q^s(x)-\eta, 0\right\}\right)
\end{aligned} \tag{4.2a}$$

$$\text{s.t. } x = (x^1, x^2, \ldots, x^m) \in X \tag{4.2b}$$

$$x^i \in \{0,1\}^{n_i} \cap \text{SOS-1} \qquad \forall i = 1,\ldots,m. \tag{4.2c}$$

As risk measures are functions on the probability distribution in which the second stage exists, calculating CVaR has no effect on the structure of the second stage.

The term $\max\{Q^s(x)-\eta, 0\}$ in (4.2a) is nonlinear and therefore must be removed from the objective. For all scenarios $s \in S$, we introduce the auxiliary variable ξ_s such that

$$\xi_s \geq 0 \qquad \forall s \in S$$

$$\xi_s \geq Q^s(x) - \eta \qquad \forall s \in S.$$

The above constraints require that the value of ξ_s that minimizes the objective always be the maximum of $Q^s(x) - \eta$ and zero. Therefore the mean risk formulation of our two-stage stochastic program is:

$$
\begin{aligned}
\min_{x} \;\; & (1+\lambda)c^T x + \sum_{s \in S} p_s Q^s(x) + \lambda \left(\eta + \frac{1}{1-\alpha} \sum_{s \in S} p_s \xi_s \right) && \text{(Mean-risk SP)} \\
\text{s.t. } & x = (x^1, x^2, \ldots, x^m) \in X \\
& x^i \in \{0,1\}^{n_i} \cap \text{SOS-1} && \forall i = 1, \ldots, m \\
& \xi_s \geq 0 && \forall s \in S \\
& \xi_s \geq Q^s(x) - \eta && \forall s \in S
\end{aligned}
$$

where $Q^s(x)$ is defined by (3.14).

Chapter 5

Decomposition and Algorithm for Solution

5.1 Challenges in Mixed Integer Nonlinear Programming

In Chapter 3, we derived a convex reformulation for the recourse function $Q^s(x)$ that linearly separates first and second stage variables, thus $Q^s(x)$ is convex in x and can be approximated by supporting hyperplanes. Using this formulation of $Q^s(x)$ (3.14) we can write a convex mixed integer nonlinear program (MINLP) for the mean risk model .

$$\min_{\substack{x \\ \tau,\chi,t,u, \\ v,w,y,z}} (1+\lambda)c^T x + \sum_{s\in S} p_s \left(\sum_{i=1}^{m} w_{s,i}\right) + \lambda \left(\eta + \frac{1}{1-\alpha}\sum_{s\in S} p_s \xi_s\right) \tag{5.1}$$

$$\text{s.t. } x = (x^1, x^2, \ldots, x^m) \in X \tag{5.2}$$

$$x^i \in \{0,1\}^{n_i} \cap \text{SOS-1} \qquad \forall i = 1,\ldots,m \tag{5.3}$$

$$\xi_s \geq 0 \qquad \forall s \in S \tag{5.4}$$

$$\xi_s \geq \sum_{i=1}^{m} w_{s,i} - \eta \qquad \forall s \in S \tag{5.5}$$

$$\text{(3.14b)–(3.14m)} \qquad \forall s \in S$$

Though $Q^s(x)$ can be solved efficiently for a fixed x, MINLPs like the above model can quickly become very expensive to solve, as the number of binary variables increases [Nemhauser and Wolsey, 1988]. MINLPs are difficult to solve due to their combinatorial and continuous domains. Many algorithmic methods have been developed in order to solve MINLPs more efficiently. Branch and bound [Gupta and Ravindran, 1985], outer approximation [Fletcher and Leyffer, 1994] and generalized Benders decomposition [Benders, 2005, Floudas, 1995, Geoffrion, 1972] are three such methods that work by creating a sequence of non-increasing upper bounds as well as a sequence of non-decreasing lower bounds that converge on the optimal solution.

For problems such as (5.1), we refer to the first stage decision variables $x \in X \subset \{0,1\}^n$ as complicating variables. It should be noted that for a fixed $\bar{x}$, the second stage problems $Q^s(\bar{x})$ and $Q^{s'}(\bar{x})$ for differing scenarios $s, s' \in S$ are independent of each other. Thus given a fixed $\bar{x} \in X$, (5.1) can be decomposed into $|S|$ independent SOCPs. Lobo et al. [1998] have shown how the primal-dual potential reduction method [Nesterov and Nemirovskii, 1994] can be applied to SOCPs. GBD

[Geoffrion, 1972] exploits this structure by decomposing difficult MINLPs into smaller subproblems. We first give an overview of Generalized Benders decomposition and then discuss how this can lead us to develop a solution algorithm for our model.

5.2 Generalized Benders Decomposition

Geoffrion [1972] generalized the original Benders decomposition [Benders, 2005] for a class of problems that includes the following general 0-1 MINLP case:

$$
\begin{aligned}
\min_{x,y}\ & f(x,y) \\
\text{s.t. } & h(x,y) = 0 \\
& g(x,y) \leq 0 \\
& x \in X = \{0,1\}^n \\
& y \in Y \subset \mathbb{R}^m
\end{aligned}
\tag{5.6}
$$

with $Y \neq \emptyset$ convex such that for all fixed $x \in X = \{0,1\}^n$: $f(x,\cdot) : \mathbb{R}^m \to \mathbb{R}$ and $g(x,\cdot) : \mathbb{R}^m \to \mathbb{R}^{p_1}$ are convex; $h(x,\cdot) : \mathbb{R}^m \to \mathbb{R}^{p_2}$ is linear; the set $Z_x = \{z \in \mathbb{R}^{p_1} : h(x,y) = 0,\ g(x,y) \leq z,\ y \in Y\}$ is closed; and (5.6) either has an optimal solution or is unbounded. These conditions guarantee that the strong duality theorem is satisfied.

Generalized Benders decomposition solves problems like (5.6) iteratively by generating an upper bound and a lower bound for the solution at each step. We first separate our minimization by variable and project (5.6) onto the x-space.

$$\begin{aligned} \min_x \ \inf_y \ & f(x,y) \\ \text{s.t. } & h(x,y) = 0 \\ & g(x,y) \leq 0 \\ & x \in X = \{0,1\}^n \\ & y \in Y \subset \mathbb{R}^m \end{aligned}$$

For our problems of interest, the inner problem is bounded for a given $x \in X$; thus, the inner problem can be updated as a minimization over y. The inner problem forms the primal problem:

$$\begin{aligned} P(\bar{x}) \ = \ \min_{y \in Y} \ & f(\bar{x}, y) \\ \text{s.t. } & h(\bar{x}, y) = 0 \\ & g(\bar{x}, y) \leq 0 \end{aligned}$$

whose solution gives an upper bound for (5.6). Solving $P(\bar{x})$ also provides the optimal solution $y*$ and Lagrange multipliers ν_*, μ_* corresponding to the equality and inequality constraints respectively. We form a relaxed master problem

$$\begin{aligned} RM \ = \ \min_{x \in X} \ & \phi_{LB} \\ \text{s.t } & \phi_{LB} \geq \min_{x \in X} \mathcal{L}(x, y, \nu, \mu) \end{aligned}$$

where $\mathcal{L}(x, y, \nu, \mu)$ is the Lagrange dual function of the primal problem. The optimal ϕ_{LB} is a lower bound for (5.6) while the optimal solution x^* gives a new fixed point that is used in the primal problem at the next iteration.

5.2.1 Algorithmic Statement of GBD

Step 1: Pick an initial point $x^1 \in X \cap V$. Solve the primal problem $P(x^1)$ and denote the optimal solution y^1. Let ν_1, μ_1 be the corresponding Lagrange dual multipliers. Set the iteration counter $I = 1$, initial upper bound $\text{UBD}_I = P(x^1)$, and convergence tolerance $\epsilon \geq 0$.

Step 2: Solve the relaxed master problem:

$$\begin{aligned} RM^I \;=\; & \min_{x \in X} \; \phi_{LB} \\ & \text{s.t. } \phi_{LB} \geq \min_{x \in X} \left\{ f(x, y^\iota) + \nu_\iota^T h(x, y^\iota) + \mu_\iota^T g(x, y^\iota) \right\}, \qquad \iota = 1, \ldots, I \end{aligned}$$

for optimal solution (x, ϕ_{LB}) which we record as (x^{I+1}, ϕ_{LB}). Set the initial lower bound as $\text{LBD} = \phi_{LB}$. If $\text{UBD} - \text{LBD} \leq \epsilon$ the algorithm terminates and (x^{I+1}, ϕ_{LB}) is the optimal solution of (5.6). Let $I = I + 1$

Step 3: Solve $P(x^I)$ for optimal solution y^I and dual multipliers ν_I and μ_I. Update the upper bound to be $\text{UBD}_I = \min\{\text{UBD}_{I-1}, P(x^I)\}$. If $\text{UBD}_I - \text{LBD}_{I-1} \leq \epsilon$ then terminate with (x^I, y^{I-1}) as the optimal solution of (5.6), otherwise return to step 2.

Figure 5.1 depicts a schematic representation of the algorithmic process of the GBD method discussed above.

5.3 A Variation of GBD Under Separability

There are several variants of the Generalized Benders Decomposition based on assumptions made about (5.6). One such variant that is particularly helpful in this

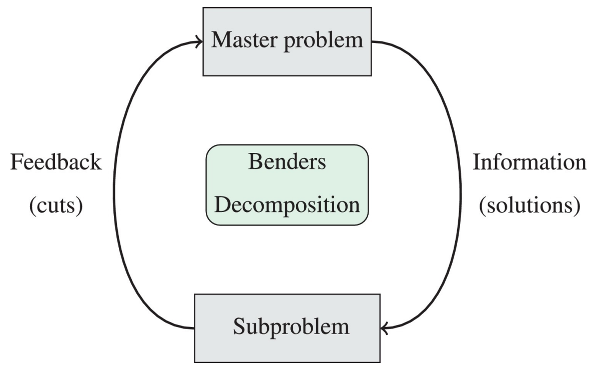

Figure 5.1: A Visual Illustration of the GBD Method [Rahmaniani et al., 2017].

research was denoted by Geoffrion as Property P [Geoffrion, 1972]. This particular variant of the GBD holds for certain classes of problems such as those with linearly separable objective and constraints. We refer to use of Property P on this class of problems are Property P under separability.

Theorem 5.1 (Floudas [1995])**.** *If the objective function f, equality constraint function h, and inequality constraint function g are linearly separable in x and y the following property holds. For every μ, $\nu \geq 0$, the infimum of $L(x, y, \mu, \nu)$ with respect to $y \in Y$ can be taken independently of x so that the support function $\mathcal{L}(x; \mu, \nu)$ can be obtained explicitly with little or no more effort than is required to evaluate it at a single value of x.*

Problems like (5.6) that meet the necessary conditions for Property P have

$$\begin{aligned} f(x,y) &= f_1(x) + f_2(y), \\ h(x,y) &= h_1(x) + h_2(y), \\ g(x,y) &= g_1(x) + g_2(y). \end{aligned}$$

Then we can write the support function $\mathcal{L}(x, y, \nu_\iota, \mu_\iota)$ explicitly as

$$\begin{aligned}
\mathcal{L}(x, y, \nu_I, \mu_I) &= \min_{x \in X} \left\{ f(x, y^I) + \nu_\iota^T h(x, y^I) + \mu_I^T g(x, y^I) \right\} \\
&= \min_{y \in Y} \left\{ f_1(x) + f_2(y) + {\nu_I}^T \left[h_1(x) + h_2(y) \right] + {\mu_I}^T \left[g_1(x) + g_2(y) \right] \right\} \\
&= f_1(x) + {\nu_I}^T h_1(x) + {\nu_I}^T g_1(x) + \min_{y \in Y} \left\{ f_2(y) + {\mu_I}^T h_2(y) + {\mu_I}^T g_2(y) \right\} \\
&= f_1(x) + {\nu_I}^T h_1(x) + {\nu_I}^T g_1(x) + f_2(y^I) + {\mu_I}^T h_2(y^I) + {\mu_I}^T g_2(y^I)
\end{aligned}$$

Under these conditions the primal problem $P(x^I)$ takes the form:

$$\begin{aligned}
P(x^I) \;=\; & \min_{y \in Y} \; f_1(x^I) + f_2(y) & \qquad & \min_{y \in Y} \; f_2(y) \\
& \text{s.t. } h_1(x^I) = -h_2(y) & & \text{s.t. } h_1(x^I) = -h_2(y) \\
& \quad g_1(x^I) \leq -g_2(y) & & \quad g_1(x^I) \leq -g_2(y)
\end{aligned}$$

Therefore the algorithmic procedure for the GBD method with Property P under separability is as follows:

5.3.1 GBD with Property P under separability

Step 1: Pick an initial point $x^1 \in X \cap V$. Solve the primal problem $P(x^1)$ and denote the optimal solution y^1. Let ν_1, μ_1 be the corresponding Lagrange dual multipliers. Set the iteration counter $I = 1$, initial upper bound $\text{UBD}_I = P(x^1)$, and convergence tolerance $\epsilon \geq 0$.

Step 2: Solve the relaxed master problem:

$$\begin{aligned} RM^I \;=\; & \min_{x \in X}\; \phi_{LB} \\ & \text{s.t. } \phi_{LB} \geq f_1(x) + {\nu_\iota}^T h_1(x) + {\nu_\iota}^T g_1(x) + L_\iota, \qquad \iota = 1, \ldots, I \end{aligned}$$

where $L_\iota = \min_{y \in Y}\left\{f_2(y^\iota) + {\mu_\iota}^T h_2(y^\iota) + {\mu_\iota}^T g_2(y^\iota)\right\} = f_2(y^\iota) + {\mu_\iota}^T h_2(y^\iota) + {\mu_\iota}^T g_2(y^\iota)$. for optimal solution (x, ϕ_{LB}) which we record as (x^{I+1}, ϕ_{LB}). Set the initial lower bound as $\text{LBD} = \phi_{LB}$. If $\text{UBD} - \text{LBD} \leq \epsilon$ the algorithm terminates and (x^{I+1}, ϕ_{LB}) is the optimal solution of (5.6). Let $I = I + 1$

Step 3: Solve $P(x^I)$ for optimal solution y^I and dual multipliers ν_I and μ_I. Update the upper bound to be $\text{UBD}_I = \min\{\text{UBD}_{I-1}, P(x^I)\}$. If $\text{UBD}_I - \text{LBD}_{I-1} \leq \epsilon$ then terminate with (x^I, y^{I-1}) as the optimal solution of (5.6), otherwise return to step 2.

5.4 Decomposition Method for the Mean-Risk Model

We previously noted that for a fixed $\bar{x} \in X$, (5.1) can be decomposed into a mixed 0-1 linear master problem and an SOCP subproblem for each scenario $s \in S$. Given $\bar{x}$ the subproblems can solved efficiently; however, these second stage costs $Q^s(\bar{x}) = \sum_{i=1}^{m} w_{s,i}$ appear in the objective of the master problem. In order to minimize over all possible decisions, we must see how these costs vary as a function of x, which complicates the structure. Therefore we want to iteratively approximate these values (along with the constraint $\xi_s \geq Q^s(x) - \eta$ for all $s \in S$) using Benders cuts. We replace instances of $Q^s(x)$ remaining in the objective with auxiliary variable ϕ_s for all $s \in S$ which we use in cut generation.

We can solve the master problem (5.1) using a branch-and-cut approach that relaxes the integrality requirement on x. At integer nodes in the branch-and-cut tree we have a fixed integer $\bar{x}$ with which we can solve the subproblems and generate supporting hyperplanes that lower approximate the function $Q^s(x)$. After finding $Q^s(\bar{x})$ for all $s \in S$, η can be calculated using these values along with p_s for all $s \in S$. We bound ϕ_s and ξ_s below using these lower approximations which are found by creating minimization problems analogous to the relaxed master problems used in the property (P) variant of GBD. By approximating $Q^s(x)$ for each $s \in S$ we using what are known as disaggregate cuts as a opposed to one optimality cut that would approximate $\sum_{s \in S} p_s Q^s(x)$. These cuts form a multi-cut approach which provides the first stage with more information and thus may improve the number of iterations needs to solve Birge and Louveaux [1988, 2011]. As previously stated, since it is assumed that (5.1) has complete recourse, no feasibility cuts are needed.

Remark. To avoid repetitive information during the decomposition process we denote the second stage variables for a given scenario $s \in S$ as

$$\mathbf{y_s} = (y_s, z_s, w_s, v_s, u_s, t_s, \tau_s, \chi_s). \tag{5.7}$$

Recall that the only constraint in $Q^s(x)$ that contains both first and second stage variables is

$$0 \leq \chi_{i,j} \leq x_j^i \qquad \forall\ i = 1, \dots, m\ j = 1, \dots, n_i$$

We separate all other constraints into feasible sets $\mathcal{X}$ and $\mathcal{Y}_\mathbf{s}$ for the first stage variables and second stage variables of scenario $s \in S$ respectively.

$$\mathcal{X} = \left\{ x \in \{0,1\}^n : \begin{array}{cc} x = (x^1, x^2, \ldots, x^m) \in X & \\ x^i \in \{0,1\}^{n_i} & i = 1, \ldots, m \\ \sum_{j=1}^{n_i} x^i_j = 1 & i = 1, \ldots, m \\ \xi_s \geq 0 & \end{array} \right\}$$

$$\mathcal{Y}_\mathbf{s} = \left\{ \mathbf{y_s} : \begin{array}{ll} By + Cz \geq d & z \geq 0, \\ \forall i = 1, \ldots, m : & \sum_{j=1}^{n_i} v_{ij} = y_i, \ \sum_{j=1}^{n_i} v_{ij} = y_i, \ y_i \geq 0 \\ \forall i, \ \ \forall j = 1, \ldots, n_i : & v_{ij}, u_{ij}, \chi_{ij} \geq 0 \ t_{i,j,1} \geq 0 \\ & \sum_{k=1}^{d_i} a^{i,j}_k t_{i,j,k} \leq u_{i,j}, \ v_{i,j} \leq t_{i,j,1} \\ \forall i, \ \forall j, \ \ \forall k = 2, \ldots, d_i : & t_{i,j,k} \geq 0, \ A_{i,j,k} \begin{pmatrix} \chi_{i,j} \\ v_{i,j} \\ t_{i,j,k} \\ \tau_{i,j,k} \end{pmatrix} \geq_{\mathbf{K}_k} 0 \\ & L_k = \min_{\ell \in \mathbb{Z}_+} \{2^\ell \geq k\} \\ \forall i, \ \forall j, \ \forall k, \ \ \ \forall \ell = 1, \ldots, L_k - 1 & \tau_{i,j,k} \in \mathbb{R}^{L_k - 1}_{\geq 0} \end{array} \right\}$$

To apply the property (P) variant of GBD we must define for fixed $\bar{x}$ a primal problem analogous to $P(\bar{x})$. Given first stage feasible decision $\bar{x}$, for all $s \in S$ we define the primal problem $P_s(\bar{x})$ as:

$$\begin{aligned} P_s(\bar{x}) \ = \ Q^s(\bar{x}) \ = \ \min_{\mathbf{y_s}} \ & \sum_{i=1}^{m} w_i \\ \text{s.t.} \ & \chi_{ij} \leq x^i_j \qquad\qquad \forall i = 1, \ldots, m, \ j = 1, \ldots, n_i \\ & \mathbf{y_s} \in \mathcal{Y}_\mathbf{s} \end{aligned}$$

Notice that with $\bar{x}$ fixed, the objective is only in terms of second stage variables.

Thus, the linearly separated form of the objective is:

$$f(x, \mathbf{y_s}) = f_1(s) + f_2(\mathbf{y_s}) = 0 + \sum_{i=1}^{m} w_i = Q^s(\bar{X})$$

Similarly the inequality constraint can be rewritten and then linearly separated as:

$$g^{ij}(x, \mathbf{y_s}) = \chi_{ij} - x_j^i = g_1^{ij}(x) + g_2^{ij}(\mathbf{y_s}) \leq 0$$

with $g_1^{ij}(x) = -x_j^i$ and $g_2^{ij}(\mathbf{y_s}) = \chi_{ij}$. We can condense these inequalities in $\mathbb{R}$ into a vector inequality $g(x, \mathbf{y_s}) = g_1(x) + g_2(\mathbf{y_s}) \leq 0$ in $\mathbb{R}^n$.

We solve the primal problem (or second stage) and obtain the optimal solution $\mathbf{y_s}^*$, as well as the optimal dual multiplier associated with inequality constraint $\chi_{ij} \leq x_j^i$, which we denote $\mu_{s,i,j}^*$. We can condense these multipliers into the vector $\mu_s^* \in \mathbb{R}^n$. Recall from the property (P) variant of GBD that the lower approximating hyperplane (the right-hand side of the inequality constraint) is $f_1(x) + \mu_s^{*T} g_1(x) + L_*$ where $L_* = \min_{\mathbf{y_s} \in \mathcal{Y}_\mathbf{s}} \left\{ f_2(\mathbf{y_s}) + \mu_s^T g_2(\mathbf{y_s}) \right\}$. Notice that we do not need to explicitly solve for L_*, as strong duality gives us that the solution to this minimization is the same as the optimal solution $\mathbf{y_s}^*$ found for $Q^s(\bar{x})$. Additionally, we have that for any optimal solution of $Q^s(\bar{x})$, $\chi_{i,j} = \bar{x}_j^i$ for all $i = 1, \ldots, m$ $j = 1, \ldots, n_i$. Thus the lower approximation of $Q^s(x)$ by supporting hyperplanes is

$$\begin{aligned} f_1(x) + \mu_s^{*T} g_1(x) + f_2(\mathbf{y_s}^*) + \mu_s^{*T} g_2(\mathbf{y_s}^*) &= 0 + \mu_s^{*T}(-x) + Q^s(\bar{x}) + \mu_s^{*T} \chi_s \\ &= Q^s(\bar{x}) + \mu_s^{*T} (\bar{x} - x) \\ &= Q^s(\bar{x}) + \sum_{i=1}^{m} \sum_{j=1}^{n_i} \mu_{s,i,j}^* \left(\bar{x}_j^i - x_j^i \right). \end{aligned}$$

Therefore the relaxed master problems for each $s \in S$ are

$$
\begin{aligned}
RM_{\phi_s} \;=\; & \min_{x \in \mathcal{X}} \; \phi_s \\
& \text{s.t. } \phi_s \geq Q^s(\bar{x}) + \mu_s^{*T}\left(\bar{x} - x\right) = Q^s(\bar{x}) + \sum_{i=1}^{m}\sum_{j=1}^{n_i} \mu_{s,i,j}^{*}\left(\bar{x}_j^i - x_j^i\right)
\end{aligned}
$$

$$
\begin{aligned}
RM_{\xi_s} \;=\; & \min_{x \in \mathcal{X}} \; \xi_s \\
& \text{s.t. } \phi_s \geq Q^s(\bar{x}) + \mu_s^{*T}\left(\bar{x} - x\right) - \eta = Q^s(\bar{x}) + \sum_{i=1}^{m}\sum_{j=1}^{n_i} \mu_{s,i,j}^{*}\left(\bar{x}_j^i - x_j^i\right) - \eta
\end{aligned}
$$

where η is the Value-at-Risk of the second stage problems and is calculated as $Q^s(\bar{x})$ is found for all $s \in S$.

5.5 Solution Algorithm

We now give the solution algorithm for solving the model:

$$
\begin{aligned}
\min_{\substack{x,\eta, \\ \mathbf{y_s},\phi_s,\xi_s \\ \forall s \in S}} \; & (1+\lambda)c^T x + \sum_{s \in S} p_s \phi_s + \lambda\left(\eta + \frac{1}{1-\alpha}\sum_{s \in S} p_s \xi_s\right) \\
\text{s.t. } & x \in \mathcal{X} \\
& \mathbf{y_s} \in \mathcal{Y}_\mathbf{s} && \forall s \in S \\
& \chi_s \leq x && \forall s \in S \\
& \xi_s \geq 0 && \forall s \in S
\end{aligned}
$$

with given weight factor $\lambda \geq 0$ and risk level α.

Step 1: Set iteration counter $I = 0$ and begin building a branch-and-cut tree with which to solve the following master problem.

$$\begin{aligned}
&\min_{x,\eta,\phi,\xi} (1+\lambda)c^T x + \sum_{s\in S} p_s\phi_s + \lambda\left(\eta + \frac{1}{1-\alpha}\sum_{s\in S} p_s\xi_s\right) \\
&\text{s.t. } x \in \mathcal{X} \\
&\quad \xi_s \geq 0 \qquad\qquad \forall s \in S
\end{aligned}$$

When an integer node is found, let $I = I + 1$ and denote the current optimal solution to the master problem as $(\bar{x}^I, \bar{\eta}, \bar{\phi}^I, \bar{\xi}^I)$. Solve the second stage problem $Q^s(\bar{x}^I)$ for all $s \in S$ and let μ_s^I denote the optimal dual multiplier associated with the one mixed constraint $\chi_s \leq x$. .

Step 2: For all $s \in S$

- If $\bar{\phi}_s^I < Q^s(\bar{x}^I)$, add the following optimality cut to the master problem (all existing nodes in the branch-and-cut tree)

$$\text{Cut 1:} \qquad \phi_s \geq Q^s(\bar{x}^I) + \mu_s^{I^T}\left(\bar{x}^I - x\right).$$

- If $\bar{\xi}_s^I < Q^s(\bar{x}^I) - \bar{\eta}$, add the following optimality cut to the master problem (all existing nodes in the branch-and-cut tree)

$$\text{Cut 2:} \qquad \xi_s \geq Q^s(\bar{x}^I) + \mu_s^{I^T}\left(\bar{x}^I - x\right) - \eta.$$

Step 3: Prune the current integer node and continue solving the master problem with added cut constraints. When another integer node is reached, let $I = I + 1$ and denote the current optimal solution to the master problem as $(\bar{x}^I, \bar{\eta}, \bar{\phi}^I, \bar{\xi}^I)$. Solve the second stage problem $Q^s(\bar{x}^I)$ for all $s \in S$ and let μ_s^I denote the optimal dual multiplier associated with the one mixed constraint $\chi_s \leq x$. Return to Step 2.

Chapter 6

A Mean-Risk Program for Transportation Network Protection

6.1 Introduction to Problem

In the United States, transportation networks are some of the most vulnerable critical infrastructure systems. Together these networks stretch more than 4 million miles of roads and make up the Highway and Motor Carrier subset of the Department of Homeland Security's Transportation Systems critical infrastructure sector. In their 2017 infrastructure report, the American Society of Civil Engineers reported that American's spent 6.9 billion hours in traffic delays in 2014 resulting in 3.1 billion gallons of fuel wasted (worth $160 billion). Due to a pattern of under-funding, the back log of highway and bridge capital needs exceeds $836 billion dollars [of Civil Engineers, 2017]. Despite the United State's dependency on the networks in this massive system, the Department of Transportation (DOT) received less that 3% of the 2019 federal budget to maintain these and many other infrastructure systems.

In this example we use the mean-risk two stage stochastic model to study transportation network protection. A transportation network can be modeled as a directed graph $G(N, A)$ with nodes $(i \in N)$ representing location of significance and edges $(a \in A)$ representing roads connecting two locations. Given information includes the subset of critical edges $\bar{A} \subsetneq A$, edge capacities c_a/c_{ij}, normal flow demand d^{rs} from some origin node $r \in O$ to another node $s \in D$, free flow speed of each link f_a, a set of possible retrofitting strategies for critical edges $h \in H$ with known cost b_a^h, and a given budget b_0. For a set of possible future scenarios $k \in K$ of probability p_k, we are given or generate the ratio of expected remaining capacity for edge a given strategy h, $\theta_a^{h,k}$.

The first stage decision is a binary vector u_a^h that selects a retrofitting strategy $h \in H$ for critical edge $a \in \bar{A}$. Exactly one strategy can be selected for each critical edge. The second stage variables deal with how demand flows across the network. The units of flow across edge $a \in A$ due to demand from node $r \in O$ to node $s \in D$ is denoted x_a^{rs} and must be nonnegative and obey flow balance constraints. The aggregate of these flow is v_a the total flow across edge $a \in A$. A table listing the complete notation used in this example is included in Appendix C.

6.2 Optimization Parameters

6.2.1 First Stage Parameters

The objective in the first stage is to minimize the investment cost due to retrofitting plus the mean-risk expectation of the second stage. Therefore the objective of the first stage is

$$\min \ (1+\lambda)b^T u + \sum_{k \in K} p_k Q^k(u) + \lambda \left(\eta + \frac{1}{1-\alpha} \sum_{k \in K} p_k \max[Q^k(u) - \eta, 0] \right). \tag{6.1}$$

where η is the Value-at-Risk of the second stage. The only constraints to the first stage are binary and budgetary.

$$\sum_{a \in \bar{A}} \sum_{h \in H} b_a^h u_a^h \leq b_0$$

$$\sum_{h \in H} u_a^h = 1 \qquad \forall a \in \bar{A}$$

$$u_a^h \in \{0, 1\} \qquad \forall a \in \bar{A},\ h \in H$$

We refer to the above constraints collectively as the set $u \in U$.

In order to linearize the calculation for CVaR in (6.1) we replace $\max\{Q^k(u) - \eta, 0\}$ with auxiliary variable ξ_k and enforce the following constraints:

$$\xi_k \geq 0 \qquad \forall k \in K$$

$$\xi_k \geq Q^k(u) - \eta \qquad \forall k \in K$$

6.2.2 Second Stage Parameters

During the second stage we encounter more complicated equations. The objective is to minimize the total cost of flow over the network. To calculate this cost we use the Bureau of Public Roads (BPR) [of Public Roads, 1964] function for link travel time per unit flow which uses a ratio of flow across the link over capacity of the link. Because vulnerable edges in $\bar{A}$ have an updated capacity in the second stage based on the chosen retrofitting strategy and scenario we must reflect this choices in the denominator.

$$\forall a \in A \setminus \bar{A}: \qquad f_a \left[1 + \zeta \left(\frac{v_a}{c_a} \right)^4 \right]$$

$$\forall a \in \bar{A}: \qquad f_a \left[1 + \zeta \left(\frac{v_a}{c_a \sum_{h \in H} \theta_a^h u_a^h} \right)^4 \right]$$

If the ratio of flow to capacity is greater than one the cost begins to grow by the power of 4. Once this is multiplied by the flow over the link and our parameter γ which translates travel time to monetary value we are given the objective of the second stage.

$$Q^k(u) = \min_{x,v} \gamma \left[\sum_{a \in A} f_a v_a^k + \zeta \left(\sum_{a \in A \setminus \bar{A}} \frac{f_a}{c_a^4} (v_a^k)^5 + \sum_{a \in \bar{A}} \frac{f_a}{c_a^4} \frac{(v_a^k)^5}{(\sum_{h \in H} \theta_a^{h,k} u_a^h)^4} \right) \right] \tag{6.2}$$

The first constraints for the second stage deal with the flow demand over the network. We can break these constraints into four prongs: nonnegativity, flow balance at each node in N, flow balance supply from each node in O, and flow demand to each node in D. For simplicity all of the following constraints make up the set $x \in X$ in later formulations.

At the end of the second stage there must be no existing supply or demand for flow at any node. Each node begins with a positive or negative value of flow depending on it being a resource for flow or a sink. These values must all be zero at the end of the stage.

$$\sum_{j \in \delta^+(i)} x_{ij}^{rs,k} - \sum_{j \in \delta^-(i)} x_{ji}^{rs,k} = 0 \qquad \forall (r,s) \in \mathcal{OD},\ i \in N \setminus \{r,s\}$$

A supply of d^{rs} units of flow originate at node $r \in N$ and need to travel to node $s \in N$. Regardless of where these units of flow actually end up, node $r \in N$ must push out d^{rs} more units of flow than it receives in order to push all of this flow along.

$$\sum_{j \in \delta^+(r)} x_{rj}^{rs,k} - \sum_{j \in \delta^-(r)} x_{jr}^{rs,k} = d^{rs} \qquad \forall (r,s) \in \mathcal{OD}$$

Node $s \in N$ has a demand for d^{rs} units of flow from node $r \in N$. The amount of flow pushed into node $s \in N$ (due to OD-pair (r, s)) needs to be d^{rs} units of flow less than what it pushes out for demand to be met.

$$\sum_{j \in \delta^+(s)} x_{sj}^{rs,k} - \sum_{j \in \delta^-(s)} x_{js}^{rs,k} = -d^{rs} \qquad \forall (r,s) \in \mathcal{OD}$$

Finally negative flow is not possible in our model and thus we must restrict x to positive values.

$$x_a^{rs,k} \geq 0 \qquad \forall (r,s) \in \mathcal{OD} \ \forall a \in A$$

Remark. Our constraints require that all flow demand be met, regardless of link capacity. We are able to make this generalization due to the way the objective function increases when flow surpasses current capacity. This forces the model to distribute flow to any link with remaining capacity before overloading a link.

The flow over each edge due to demand in an origin destination pair is then added to calculate the aggregate flow over each link, v_a.

$$\sum_{(r,s) \in \mathcal{OD}} x_a^{rs,k} = v_a^k \qquad \forall a \in A$$

Therefore, for each scenario $k \in K$ and given a value for u the recourse function is:

$$Q^k(u) = \min_{x,v} \ \gamma \left[\sum_{a \in A} f_a v_a^k + \zeta \left(\sum_{a \in A \setminus \bar{A}} \frac{f_a}{c_a^4} (v_a^k)^5 + \sum_{a \in \bar{A}} \frac{f_a}{c_a^4} \frac{(v_a^k)^5}{(\sum\limits_{h \in H} \theta_a^{h,k} u_a^h)^4} \right) \right] \tag{6.3}$$

$$\text{s.t. } x^k \in X$$

$$\sum_{(r,s) \in \mathcal{OD}} x_a^{rs,k} = v_a^k \qquad \forall a \in A$$

6.3 Reformulation of Second Stage

The second stage (6.3) poses a nonlinear, nonconvex, and non-separable problem. As demonstrated in Chapter 3 we must reformulate the recourse function in order to decompose and solve.

Remark. During the reformulation process of the recourse function, we simplify notation by focusing on a single $Q^k(u)$ for a single scenario $k \in K$. The only given parameter that is affected by changing the scenario is the value of the expected capacity ratio $\theta_a^{h,k}$. The notation $Q(u)$ refers to the general second stage problem and the index k is excluded from all variables,parameters and constraints.

6.3.1 Separating First and Second Stage Variables

To deal with the nonlinearity of the objective and the non-separability we replace each instance of v_a^5 in the objective with a nonnegative auxiliary variable y_a and add the constraints:

$$y_a \geq v_a^5 \qquad \forall a \in A \setminus \bar{A} \tag{6.4}$$

$$y_a \geq \frac{v_a^5}{\left(\sum_{h \in H} \theta_a^h u_a^h\right)^4} \qquad \forall a \in \bar{A}. \tag{6.5}$$

Lemma 6.1 (Lu et al. [2017]). *For $a \in \bar{A}$, $u \in U$, $\left(\sum_{h \in H} \theta_a^h u_a^h\right)^4 = \sum_{h \in H} (\theta_a^h)^4 u_a^h$.*

Proof. Since $\sum\limits_{h \in H} u_a^h = 1$ and $u_a^h \in \{0, 1\}$, it is always the case that for each $a \in \bar{A}$ there is some $h' \in H$ such that $u_a^{h'} = 1$ and $u_a^h = 0 \ \forall h \in H \setminus \{h'\}$. Therefore $\left(\sum_{h \in H} \theta_a^h u_a^h\right)^4 = \sum_{h \in H} (\theta_a^h)^4 u_a^h = (\theta_a^{h'})^4$. □

Adding the auxiliary variables y_a makes the objective function of $Q(u)$ linear, and by applying Lemma (6.1) to (6.5) we get

$$y_a \geq \frac{v_a^5}{\sum_{h \in H} (\theta_a^h)^4 u_a^h} \Longrightarrow v_a^5 \leq \left[\sum_{h \in H} (\theta_a^h)^4 u_a^h\right] y_a \qquad \forall a \in \bar{A}. \tag{6.6}$$

As shown above, we can multiply y_a in the first inequality of (6.6) by the denominator of the right hand side forming a inequality involving only positive exponents. For constraints associated with stable edges $a \in A \setminus \bar{A}$, nonlinearity is now the main problem. However, as in Chapter 3, the constraints added for critical edges cause a disjunction in the feasible set of second stage solutions we now remedy using perspective reformulation as demonstrated in Section (3.1.1).

6.3.2 Convexifying the Union of Disjunctive Sets

For vulnerable edges $a \in \bar{A}$, we must find the convex hull of the union of the sets formed by each possible value of u_a^h. The inequalities defining each set (6.6), are convex constraints, therefore we are formulating the convex hull of a union of convex set.

Remark. For brevity, we focus on the disjunctions for a single vulnerable edge $a \in \bar{A}$ therefore the index a is excluded, as the disjunction for each edge is the same. For the remainder of constraints, we also denote $(\theta^h)^4 = \beta_h$ since this is a constant value and allows us to simplify constraints.

The feasible set of solutions associated with each strategy $h \in H$ is

$$P_h = \left\{(u, v, y) \in \{0,1\}^{|H|} \times \mathbb{R}_+ \times \mathbb{R}_+ \ : \ v^5 \leq \beta_h y; \ u = e_h\right\}$$

The inequality constraint in P_h can be rewritten as the following function being bounded below zero

$$p_h(v, y) \leq 0 \text{ where } p_h : \mathbb{R}^2_+ \mapsto \mathbb{R}, \ p_h(v, y) = v^5 - \beta_h y.$$

The perspective function of $p_h(v, y)$ is defined as

$$\tilde{p}_h(v, y) = \begin{cases} \frac{v^5}{\lambda_h^4} - \beta_h y & \lambda_h > 0 \\ \infty & \text{otherwise} \end{cases}.$$

Therefore, a solution $(u, v, y) \in conv\left(\bigcup_{h \in H} P_h\right)$ if and only if there exists some $(\{w_h\}_h, \{z_h\}_h) \in \mathbb{R}^{|H|}_+ \times \mathbb{R}^{|H|}_+$ such that the following set is feasible Ceria and Soares [1999]:

$$\Pi = conv(\bigcup_{h \in H} P_h) = \left\{ (u, v, \{w_h\}_h, y, \{z_h\}_h) \ : \ \begin{array}{c} \sum_{h \in H} w_h = v, \ \sum_{h \in H} z_h = y, \\ \forall h \in H \ w_h \leq u_h^4 \beta_h z_h \\ \forall h \in H \ w_h \geq 0, \ z_h \geq 0, \ u_h \geq 0 \end{array} \right\}.$$

When the above constraints are added to the recourse function for each critical edge $a \in \bar{A}$ the set of feasible solutions to $Q(u)$ becomes a closed convex set. The addition of these constraints also adds $2|H|$ nonnegative auxiliary variables for each critical edge: $\{w_h\}_h$ and $\{z_h\}_h$. w_h and z_h represent the flow included in v and y respectively from choosing strategy $h \in H$. When $u_h = 0$ the value of w_h is forced to be zero and z_h is free. Since the summation of z_h over H must equal y which is included in the objective the minimization problem pushes z_h to zero.

Remark. The perspective reformulation Π has again introduced the binary first stage decision variable being raised to an integer power. This exponent is important in the convexification of the set Π as it allows u_h to take non-integer values between 0 and 1. However, since the recourse function $Q(u)$ is based on a given first stage decision $u \in U$ (that is always SOS-1) we are able to again apply Lemma (6.1) which would eliminate the exponent of binary variable. Be that as it may, the second-order cone representation of the inequality constraints from Π require the addition of less auxiliary variables when we leave the exponent on u_a^h. Therefore we add the following constraints to $Q(u)$

$$\begin{aligned}
&\sum_{h \in H} w_{a,h} = v_a && \forall a \in \bar{A} \\
&\sum_{h \in H} z_{a,h} = y_a && \forall a \in \bar{A} \\
&w_h \leq \beta_{a,h} z_{a,h} \left(u_a^h\right)^4 && \forall a \in \bar{A},\ h \in H \qquad (6.7) \\
&w_{a,h}, z_{a,h} \geq 0 && \forall a \in \bar{A},\ h \in H \\
&\beta_{a,h} = (\theta_a^h)^4 && \forall a \in \bar{A},\ h \in H.
\end{aligned}$$

6.3.2.1 Second Order Cone Constraints

Constraints (6.4) and (6.7) have been added to $Q(u)$ and are convex but they are not linear which poses problems for efficient computation. In order to make the second stage easier to compute we reformulate these inequalities into second-order cone constraints. Constraints (6.4) and (6.7) are inequalities in $\mathbb{R}^2$ and $\mathbb{R}^3$

respectively with rational powers, a type of function for which there are many simple SOCP formulation techniques like those mentioned in Section (3.2). We use the binary tree method described by Alizedeh and Goldfarb [2003] as it often results in the addition of few auxiliary design variables and constraints.

SOCP Reformulation for Stable Edges For (6.4) we are describing the epigraph of a 1-dimensional function $f(v) = v^5$. Let $p = 5$ as it is the highest power in the inequality. $\ell = \min\{\ell : 2^\ell \geq p\} = 3$. Therefore our binary tree diagram has at most 3 levels. We begin by multiplying each side of the inequality by $v^{2^\ell - p} = v^3$.

$$v_a^8 \leq y_a v_a^3 \tag{6.8}$$

The resulting binary tree (Figure 6.1) adds two nonnegative auxiliary variables ($s_{a,1}$ and $s_{a,2}$) and three levels of hyperbolic inequalities:

$$\begin{aligned} s_{a,1}^2 &\leq y_a v_a \\ s_{a,2}^2 &\leq s_{a,1} v_a \\ v_a^2 &\leq 1 \cdots s_{a,2}. \end{aligned}$$

By working from the bottom level up we can demonstrate how these inequalities are equivalent to (6.8).

$$\begin{aligned} v_a^2 \leq 1 \cdot s_{a,2} &\Longrightarrow v_a^4 \leq 1 \cdot s_{a,2}^2 = s_{a,1} v_a \\ &\Longrightarrow v_a^8 \leq s_{a,1}^2 v_a^2 = y_a v_a v_a^2 = y_a v_a^3 \\ &\Longrightarrow v_a^8 \leq y_a v_a^3. \end{aligned}$$

These hyperbolic inequalities can easily be translated into their conic quadratic equivalents using the method described in Observation (3.5).

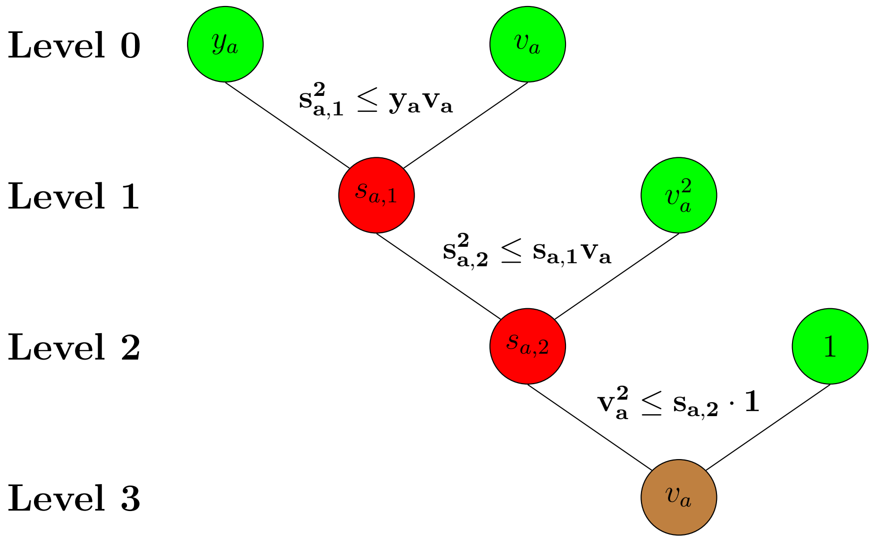

Figure 6.1: The Binary tree created for SOCP constraints of stable edges $a \in A \setminus \bar{A}$.

SOCP Reformulation for Critical Edges We begin by multiplying each side of constraint (6.7) by $w_{a,h}^3$ and grouping the constant $\beta_{a,h}$ with the variable that is raised to 1, $z_{a,h}$.

$$w_{a,h}^8 \leq (\beta_{a,h} z_{a,h})(u_a^h)^4 w_{a,h}^3.$$

As in the case of stable edges we add two auxiliary variables $s_{a,h,1}$, $s_{a,h,2}$ and 3 constraints in the binary tree for each edge (Figure 6.2) The hyperbolic version of the CQR is:

$$s_{a,h,1}^2 \leq (\beta_{a,h} z_{a,h}) w_{a,h} \tag{6.9}$$

$$s_{a,h,2}^2 \leq s_{a,h,1} w_{a,h} \tag{6.10}$$

$$w_{a,h}^2 \leq s_{a,h,2} u_a^h. \tag{6.11}$$

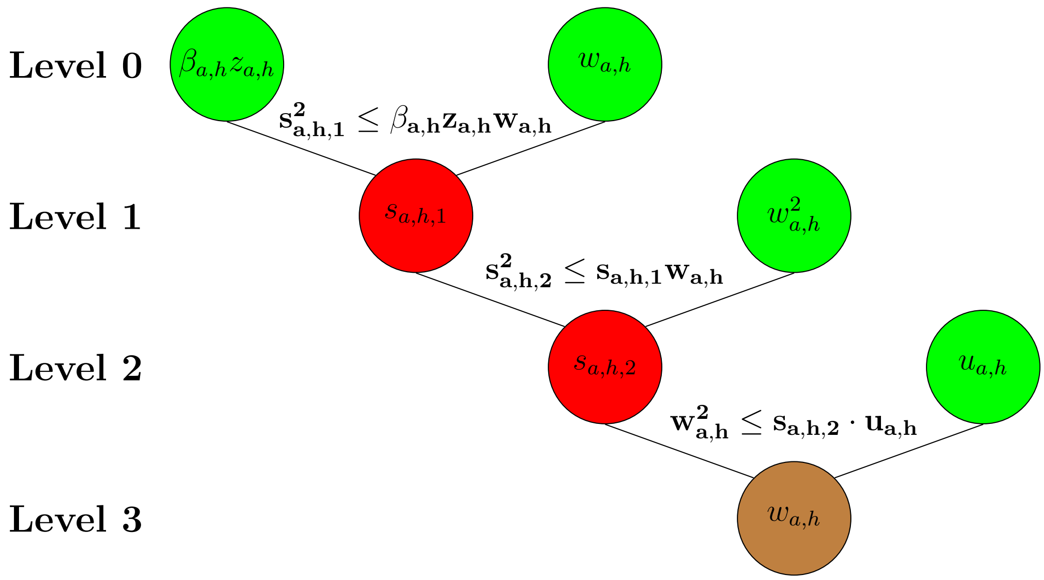

Figure 6.2: The Binary tree created for SOCP constraints of critical edges $a \in \bar{A}$, $\forall h \in H$.

Therefore the entirety of second-order constraints added to $Q(u)$ is:

$$s^2_{a,1} \leq y_a v_a \;\Leftrightarrow\; \left\| \begin{pmatrix} 2s_{a,1} \\ y_a - v_a \end{pmatrix} \right\|_2 \leq y_a + v_a \qquad \forall a \in A \setminus \bar{A}$$

$$s^2_{a,2} \leq s_{a,1} v_a \;\Leftrightarrow\; \left\| \begin{pmatrix} 2s_{a,2} \\ s_{a,1} - v_a \end{pmatrix} \right\|_2 \leq s_{a,1} + v_a \qquad \forall a \in A \setminus \bar{A}$$

$$v^2_a \leq 1 \cdot s_{a,2} \;\Leftrightarrow\; \left\| \begin{pmatrix} 2v_a \\ 1 - s_{a,2} \end{pmatrix} \right\|_2 \leq 1 + s_{a,2} \qquad \forall a \in A \setminus \bar{A}$$

$$s_{a,1}, s_{a,2} \geq 0 \qquad \forall a \in A \setminus \bar{A}$$

$$s^2_{a,h,1} \leq (\beta_{a,h}z_{a,h})w_{a,h} \;\Leftrightarrow\; \left\| \begin{pmatrix} 2s_{a,h,1} \\ (\beta_{a,h}z_{a,h}) - w_{a,h} \end{pmatrix} \right\|_2 \leq (\beta_{a,h}z_{a,h}) + w_{a,h} \qquad \forall a \in \bar{A},\; h \in H$$

$$s^2_{a,h,2} \leq s_{a,h,1} w_{a,h} \;\Leftrightarrow\; \left\| \begin{pmatrix} 2s_{a,h,2} \\ s_{a,h,1} - w_{a,h} \end{pmatrix} \right\|_2 \leq s_{a,h,1} + w_{a,h} \qquad \forall a \in \bar{A},\; h \in H$$

$$w^2_{a,h} \leq s_{a,h,2} u^h_a \;\Leftrightarrow\; \left\| \begin{pmatrix} 2w_{a,h,2} \\ s_{a,h,2} - u^h_a \end{pmatrix} \right\|_2 \leq s_{a,h,2} + u^h_a \qquad \forall a \in \bar{A},\; h \in H$$

$$s_{a,h,1}, s_{a,h,2} \geq 0 \qquad \forall a \in \bar{A},\; h \in H.$$

After correcting the nonlinearity and nonconvexity of $Q(u)$ we now must deal with separability by replacing u_a^h with $\upsilon_{a,h}$ in the hyperbolic inequalities associated with critical edges $a \in \bar{A}$. We also add the constraint

$$0 \leq \upsilon_{a,h} \leq u_a^h \qquad \forall a \in \bar{A},\ h \in H$$

Thus, we have the following final formulation for the recourse function (hyperbolic inequalities are listed in lieu of the conic quadratic equivalents due to space):

$$Q(u) = \min_{x,v,y,w,z,s,\upsilon} \gamma \left[\sum_{a \in A} f_a v_a + \zeta \sum_{a \in A} \frac{f_a}{c_a^4} y_a \right]$$

$$\text{s.t. } x \in X \tag{6.12a}$$

$$s_{a,1}, s_{a,2} \geq 0 \qquad \forall a \in A \setminus \bar{A} \tag{6.12b}$$

$$\sum_{(r,s) \in \mathcal{OD}} x_a^{rs} = v_a \qquad \forall a \in A \tag{6.12c}$$

$$s_{a,1}^2 \leq y_a v_a \qquad \forall a \in A \setminus \bar{A} \tag{6.12d}$$

$$s_{a,2}^2 \leq s_{a,1} v_a \qquad \forall a \in A \setminus \bar{A} \tag{6.12e}$$

$$v_a^2 \leq 1 \cdot s_{a,2} \qquad \forall a \in A \setminus \bar{A} \tag{6.12f}$$

$$v_a = \sum_{h \in H} w_{a,h} \qquad \forall a \in \bar{A} \tag{6.12g}$$

$$y_a = \sum_{h \in H} z_{a,h} \qquad \forall a \in \bar{A} \tag{6.12h}$$

$$s_{a,h,1}, s_{a,h,2} \geq 0 \qquad \forall a \in \bar{A},\ h \in H. \tag{6.12i}$$

$$w_{a,h}, z_{a,h} \geq 0 \qquad \forall a \in \bar{A},\ h \in H \tag{6.12j}$$

$$s_{a,h,1}^2 \leq (\beta_{a,h} z_{a,h}) w_{a,h} \qquad \forall a \in \bar{A},\ h \in H \tag{6.12k}$$

$$s_{a,h,2}^2 \leq s_{a,h,1} w_{a,h} \qquad \forall a \in \bar{A},\ h \in H \tag{6.12l}$$

$$w_{a,h}^2 \leq s_{a,h,2} \upsilon_{a,h} \qquad \forall a \in \bar{A},\ h \in H \tag{6.12m}$$

$$0 \leq \upsilon_{a,h} \leq u_a^h \qquad \forall a \in \bar{A},\ h \in H \tag{6.12n}$$

The second stage problem (6.12) is now linear in the first stage decision variable u, convex in v, and maintains separability between first and second stage variables. Our model also maintains complete recourse (one of the necessary assumptions of Chapter 2).

6.4 Decomposition

Combining the first and second stages we can formulate one convex mixed integer nonlinear program (MINLP).

$$\min_{u,\eta,x,v,y,w,z,s,\upsilon} \quad (1+\lambda)b^T u + \sum_{k\in K} p_k \left(\gamma \left[\sum_{a\in A} f_a v_a + \zeta \sum_{a\in A} \frac{f_a}{c_a^4} y_a \right] \right) + \lambda \left(\eta + \frac{1}{1-\alpha} \sum_{k\in K} p_k \xi_k \right)$$

$$\text{s.t. } u \in U$$

$$\xi_k \geq 0 \ \forall k \in K$$

$$\xi_k \geq \gamma \left[\sum_{a\in A} f_a v_a + \zeta \sum_{a\in A} \frac{f_a}{c_a^4} y_a \right] - \eta \ \forall k \in K$$

$$\text{(6.12b) - (6.12n) } \forall k \in K$$

6.4.1 Benders Decomposition with Property (P)

Due to our reformulation of the second stage problem, we can decompose the above MINLP into a master problem dealing with only first stage variables (u, η, ξ) and $|K|$ subproblems (one for each scenario $k \in K$, corresponding to $Q^k(u)$). Recall that we must again replace instances of the the recourse objective with an auxiliary variable ϕ_k that can be lower approximated by cutting planes.

For each scenario $k \in K$ we have the subproblem $Q^k(u)$ (6.12). The master problem takes the following form:

$$\min_{u,\eta,\xi} \ (1+\lambda)b^T u + \sum_{k\in K} p_k \phi_k + \lambda \left(\eta + \frac{1}{1-\alpha} \sum_{k\in K} p_k \xi_k \right) \tag{6.13a}$$

$$\text{s.t. } u \in U \tag{6.13b}$$

$$\xi_k \geq 0 \qquad \forall k \in K \tag{6.13c}$$

Both the master and subproblems can be restated in a way that highlights complicating constraints (those that contain both first and second stage variables).

$$\min_{u,\eta,\xi} \ (1+\lambda)b^T u + \sum_{k\in K} p_k\phi_k + \lambda\left(\eta + \frac{1}{1-\alpha}\sum_{k\in K} p_k\xi_k\right) \tag{6.14a}$$

$$\text{s.t. } (u,\eta,\xi) \in \mathbb{U} \tag{6.14b}$$

$$Q^k(u) \ = \min_{x,v,y,w,z,s,\upsilon} \ \gamma\left[\sum_{a\in A} f_a v_a^k + \zeta \sum_{a\in A} \frac{f_a}{c_a^4} y_a^k\right]$$

$$\text{s.t. } (x^k, v^k, y^k, w^k, z^k, s^k, \upsilon^k) \in \mathbb{X} \tag{6.15a}$$

$$\upsilon_{a,h}^k \leq u_a^h \qquad \forall a \in \bar{A},\ h \in H \tag{6.15b}$$

where $\mathbb{U}$ and $\mathbb{X}$ are the sets of constraints that contain only first and second stage variables respectively. The only remaining constraints that include first and second stage variables are those bounding the auxiliary variable $\upsilon_{a,h}$ that was added to achieve separability.

$$\upsilon_{a,h} \leq u_a^h \qquad \forall a \in \bar{A},\ h \in H$$

Using Benders Property (P) on the mixed constraints, given a feasible first stage decision $\bar{u}$ and vector μ_k of optimal dual multipliers for the constraints (6.15b) for all

$k \in K$ we can generate the following optimality cuts.

$$\phi_k \geq Q^k(\bar{u}) - \mu_k{}^T(u - \bar{u}) = Q^k(\bar{u}) - \sum_{a \in \bar{A}} \sum_{h \in H} \mu_{k,a,h}(u_a^h - \bar{u}_a^h) \tag{6.16}$$

$$\xi_k \geq Q^k(\bar{u}) - \mu_k{}^T(u - \bar{u}) - \eta = Q^k(\bar{u}) - \sum_{a \in \bar{A}} \sum_{h \in H} \mu_{k,a,h}(u_a^h - \bar{u}_a^h) - \eta \tag{6.17}$$

We now give the solution algorithm for the transportation network model.

6.4.2 Transportation Network Model Solution Algorithm

Step 1: Set iteration counter $I = 0$ and begin building a branch-and-cut tree with which to solve the following master problem (6.14).

$$\min_{u,\eta,\xi} \ (1+\lambda)b^T u + \sum_{k \in K} p_k \phi_k + \lambda \left(\eta + \frac{1}{1-\alpha} \sum_{k \in K} p_k \xi_k \right) \tag{6.18a}$$

$$\text{s.t. } u \in U \tag{6.18b}$$

$$\xi_k \geq 0 \tag{6.18c}$$

When an integer node is found, let $I = I + 1$ and denote the current optimal solution to the master problem as $(\bar{u}^I, \bar{\eta}, \bar{\phi}^I, \bar{\xi}^I)$. Solve the second stage problem $Q^k(\bar{u}^I)$ for all $k \in K$ and let μ_k^I denote the optimal dual multiplier associated with the one mixed constraint $v_k \leq u$.

Step 2: For all $k \in K$

- If $\bar{\phi}_k^I < Q^k(\bar{u}^I)$, add the following optimality cut to the master problem (all existing nodes in the branch-and-cut tree)

Cut 1: $\phi_k \geq Q^(\bar{u}^I) + \mu_k^{I\,T}\left(\bar{u}^I - u\right).$

- If $\bar{\xi}_k^I < Q^k(\bar{u}^I) - \bar{\eta}$, add the following optimality cut to the master problem (all existing nodes in the branch-and-cut tree)

$$\text{Cut 2:} \qquad \xi_k \geq Q^k(\bar{u}^I) + {\mu_k^I}^T \left(\bar{u}^I - u\right) - \eta.$$

Step 3: Prune the current integer node and continue solving the master problem with added cut constraints. When another integer node is reached, let $I = I + 1$ and denote the current optimal solution to the master problem as $(\bar{u}^I, e\bar{t}a, \bar{\phi}^I, \bar{\xi}^I)$. Solve the second stage problem $Q^k(\bar{u}^I)$ for all $k \in K$ and let μ_k^I denote the optimal dual multiplier associated with the one mixed constraint $\upsilon_k \leq u$. Return to Step 2.

Chapter 7

Conclusion

7.1 Discussion on Model Size and Complexity

In this study we were able to develop a general form, mean-risk SP for making m network protection decisions (with n_i possible strategies for decision $i = 1, \ldots, m$). Recall that our first stage decision x is a column vector of dimension $n = \sum_{i=1}^{m} n_i$. Let N denote the sum over all $i = 1, \ldots, m$, $j = 1, \ldots, n_i$ of the degree d_i of the univariate polynomial $g_{i,j}(y_i)$.

$$N = \sum_{i=1}^{m} \sum_{j=1}^{n_i} d_i = \sum_{i=1}^{m} d_i \sum_{j=1}^{n_i} 1 = \sum_{i=1}^{m} n_i d_i$$

We now discuss the size and complexity of our reformulated recourse function. In order to find the convex hull of a union of convex sets to get rid of disjunctions in the second stage problem we added $2n + m$ variables and $3m$ constraints (m of which were nonlinear). Before creating second order conic constraints, for $i = 1, \ldots, m$, $j = 1, \ldots, n_1$ we split the d_i degree polynomial inequality into d_i inequalities in $\mathbb{R}^3$

which created N variables and n constraints. When formulating the conic quadratic constraints, we create N variables $t_{i,j,k}$. If $a_{i,j}^k$ is nonzero for all $i = 1, \ldots, m$ $j = 1, \ldots, n_i$ $k = 1, \ldots, d_i$ (meaning each univariate polynomial $g_{i,j}(y_i)$ of degree d_i does include all d_i terms) the binary tree method [Alizedeh and Goldfarb, 2003] creates

$$\mathcal{L}_i = \sum_{k=1}^{d_i} [L_k - 1] = \sum_{k=1}^{d_i} \left[\min\{L \in \mathbb{Z} : 2^L \geq k\} - 1\right] = \sum_{k=1}^{d_i} [\lceil \log_2(k) \rceil - 1]$$

variables and constraints for all $i = 1, \ldots, m$ and $j = 1, \ldots, n_i$. Then the number of variables and constraints created for each $I = 1, \ldots, m$ is

$$\begin{aligned}
\mathbb{L}(I) = \sum_{j=1}^{n_I} \mathcal{L} &= \sum_{j=1}^{n_I} \sum_{k=1}^{d_I} [\lceil \log_2(k) \rceil - 1] \\
&= \sum_{k=1}^{d_I} [\lceil \log_2(k) \rceil - 1] \left(\sum_{j=1}^{n_I} 1 \right) \\
&= n_I \sum_{k=1}^{d_I} [\lceil \log_2(k) \rceil - 1] \\
&= n_I \mathcal{L}_I
\end{aligned}$$

In total, the binary trees create

$$\mathbb{M} = \sum_{I=1}^{m} \mathbb{L}(I) = \sum_{i=1}^{m} n_i \mathcal{L}_i = n_i \mathbb{L} \sum_{k=1}^{d_i} [\lceil \log_2(k) \rceil - 1]$$

variables and constraints. This data is included in Table (7.1).

Though this can seem like a worrisome amount of variables and constraints, it is important to note that for any first stage feasible x, $x^i \in \{0,1\}^{n_i} \cap$ SOS-1 for $i = 1, \ldots, m$. Thus for $i = 1, \ldots, m$, $x_j^i = 0$ for $|n_i| - 1$ j's which bounds $\chi_{i,j} = 0$ for

Technique	# of Variables Added	# of Constraints Added
Perspective Reformulation	$2n + m$	$3m$
SOCP Reformulation	$N + \mathbb{M}$	$n + \mathbb{M}$
Separability	n	n
Total:	$N + 3n + m + \mathbb{M}$	$2n + 3m + \mathbb{M}$

Table 7.1: Number of Variables and Constraints Added to each $Q^s(x)$ throughout the Reformulation Process.

the same $|n_i| - 1$ j's.

$$(v_{i,j})^2 \leq \tau_{i,j,k,L_k-1}\chi_{i,j} \tag{7.1}$$

Since the variable $\chi_{i,j}$ is always on the right hand side of the hyperbolic inequality created at level $L_k - 1$ of the binary tree diagram associated with i, j (7.1), whenever $\chi_{i,j} = 0$, $v_{i,j}$ must also equal zero.

$$\begin{aligned}
\chi_{i,j} = 0 &\implies v_{i,j} = 0 \\
(\text{At level } L_k - 2) &\implies \begin{cases} (\tau_{i,j,k,L_k-1})^2 \leq \tau_{i,j,k,L_k-2} v_{i,j} & \text{if } L_k - 1 \in J_v \\ (\tau_{i,j,k,L_k-1})^2 \leq \tau_{i,j,k,L_k-2} \chi_{i,j} & \text{if } L_k - 1 \in J_x \end{cases} \\
&\implies \begin{cases} (\tau_{i,j,k,L_k-1})^2 \leq \tau_{i,j,k,L_k-2} 0 & \text{if } L_k - 1 \in J_v \\ (\tau_{i,j,k,L_k-1})^2 \leq \tau_{i,j,k,L_k-2} 0 & \text{if } L_k - 1 \in J_x \end{cases} \\
&\implies \tau_{i,j,k,L_k-1} = 0
\end{aligned}$$

Thus if $x^i_j = 0$, then $\chi_{i,j} = v_{i,j} = 0$ and $\tau_{i,j,k,\ell} = 0$ for all $k = 1, \ldots, d_i$, $\ell =$

$1, \ldots, L_k - 1$. Therefore,

$$\left|\{x_j^i \neq 0 : j = 1, \ldots, n_i\}\right| = 1 \implies \begin{cases} |\{\chi_{i,j} \neq 0 : j = 1, \ldots, n_i\}| = 1 \\ |\{v_{i,j} \neq 0 : j = 1, \ldots, n_i\}| = 1 \end{cases}$$

$$\implies \left| \left\{ \tau_{i,j,k,\ell} \neq 0 : \begin{array}{r} j = 1, \ldots, n_i \\ k = 1, \ldots, d_i \\ \ell = 1, \ldots, L_k - 1 \end{array} \right\} \right| = \sum_{i=1}^{m} \sum_{j=1}^{n_i} \chi_{i,j} \mathcal{L}_i = \sum_{i=1}^{m} \mathcal{L}_i$$

Therefore each binary tree diagram creates at most $\mathcal{M} = \sum_{i=1}^{m} \mathcal{L}_i$ nonzero variables. This does not change the number of variables and constraints that were added to the second stage; however, it does limit the number of active variables over which $Q(x)$ is minimized given first stage feasible $x \in X$. Therefore it is important to consider the number of variables created during reformulation that may be nonzero. This data is included in Table (7.2). Therefore we have created at most $3m + \sum_{i=1}^{m} \sum_{k=1}^{d_i} \lceil \log_2(k) \rceil$

Technique	**Variables Added**	**# Nonzero Variables**	**Constraints Added**
Perspective Reform.	$n + m$	$3m$	$3m$
SOCP Reform.	$N + \mathbb{M}$	$\sum_{i=1}^{m} d_i + \mathcal{M}$	$n + \mathbb{M}$
Separability	n	m	n
Total:	$N + 2n + m + \mathbb{M}$	$3m + \sum_{i=1}^{m} \sum_{k=1}^{d_i} L_k$	$2n + 3m + \mathbb{M}$

Table 7.2: Number of Variables and Constraints Added to each $Q^s(x)$ throughout the Reformulation Process.

nonzero variables in each subproblem. It is important to note that for many networks within the realm of this research have arc cost functions with relatively low degree. For instance the second stage objective of the transportation network model discussed in Chapter (6) could be partitioned into the costs associated with each edge and that arc cost function was a univariate polynomial of degree 5.

Aktürk et al. [2009] used very similar reformulation techniques for their machine-job assignment problem. For their experiments with both quadratic and cubic objectives, the reformulation derived based on our discussed practices gave promising results, solving the model to optimality more frequently for increasingly large problems.

7.2 Contributions of this Research

One major benefit of using perspective cuts and conic quadratic constraints reformulation is the abundance of state-of-the-art solvers that can directly be applied to problems like our reformulation second stage [Góez and Anjos, 2017, Frangioni and Gentile, 2009]. Many experimental trials have returned promising results for problems such as unit commitment [Yuan et al., 2013] and network design [Günlük and Linderoth, 2008] as well as general mixed integer nonlinear programs when compared to using cutting planes [Frangioni and Gentile, 2009].

Our mean-risk model SP can be applied to resource allocation decisions for any network that satisfies the assumptions from Chapter 1 regarding feasibility and the structure of the second stage. As the uncertainty surrounding vulnerable infrastructure systems builds, there is an increasing need for mean-risk models that can balance efficiency and robustness based on the needs of the decision maker and system. Within networked systems, the underlying effects of congestion result in arc cost per unit flow to take nonlinear forms. In order to solve network protection problems to optimality and build the resilience of these systems we must develop a way to deal with the nonlinearity and other computational obstacles embedded within these models. Though the reformulation techniques discussed in here are becoming increasingly popular, there are still gaps in published research pertaining to perspective and second order cone reformulation, specifically in stochastic MINLPs such as this model.

Appendices

Appendix A Proof of Unique Partition (J_x, J_v)

Proposition 3.7. *Given $k > 0$ and $L = \max\{\ell \in \mathbb{Z}_+ : 2^\ell \geq k\}$ the above process forms a unique partition (J_x, J_v) of the set $\{0, 1, \ldots, L-1\}$ such that*

$$\sum_{\ell \in J_x} 2^\ell = k - 1 \qquad \textit{and} \qquad \sum_{\ell \in J_v} 2^\ell = 2^L - k. \tag{3.11}$$

Thus for all $k > 0$ there is a unique binary tree that can be used to derive the CQR of (3.9) and at levels $\ell = 0, \ldots, L-1$ there is exactly one leaf node corresponding to either x^{2^ℓ} or v^{2^ℓ}.

Proof. It can easily be shown that $2^L - k + (k-1) = 2^L - 1 = \sum_{\ell=0}^{L-1} 2^\ell$. In order to show that (J_x, J_v) is a partition we must show that $J_x \cup J_x = \{0, 1, \ldots, L-1\}$ and $J_x \cap J_v = \emptyset$. From the construction in step 3.2.2.1 we have that $J_x \cup J_x = \{0, 1, \ldots, L-1\}$. Suppose by contradiction that there exists $\ell^* \in J_x \cap J_v$. Since the union of J_x and J_v equal the entire set,

$$\begin{aligned}
\sum_{\ell \in J_x} 2^\ell + \sum_{\ell \in J_v} 2^\ell &= 2(2^{\ell^*}) + \sum_{\ell \in J_x \setminus \{\ell^*\}} 2^\ell + \sum_{\ell \in J_v \setminus \{\ell^*\}} 2^\ell \\
&= 2\left(2^{\ell^*}\right) + \sum_{\ell \in J_x \cup J_v \setminus \{\ell^*\}} 2^\ell \\
&= 2\left(2^{\ell^*}\right) + 2^0 + 2^1 + 2^2 + \cdots + 2^{\ell^*-1} + 2^{\ell^*+1} + \cdots + 2^{L-1} \\
&= 2^{\ell^*+1} + 2^0 + 2^1 + 2^2 + \cdots + 2^{\ell^*-1} + 2^{\ell^*+1} + \cdots + 2^{L-1} \\
&= 2^0 + 2^1 + 2^2 + \cdots + 2^{\ell^*-1} + 2\left(2^{\ell^*+1}\right) + \cdots + 2^{L-1} \\
&= 2^L - 1 + 2^{\ell^*} \\
&\geq 2^L - 1.
\end{aligned}$$

This returns a contradiction, therefore $J_x \cap J_v = \emptyset$. Since $\sum_{i=0}^{\ell-1} = 2^\ell - 1$, there is only one way to partition the set $\{0, 1, \ldots, L-1\}$ such that

$$\sum_{\ell \in J_x} 2^\ell = k - 1 \qquad \text{and} \qquad \sum_{\ell \in J_v} 2^\ell = 2^L - k. \tag{2}$$

This is due to the fact that no power of 2 can be rewritten as the sum of smaller powers using each power at most once.

Therefore the above process generates a unique partition (J_x, J_v) of the set $\{0, 1, \ldots, L-1\}$ and defines the leaf nodes of a unique binary tree that can be used to derive the CQR of and at levels $\ell = 0, \ldots, L-1$ there is exactly one leaf node corresponding to either x^{2^ℓ} or v^{2^ℓ}. $\square$

Appendix B An Explicit Description of A_k

Recall the CQR for the nonlinear constraint

$$\sum_{k=1}^{d} \frac{a_k^{i,j}(v_{i,j})^k}{(x_j^i)^{k-1}} \leq u_{i,j} \qquad \forall i = 1,\dots,m \; j = 1,\dots,n_i \tag{3}$$

found using the binary tree method. For $k = 1,\dots,d_i$ let $L_k = \min\{\ell \in \mathbb{Z}_+ : 2^\ell \geq k\}$. Then for all $i = 1,\dots,m$, $j = 1,\dots,n_i$:

$$v_{i,j} \leq t_{i,j,1} \tag{4}$$

$$A_k \begin{pmatrix} x_j^i \\ v_{i,j} \\ t_{i,j,k} \\ \tau_{i,j,k} \end{pmatrix} \geq_{\mathbf{K}_k} 0 \qquad \forall k = 2,\dots,d_i \tag{5}$$

$$t_{i,j,k} \geq 0 \qquad \forall k = 1,\dots,d_i \tag{6}$$

$$\tau_{i,j,k} \in \mathbb{R}^{L_k - 1} \qquad \forall k = 2,\dots,d_i \tag{7}$$

$$\tau_{i,j,k,\ell} \geq 0 \qquad \forall k = 2,\dots,d_i;\; \ell = 1,\dots L_k - 1 \tag{8}$$

$$\sum_{k=1}^{d} a_k^{i,j} t_{i,j,k} \leq u_{i,j} \tag{9}$$

$$x_j^i, v_{i,j}, u_{i,j} \geq 0 \tag{10}$$

The cone $\mathbf{K}_k$ in constraint is the direct product of L_k Lorentz cones $\mathbf{L}^3$. The matrix A_k is an extraordinarily sparse $3L_k \times L_k + 2$ matrix for all $k = 2,\dots,d_i$. We denote the $L_k + 2$ columns of A_k based on the variable $t_{i,j,k}$, $\tau_{i,j,k,1},\dots \tau_{i,j,k,L_k-1}$, x_j^i or $v_{i,j}$ that elements in the column are multiplied with. For example the first column of A_k is denoted $A_k^{x_j^i}$ as is contains coefficients of x_j^i.

Remark. The index k is removed removed from the superscript of $A_k^{t_{i,j,k}}$ and $A_k^{\tau_{i,j,k,\ell}}$ as they are redundant. Instead these columns are denoted as $A_k^{t_{i,j}}$ and $A_k^{\tau_{i,j,\ell}}$ for $\ell = 1,\dots,L_k - 1$.

The $3L_k$ elements in column $A_k^{x_j^i}$ are denoted $\left(A_k^{x_j^i}\right)_h$ for $h = 1, \ldots, 3L_k$ and similarly for all other columns.

Let (J_x^k, J_v^k) be the partition of $\{0, 1, 2, \ldots L_k - 1\}$ found during the binary tree procedure corresponding to the unique representation of $(x_j^i)^{k-1}$ and $(v_{i,j})^{2^{L_k}-k}$ as products of x_j^i and $v_{i,j}$ raised to powers of 2. For $\ell = 0, \ldots, L_k - 1$, the rows $h = 3\ell + 2$ and $h = 3\ell + 3$ are partitioned similarly for columns $A_k^{x_j^i}$ and $A_k^{v_{ij}}$. The two by two submatrices

$$\begin{bmatrix} \left(A_k^{x_j^i}\right)_{3\ell+2} & \left(A_k^{v_{ij}}\right)_{3\ell+2} \\ \left(A_k^{x_j^i}\right)_{3\ell+3} & \left(A_k^{v_{ij}}\right)_{3\ell+3} \end{bmatrix} \tag{11}$$

for $\ell = 0, \ldots, L_k - 1$ always contain exactly two nonzero elements: 1 and -1. There are two possible forms the submatrices can take:

$$\begin{bmatrix} \left((A_k^{x_j^i}\right)_{3\ell+2} & \left(A_k^{v_{ij}}\right)_{3\ell+2} \\ \left((A_k^{x_j^i}\right)_{3\ell+3} & \left(A_k^{v_{ij}}\right)_{3\ell+3} \end{bmatrix} = \begin{cases} \left[\begin{smallmatrix} -1 & 0 \\ 1 & 0 \end{smallmatrix}\right] & \ell \in J_x^{i,j,k} \\ \left[\begin{smallmatrix} 0 & -1 \\ 0 & 1 \end{smallmatrix}\right] & \ell \in J_v^{i,j,k} \end{cases}. \tag{12}$$

Other than these submatrices, there is only one additional nonzero element in the first two columns of $A_{i,j,k}$.

$$\left(A_k^{x_j^i}\right)_{3\ell+1} = \left(A_k^{v_{ij}}\right)_{3\ell+1} = 0 \qquad\qquad \ell = 0, \ldots, L_k - 2 \tag{13}$$

$$\left(A_k^{x_j^i}\right)_{3(L_k-1)+1} = 0 \tag{14}$$

$$\left(A_k^{v_{ij}}\right)_{3(L_k-1)+1} = 2. \tag{15}$$

Therefore in the first two columns of A_k there are exactly $2L_k + 1$ nonzero elements and $4L_k - 1$ zeros. Though these columns are sparse themselves, they are the densest columns of A_k by far.

The third column of A_k has only 2 nonzero elements.

$$A_k^{t_{i,j}} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix} \tag{16}$$

The only elements in the third column $A_k^{t_ij}$ come from the hyperbolic inequality $\tau_{i,j,k,1}^2 \leq t_{i,j,k} x_j^i$ (or $\tau_{i,j,k,1}^2 \leq t_{i,j,k} v_{i,j}$) that is then translated into the following conic quadratic inequality in $\mathbb{R}^3$:

$$\left\| \begin{pmatrix} 2\tau_{i,j,k} \\ t_{i,j,k} - x_j^i \end{pmatrix} \right\|_2 \leq t_{i,j,k} + x_j^i \qquad \left(\left\| \begin{pmatrix} 2\tau_{i,j,k} \\ t_{i,j,k} - v_{i,j} \end{pmatrix} \right\|_2 \leq t_{i,j,k} + v_{i,j} \right). \tag{17}$$

The remaining columns $A_k^{\tau_{i,j,1}}, \ldots, A_k^{\tau_{i,j,L_k}}$ all contain exactly 3 nonzero elements each.

$$\left(A_k^{\tau_{i,j,\ell}}\right)_h = 0 \qquad h = 1, \ldots, 3(\ell - 1) \tag{18}$$

$$\left(A_k^{\tau_{i,j,\ell}}\right)_{3\ell-2} = 2 \tag{19}$$

$$\left(A_k^{\tau_{i,j,\ell}}\right)_h = 0 \qquad h = 3\ell - 1, 3\ell, 3\ell + 1 \tag{20}$$

$$\left(A_k^{\tau_{i,j,\ell}}\right)_{3\ell+2} = 1 \tag{21}$$

$$\left(A_k^{\tau_{i,j,\ell}}\right)_{3\ell+3} = 1 \tag{22}$$

$$\left(A_k^{\tau_{i,j,\ell}}\right)_h = 0 \qquad h = 3(\ell + 1) + 1, \ldots, 3L_k \tag{23}$$

These terms correspond to the hyperbolic inequalities

$$\tau_{i,j,k,\ell}^2 \leq \tau_{i,j,k,\ell-1} x_j^i \qquad \left(\tau_{i,j,k,\ell}^2 \leq \tau_{i,j,k,\ell-1} v_{i,j}\right) \quad \text{and} \tag{24}$$

$$\tau_{i,j,k,\ell+1}^2 \leq \tau_{i,j,k,\ell} x_j^i \qquad \left(\tau_{i,j,k,\ell+1}^2 \leq \tau_{i,j,k,\ell} v_{i,j}\right). \tag{25}$$

Using the above information we can create a visualization for A_k for one possible instance k. It should be noted that since the partition $\left(J_x^k, J_v^k\right)$ is unique, the matrix A_k is also unique. For any $2^{L_k-1}+1 \leq \kappa \leq 2^{L_k}$ the matrix A_κ only differs from A_k in the first two columns in the first $3(L_k-1)$ rows.

$$
\begin{array}{c}
\begin{array}{ccccccccc}
x_j^i & v_{i,j} & t_{i,j,k} & \tau_{i,j,k,1} & \tau_{i,j,k,2} & \cdots & \cdots & \tau_{i,j,k,L_k-2} & \tau_{i,j,k,L_k-1}
\end{array}
\\
\begin{array}{c}
\\ \mathbf{L}^3 \\ \\ \\ \mathbf{L}^3 \\ \\ \\ \\ \mathbf{L}^3 \\ \\ \\ \mathbf{L}^3 \\ \\
\end{array}
\left[
\begin{array}{ccccccccc}
0 & 0 & 0 & 2 & 0 & \cdots & \cdots & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\
1 & 0 & 1 & 0 & 0 & \cdots & \cdots & 0 & 0 \\
0 & 0 & 0 & 0 & 2 & \cdots & \cdots & 0 & 0 \\
0 & -1 & 0 & 1 & 0 & \cdots & \cdots & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & \cdots & \cdots & 0 & 0 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \cdots & \vdots & \vdots \\
0 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 2 \\
0 & -1 & 0 & 0 & 0 & \cdots & \cdots & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & \cdots & \cdots & 1 & 0 \\
0 & 2 & 0 & 0 & 0 & \cdots & \cdots & 0 & 0 \\
-1 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1 \\
1 & 0 & 0 & 0 & 0 & \cdots & \cdots & 0 & 1
\end{array}
\right]
\end{array}
\tag{26}
$$

Appendix C Notation associated with Transportation Network Model

Index and Set

$i/j \in N$	Nodes in network
$a/(i,j) \in A$	Links in network
$\bar{A}$	Set of all critical links in the network, a subset of A
$r \in O$	Subset of nodes from which there is a flow demanded
$s \in D$	Subset of nodes with which there is a demand of flow
$rs/(r,s) \in \mathcal{OD}$	Origin-Destination node pairs for which there is flow demand, a subset of $O \times D$
$h \in H$	All possible retrofitting strategies
$k \in K$	Discrete set of possible future scenarios

Input Parameters

c_a	Units of capacity of link a
f_a	Free flow time of link a
d^{rs}	Units of flow demanded for origin-destination pair rs
b_a^h	Investment cost associated with choosing strategy h for critical link $a \in \bar{A}$
p_k	Probability of scenario k
$\theta_a^{h,k}$	Expected remaining capacity ratio for link a given strategy h and scenario k

Decision Variables

u_a^h	Binary variable equal to 1 if strategy h is selected for critical link a, 0 otherwise
x_a^{rsk}	Quantity of flow across link a to meet demand for O-D pair rs in scenario k
v_a^k	Total flow over link a due to all flow demand in scenario k

Table 3: Notation of initial variables and given information for the transportation network problem.

Model Parameters

b_0	Total budget for retrofitting investment costs
β	Empirical data value (often assumed 0.15) used in BPR calculation of cost per unit flow
α	Confidence level used in risk measure
λ	Weight factor representing the trade-off between the risk measure and expected value
γ	Parameter converting travel time to a monetary value

Table 4: Parameters whose values are set by decision makers to best fit the model to their needs.

Bibliography

M. Aktürk, A. Atamtürk, and S. Gürel. A strong conic quadratic reformulation for machine-job assignment with controllable processing times. *Operations Research Letters*, 37(3):187–191, 2009.

F. Alizedeh and D. Goldfarb. Second-order cone programming. *Mathematical Programming*, 95(1):3–51, 2003.

Philippe Artzner, Freddy Delbaen, Jean-Marc Eber, and David Heath. Coherent measures of risk. *Mathematical Finance*, 9(3):203–228, 1999. doi: 10.1111/1467-9965.00068.

G Barbarosoğlu and Y Arda. A two-stage stochastic programming framework for transportation planning in disaster response. *Journal of the Operational Research Society*, 55(1):43–53, 2004.

A. Ben-Tal and A. Nemirovski. *Lectures on Modern Convex Optimization: Analysis, Algorithms, and Engineering Applications*. SIAM, 2001.

Aharon Ben-Tal, Laurent El Ghaoui, and Arkadi Nemirovski. *Robust optimization*, volume 28. Princeton University Press, 2009.

J. F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Computational Management Science*, 2(1):3–19, Jan 2005. ISSN 1619-6988. doi: 10.1007/s10287-004-0020-y. URL `https://doi.org/10.1007/s10287-004-0020-y`.

John R. Birge and François V. Louveaux. A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384 – 392, 1988.

J.R. Birge and F. Louveaux. *Introduction to Stochastic Programming*. Springer Series in Operations Research and Financial Engineering. Springer New York, 2011. ISBN 9781461402374. URL `https://books.google.com/books?id=Vp0Bp8kjPxUC`.

Miguel Carrion. Risk-aversion modeling: Medium-term electricity retailer decision-making. In *2008 IEEE Power and Energy Society General Meeting-Conversion and Delivery of Electrical Energy in the 21st Century*, pages 1–3. IEEE, 2008.

Miguel Carrión, Uwe Gotzes, and Rüdiger Schultz. Risk aversion for an electricity retailer with second-order stochastic dominance constraints. *Computational Management Science*, 6(2):233–250, 2009.

S. Ceria and J. Soares. Convex programming for disjunctive convex optimization. *Mathematical Programming*, 86:595–614, 1999.

T. Cotton and L. Ntaimo. Computational study of decomposition algorithms for mean-risk stochastic linear programs. *Mathematical Programming Computation*, 7: 471–499, 2015.

Presidential Policy Directive. Critical infrastructure security and resilience. ppd-21, released february 12, 2013, 2013.

M. Fischetti, I. Ljubić, and M. Sinnl. Benders decomposition without separability: A computational study for capacitated facility location problems. *European Journal of Operational Research*, 253(3):557 – 569, 2016.

Roger Fletcher and Sven Leyffer. Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66(1):327–349, Aug 1994. ISSN 1436-4646. doi: 10.1007/BF01581153.

C. Floudas. *Nonlinear and Mixed-Integer Optimization: Fundamentals and Applications*. Oxford University Press, 1995.

A. Frangioni and C. Gentile. A computational comparison of reformulations of the perspective relaxation: Socp vs. cutting planes. *Operations Research Letters*, 37(3):206 – 210, 2009. ISSN 0167-6377. doi: https://doi.org/10.1016/j.orl.2009.02.003. URL `http://www.sciencedirect.com/science/article/pii/S0167637709000339`.

Arthur Geoffrion. Generalized benders decomposition. *Journal of Optimization Theory and Applications*, 10:237–260, 10 1972. doi: 10.1007/BF00934810.

Julio C Góez and Miguel F Anjos. Second-order cone optimization formulations for service system design problems with congestion. In *Modeling and Optimization: Theory and Applications*, pages 97–120. Springer, 2017.

O. Günlük and J. Linderoth. Perspective relaxation of mixed integer nonlinear programs with indicator variables. *Integer Programming and Combinatorial Optimization*, 5035:1–16, 2008.

O. Günlük and J. Linderoth. Perspective reformulation and applications. In *The IMA Volumes in Mathematics and its Applications*. Springer, 2012.

Omprakash K. Gupta and A. Ravindran. Branch and bound experiments in convex nonlinear integer programming. *Management Science*, 31(12):1533–1546, 1985. doi: 10.1287/mnsc.31.12.1533.

Peter Kall and Stein Wallace. *Stochastic Programming*, volume 46. 01 1994. doi: 10.2307/2584504.

Pavlo Krokhmal, Michael Zabarankin, and Stan Uryasev. Modeling and optimization of risk. In *HANDBOOK OF THE FUNDAMENTALS OF FINANCIAL DECISION MAKING: Part II*, pages 555–600. World Scientific, 2013.

Changzheng Liu, Yueyue Fan, and Fernando Ordóñez. A two-stage stochastic programming model for transportation network protection. *Computers & Operations Research*, 36(5):1582 – 1590, 2009. ISSN 0305-0548. Selected papers presented at the Tenth International Symposium on Locational Decisions (ISOLDE X).

M. Lobo, L. Vandenberghe, and S. Boyd. Applications of second-order cone programming. *Linear Algebra and its Applications*, 284(1-3):193–228, 1998.

Jie Lu, Sez Atamturktur, and Yongxi Huang. Bi-level resource allocation framework for retrofitting bridges in a transportation network. *Transportation Research Record: Journal of the Transportation Research Board*, 2550:31–37, 01 2016. doi: 10.3141/2550-05.

Jie Lu, Akshay Gupte, and Yongxi Huang. A mean-risk mixed integer nonlinear program for transportation network protection. *European Journal of Operational Research*, 265, 07 2017. doi: 10.1016/j.ejor.2017.07.025.

Andreas Märkert and Rüdiger Schultz. On deviation measures in stochastic integer programming. *Operations Research Letters*, 33(5):441–449, 2005.

ApS MOSEK. Mosek modeling cookbook, 2018.

G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience series in discrete mathematics and optimization. Wiley, 1988. ISBN 9780471828198. URL `https://books.google.com/books?id=uG4PAQAAMAAJ`.

Y. Nesterov and A. Nemirovskii. *Interior-point Polynomial Algorithms in Convex Programming*. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics (SIAM, 3600 Market Street, Floor 6, Philadelphia, PA 19104), 1994. ISBN 9781611970791. URL `https://books.google.com/books?id=MgWY_clJXusC`.

N. Noyan, M. Meraklı, and S. Küçükyavuz. Two-stage stochastic programming under multivariate risk constraints with an application to humanitarian relief network design. *Mathematical Programming*, 01 2017. doi: 10.1007/s10107-019-01373-4.

Nilay Noyan. Risk-averse two-stage stochastic programming with an application to disaster management. *Computers & Operations Research*, 39(3):541–559, 2012.

American Society of Civil Engineers. 2017 infrastructure report card. 2017.

Bureau of Public Roads. *Traffic assignment manual*. Urban Planning Division, US Department of Commerce, 1964.

WLodzimierz Ogryczak and Andrzej Ruszczynski. Dual stochastic dominance and related mean-risk models. *SIAM Journal on Optimization*, 13(1):60–78, 2002.

Georg Pflug. Someremarks on the value-at-risk and the conditional value-at-risk. 09 2000. doi: 10.1007/978-1-4757-3150-7_15.

R. Rahmaniani, T. Crainic, M. Gendreau, and W. Rei. The benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801 – 817, 2017. ISSN 0377-2217. doi: https://doi.org/10.1016/j.ejor.2016.12.005. URL http://www.sciencedirect.com/science/article/pii/S0377221716310244.

R. Tyrrell Rockafellar and Stanislav Uryasev. Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41, 2000.

Rüdiger Schultz and Frederike Neise. Algorithms for mean-risk stochastic integer programs in energy. In *2006 IEEE Power Engineering Society General Meeting*, pages 8–pp. IEEE, 2006.

Rüdiger Schultz and Stephan Tiedemann. Risk aversion via excess probabilities in stochastic programs with mixed-integer recourse. *SIAM Journal on Optimization*, 14(1):115–138, 2003.

A. Shapiro, D. Dentcheva, and A. Ruszczyński. *Lectures on Stochastic Programming*. Society for Industrial and Applied Mathematics, 2009. doi: 10.1137/1.9780898718751. URL https://epubs.siam.org/doi/abs/10.1137/1.9780898718751.

Stein W Wallace and William T Ziemba. *Applications of stochastic programming*. SIAM, 2005.

Xiaohui Yuan, Hao Tian, Shuangquan Zhang, Bin Ji, and Yanhong Hou. Second-order cone programming for solving unit commitment strategy of thermal generators. *Energy Conversion and Management*, 76:20 – 25, 2013. ISSN 0196-8904. doi: https://doi.org/10.1016/j.enconman.2013.07.019. URL http://www.sciencedirect.com/science/article/pii/S0196890413003907.