

ROBUST SHORTEST PATHS UNDER UNCERTAINTY USING
CONDITIONAL VALUE-AT-RISK

By
JULIE ANNA BRIGHT
Bachelor of Science in Industrial Engineering and
Management
Oklahoma State University
Stillwater, OK 74075
2009

Submitted to the Faculty of the
Graduate College of
Oklahoma State University
in partial fulfillment of
the requirements for
the Degree of
MASTER OF SCIENCE
December 2011

COPYRIGHT ©

By

JULIE ANNA BRIGHT

December 2011

ROBUST SHORTEST PATHS UNDER UNCERTAINTY USING
CONDITIONAL VALUE-AT-RISK

Thesis Approved:

Balabhaskar Balasundaram

Thesis Advisor

Manjunath Kamath

Ricki Ingalls

Sheryl Tucker

Dean of the Graduate College

TABLE OF CONTENTS

Chapter	Page
1 Introduction	1
1.1 Organization of this Thesis	2
1.2 The Deterministic Shortest Path Problem	2
1.2.1 Formal Definition	2
1.2.2 Classical Results	3
1.2.3 Linear Programming Formulation	4
1.3 Stochastic Shortest Path Problems	5
1.4 Research Problem	6
1.5 Research Tasks	7
2 Review of Modeling Approaches	8
2.1 Stochastic Programming Models	8
2.2 Robust Optimization Models	9
2.3 Other Modeling Approaches	10
2.4 Models Based on Conditional Value-at-Risk	11
2.5 Formal Definition of CVaR	13
2.6 Using CVaR in Optimization	15
3 Model Development and Analysis	17
3.1 Modeling Arc Failures	17
3.2 Formulation of the CVARSPP	17
3.2.1 Additional Parameters	18

3.2.2	CVARSP Optimization Model	18
3.3	Loss Functions	20
3.3.1	Path Reliability	20
3.3.2	Number of Arc Failures	22
3.3.3	Number of Detours	23
3.4	Comparison of Loss Functions	29
3.4.1	Magnitude of Losses	29
3.4.2	Dependence of CVaR on β	30
4	Computational Experiments	31
4.1	Implementation Details	31
4.2	Experiments	32
4.2.1	Group 1: Complete Enumeration	33
4.2.2	Group 2: Branch Networks	34
4.2.3	Group 3: Uniform Random Networks	35
4.3	Comparison of Loss Functions	36
5	Conclusion and Future Work	38
	Bibliography	39
A	Optimization Models	44
A.1	Parameters	44
A.2	Decision Variables	44
A.3	Path Reliability Loss Function	45
A.4	Arc Failure Loss Function	46
A.5	Detours Loss Function	47
B	Database Schema	48
B.1	Database Tables	48

B.2 Database Schema	53
C Objective Function Graphs	55
D Test Network Diagrams	70
D.1 Small Network: exp0002	70
D.2 Branch Networks	71
D.3 Uniform Random Networks	72

LIST OF TABLES

Table		Page
3.1	Summary of Loss Functions (LF)	29
4.1	Summary of Test Bed	32

LIST OF FIGURES

Figure	Page
2.1 An example of VaR and CVaR	12
3.1 Path Reliability Loss Function: The loss for this path is 1.	21
3.2 Arc Failures Loss Function: The loss for this path is 3.	22
3.3 Detours Loss Function: The loss for this path is 2.	24
4.1 Network exp0002	34
4.2 Branch Network Structure	35
B.1 Legend: Crow’s Foot Notation	53
B.2 Database Schema	54
C.1 exp0002: Level sets of the optimal path length	56
C.2 Change in optimal path properties	57
C.3 branch02: Level sets of the optimal path length	58
C.4 Change in optimal path properties	59
C.5 branch03: Level sets of the optimal path length	60
C.6 Change in optimal path properties	61
C.7 branch04: Level sets of the optimal path length	62
C.8 Change in optimal path properties	63
C.9 gnp01: Level sets of the optimal path length	64
C.10 Change in optimal path properties	65
C.11 gnp02: Level sets of the optimal path length	66
C.12 Change in optimal path properties	67

C.13 gnp03: Level sets of the optimal path length	68
C.14 Change in optimal path properties	69
D.1 Network exp0002	70
D.2 Network branch02	71
D.3 Network branch03	71
D.4 Network branch04	72
D.5 Network gnp01 — $GNP(25,0.1)$	72
D.6 Network gnp02 — $GNP(25,0.25)$	73
D.7 Network gnp03 — $GNP(25,0.4)$	73

CHAPTER 1

Introduction to Shortest Path Problems Under Uncertainty

The shortest path problem (SPP) is to find the path through a network with the lowest “cost” — whether that cost is measured in miles, money, minutes, or any other metric. This problem has been extensively studied over the past 50 years (Pollack and Wiebenson, 1960; Dreyfus, 1969; Deo and Pang, 1984). When the structure and costs of the network are known, then very fast and powerful tools exist to find a shortest path, such as Dijkstra’s algorithm (Dijkstra, 1959).

In the real world, though, networks are afflicted with failures, and costs can fluctuate randomly. This limits the applicability of classical, deterministic SPP algorithms. Often the definition of a “shortest” path needs to be adjusted. Different mathematical models and algorithms are needed to handle uncertainty. The family of shortest path problems under uncertainty (modeled by probability distributions) are collectively called stochastic shortest path problems (SSPPs).

This thesis investigates one type of SSPP. In this SSPP, the arcs of a network can fail randomly and independently with some probability. This model is useful for roads that close due to accidents or severe weather or for PERT charts where activities may not get done. The effect of arc failures on the overall network and the shortest path will be investigated in order to create measures to quantify the loss of an arc. Secondly, a quantitative measure of downside risk called Conditional Value-at-Risk (CVaR) will be used to find short and robust paths in the network (Rockafellar and Uryasev, 2002).

1.1 Organization of this Thesis

Chapter 1 discusses the classical or deterministic shortest path problem (SPP), including a formal definition and a review of the most common algorithms, before providing some motivating examples to show how uncertainty in graphs affects the SPP. This chapter also introduces the research problem. Chapter 2 reviews the modeling approaches in the literature to show what kinds of uncertainty have been captured and what kinds of uncertainty remain relatively less-studied. This chapter also introduces the concept of CVaR.

Chapter 3 formulates the CVaR-Constrained Stochastic Shortest Path Problem (CVARSPP) and develops three loss functions to model loss due to probabilistic arc failures to use in the CVARSPP. This chapter also analyzes the loss functions, deriving their distributions for uniform arc failures and some implications for their behavior in CVARSPP models. Then, Chapter 4 presents some exploratory computational experiments that illustrate the behavior of the CVARSPP models with the loss functions, leading to some preliminary guidance on setting parameters. Finally, Chapter 5 presents the conclusions of this thesis and some directions for further research.

1.2 The Deterministic Shortest Path Problem

1.2.1 Formal Definition

Consider the network $G = (N, A)$, consisting of N , a set of nodes, and $A \subseteq N \times N$, a set of directed arcs connecting pairs of nodes. For each arc $(ij) \in A$, node i is called the tail and node j is called the head. Each arc also has an associated cost $c_{ij} \in \mathbb{R}_+$. For each node $i \in N$, its in-neighborhood is defined as $N^-(i) = \{j | (ji) \in A\}$ and its out-neighborhood is defined as $N^+(i) = \{j | (ij) \in A\}$.

A directed path P is a sequence of k distinct nodes (i_1, i_2, \dots, i_k) , where $(i_j, i_{j+1}) \in$

A for all $j \in \{1, \dots, k-1\}$. Throughout the remainder of this thesis, directed paths will be referred to simply as “paths”. The edge-induced subgraph of the path P is denoted by $G(P) = (N(P), A(P))$. The path P can also be represented by an incidence vector $\mathbf{x} \in \{0, 1\}^{|A|}$ of arcs in P , which induces the subgraph $G(\mathbf{x}) = (N(\mathbf{x}), A(\mathbf{x}))$. The cost, also called weight or length, of a path P is $c_P = \sum_{(ij) \in A(P)} c_{ij}$.

Definition 1.2.1. (*Deterministic Shortest Path Problem*)

Given network $G = (N, A)$ and nodes $s, t \in N$, the Point-to-Point Shortest Path Problem is to find a minimum cost path from node s to t in G .

The Single-Source Shortest Path Problem is to find a shortest path from a source node $s \in N$ to each node $i \in N \setminus \{s\}$.

The All-Pairs Shortest Path Problem is to find a shortest path from every node $i \in N$ to every other node $j \in N$.

1.2.2 Classical Results for Shortest Path Problem

In the absence of negative cost cycles in the network, the SPP is a well-solved problem, for which several strongly-polynomial algorithms are known (Deo and Pang, 1984). These algorithms assume that the weight of each arc in the network is constant, and network structure (nodes and arcs) are known with certainty.

Algorithms for the SPP fall into two main categories: label-setting and label-correcting (Deo and Pang, 1984). Dijkstra’s algorithm (Dijkstra, 1959) is an example of a label-setting algorithm. The Bellman-Ford algorithm (Bellman, 1958) is a label-correcting algorithm derived from dynamic programming. The label-setting algorithms require all arc costs to be nonnegative, while the label-correcting algorithms only require that the network have no negative cost cycles. The Floyd-Warshall algorithm for the All-Pairs Shortest Path Problem (Floyd, 1962) is also based on dynamic programming. The Single-Source and Point-to-Point Shortest Path Problems can also be formulated as a linear program and solved with the net-

work simplex method when the network contains no negative cost cycles (Dantzig, 1957).

1.2.3 Linear Programming Formulation of the Shortest Path Problem

The linear programming (LP) formulation of Point-to-Point SPP optimization model is worth presenting on its own since this thesis extends this formulation. The constraint matrix for the LP formulation of the SPP is totally unimodular, so the extreme point optimal solutions are integral (Ahuja et al., 1993).

Inputs

Network $G = (N, A)$ is a directed network.

$s \in N$ is the source node.

$t \in N$ is the sink node.

c_{ij} is the cost of arc $(ij) \in A$.

Decision Variables

x_{ij} for all $(ij) \in A$. At any $\mathbf{x} \in \{0, 1\}^{|A|}$, $x_{ij} = 1$ implies that arc (ij) is in the solution and $x_{ij} = 0$ implies that arc (ij) is not.

Formulation 1.2.2 (Deterministic Formulation).

$$\text{Minimize } \sum_{(ij) \in A} c_{ij} x_{ij} \tag{1.1}$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (1.2)$$

$$x_{ij} \geq 0 \quad \forall (ij) \in A \quad (1.3)$$

1.3 Stochastic Shortest Path Problems

Uncertainty in the system underlying a network model may be modeled by random variation in network parameters such as arc costs and in the structure of the network, particularly node and arc failures. Once uncertainty is allowed into the network model, there is no longer one well-defined shortest path problem. A family of Stochastic Shortest Path Problems (SSPPs) arise instead, depending on which elements in the network are uncertain and how the uncertainty is modeled. Furthermore, SSPPs generally require the redefinition of the concept of feasible and optimal solutions.

One SSPP models the arc cost c_{ij} as a random variable. Applications of this model include modeling the travel time for an arc in a congested road network or the task duration in a PERT network (Ball et al., 1995a). For this problem, two ways to redefine the shortest path are in terms of expected length, which gives rise to many stochastic programming approaches, and worst-case length, in which arc costs are set at or close to their maximum possible values, which lies behind many robust optimization approaches (Ball et al., 1995b).

Uncertainty about which node is the source or sink can be modeled by randomly assigning a supply of +1 to one node or -1 to another. This would be useful in modeling ambulance routing during an emergency when the driver dis-

covers which ER has room for the patient en-route.

Allowing uncertainty in the structure of the network (e.g. uncertainty in the set of nodes or arcs, or only partial knowledge) gives rise to another set of SSPPs. One source of uncertainty is limited knowledge of the network, such as not knowing the cost or head node of an arc until the tail is reached. This is frequently used to model hide-and-seek or adversarial search on networks (Papadimitriou, 1991) or for navigating real road networks without a map, such as after a major disaster (Papadimitriou, 1991). Secondly, the nodes may be probabilistic — that is, they can fail. This is used in modeling telecommunications networks or computer networks (Jaillet, 1992).

Finally, the arcs may fail with some probability, giving rise to the subject of this thesis. Allowing arcs to fail allows modeling a wide variety of disturbances in networks. For instance, road closure in transportation networks, due to accidents, weather, or other events, can be modeled as an arc failure. Cutting through the cables in fiber optic networks, which happens during construction or fires (Ball et al., 1995a), can also be modeled as arc failure. In a social network, an on-again off-again friendship between two people can be modeled as an arc that fails after an argument and is restored after a reconciliation. It is therefore desirable to find a short, reliable path through a network. The redefinition of the shortest path depends on the trade off between reliability and shortness.

1.4 Research Problem

This thesis investigates models of the Stochastic Shortest Path Problem with Probabilistic Arc Failures (SSPP-PAF) using Conditional Value-at-Risk (CVaR) constraints. CVaR is a risk management tool that can trade off risk aversion and performance, so it is able to handle varying levels of risk tolerance. Moreover, CVaR is based on an explicit loss function, which allows more precision in modeling the effects

of arc failures. The concept of CVaR is discussed in more detail in Chapter 2. This thesis will limit itself to arcs that fail randomly and independently of each other with some probability. All other graph parameters are deterministic.

1.5 Research Tasks

The objective of this thesis is to develop CVaR-based optimization models for the SSPP-PAF. The solutions to these models should be paths that are short and robust under arc failures; specifically, the loss due to arc failure as measured by using CVaR should be low. To achieve this objective, the tasks listed below will be pursued.

1. Model the effects of arc failures using different loss functions.
2. Develop and implement optimization models to find the shortest path subject to CVaR constraints.
3. Investigate the effect of different loss functions and CVaR bound on the feasibility of the optimization model.
4. Gain insight into the difference between models using expected value and CVaR via their behavior with different loss functions on carefully-chosen test instances.

CHAPTER 2

A Review of Modeling Approaches for the Shortest Path Problem under Uncertainty

2.1 Stochastic Programming Models

Stochastic programming is one of the approaches available to model optimization problems under uncertainty. Stochastic programming models typically use expected value to model the effect of uncertainty in the objective function and to allow for recourse actions (Sen and Higle, 1999). As a measure of risk, expected value is risk neutral. It weights gains as high as losses whereas human decision-makers are often much more risk-averse (Canada et al., 2004), preferring to minimize losses rather than maximize gains.

There have been a few stochastic programming models of the SSPP-PAF. Croucher (1978) incorporated recourse decisions into a label-correcting algorithm for directed acyclic networks using dynamic programming. The paper attempts to model a traveler who has imperfect knowledge of a road network. In particular, the traveler does not know whether an arc is blocked until arriving at its head. The objective of the algorithm is to find a path with the shortest expected length from each node to the sink node given that the desired arc is unreliable. In the algorithm, each node is labeled with the expected shortest path length to the sink. Each arc in the network has a probability p_{ij} of being usable. At each node an outgoing arc is selected. If the desired arc is broken, one of the other outgoing arcs of the node is selected at random. When the node has only one outgoing arc, it is not allowed to fail. Each node is labeled with the expected shortest path length to the sink, which

is the average of the labels of the out-neighbors weighted by their probability of transversal. This algorithm is quick to compute, but it relies on the assumption that at least one outgoing arc at each node will be usable.

Andreatta and Romeo (1988) built on Croucher (1978)'s paper to explicitly find optimal recourse for a blocked arc, and they find a path that minimizes the expected length of the path including recourse, which they define as detours around a failed arc. The model uses stochastic dynamic programming to solve the problem. This model suffers from the curse of dimensionality as the number of scenarios and the number of failed arcs per scenario grow. Like the previous paper, this one seeks only to minimize the expected path length and does not give special consideration to the extremes of the distribution.

2.2 Robust Optimization Models

Robust optimization is another approach to optimization under uncertainty. Robust optimization models seek to find solutions that are feasible no matter how the underlying uncertainty resolves (Bertsimas et al., 2008). The uncertainty is captured in an uncertainty set, which is a collection of possible future states of the system being modeled. The optimal solution must be feasible for every element in the uncertainty set. The uncertainty set need not contain every possible resolution of the uncertainty, only the subset for which the the optimal solution must remain feasible. Robust optimization tends to favor minimax objectives (Bertsimas et al., 2008).

Recently Yu and Yang (1998) showed a version of the robust shortest path problem to be NP-complete. The paper considers a network with uncertain arc costs, which are captured in a set of scenarios, and develops two robust optimization models for the problem. The first model uses a minimax objective over the scenarios. The second model seeks a path with the smallest possible range of lengths.

Although the authors develop a dynamic programming algorithm for networks with bounded numbers of scenarios, the algorithm is pseudopolynomial in the number of scenarios. As the number of scenarios needed to capture the behavior of a large graph grows very quickly, this algorithm is not suitable for large scale uses (Yu and Yang, 1998).

2.3 Other Modeling Approaches

A conceptually similar approach to the SSSP-PAF is the Minimum Cost Reliability Ratio Path Problem (MCRRPP), a bicriteria optimization model proposed by Ahuja (1988). This problem considers paths from a single source node to a single sink in directed network with arc failures. The objective is to find a path with the optimal ratio of cost to reliability, defined as the probability of no arc failures on the path. The author first develops optimality criteria for a bicriteria optimization problem with the criteria of cost and reliability. The paper develops a pseudopolynomial dynamic programming algorithm with a complexity of $O(mnD \log m)$ to solve the problem, although the computational experiments found that average-case complexity was much lower. The MCRRPP trades off cost and reliability in a risk-neutral manner. By using reliability in the formulation of the MCRRPP, the model implicitly a loss of 100% when any arc on the path fails, and 0% otherwise. This approach is similar to the approach used in this thesis in that it uses arc failures and explicitly trades off cost and reliability.

In a related approach, Jaillet (1992) incorporates node failures into a SSPP model. The goal of this model is to find an easily repairable path of minimum expected length given node failures. The paper shows that the SSSP with probabilistic node failures is in general NP-complete although some restricted types of the problem are solvable in polynomial time.

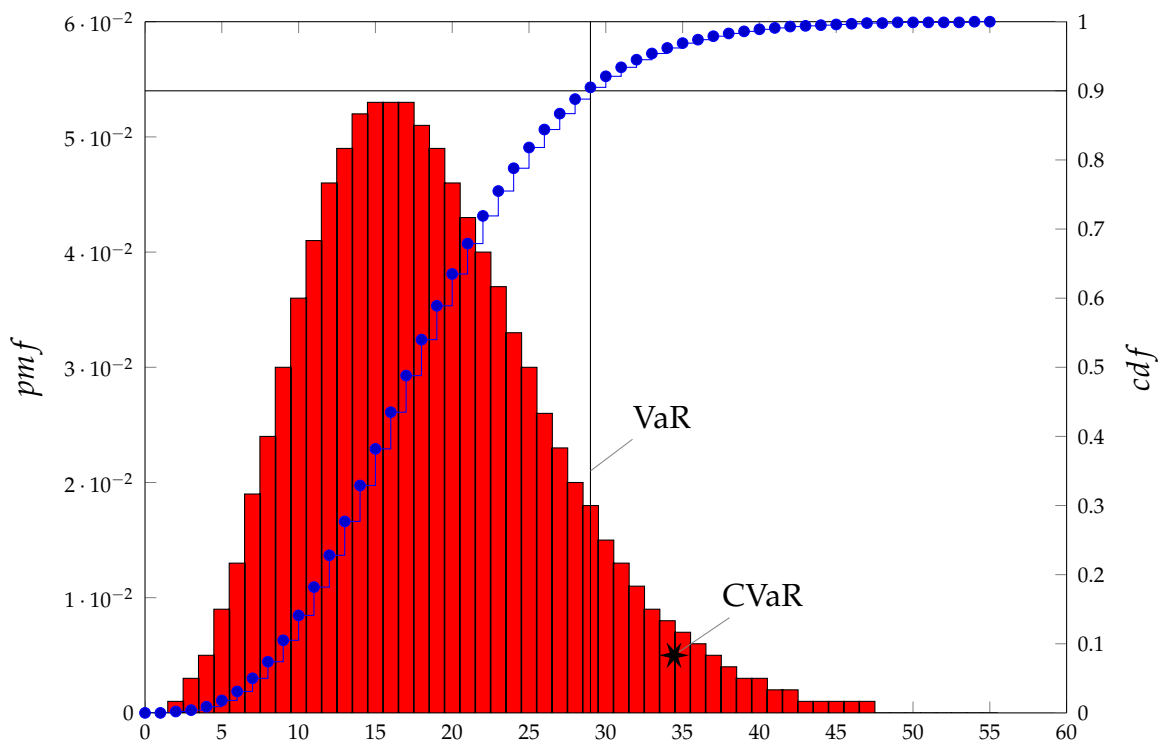
2.4 Models Based on Conditional Value-at-Risk

CVaR was developed in the late 1990s as an improvement on an earlier financial risk measure called Value-at-Risk or VaR (Acerbi and Tasche, 2002). VaR has been very popular, even being written into regulations in several countries (Rockafellar and Uryasev, 2000), but it has a number of drawbacks (Rockafellar and Uryasev, 2000). For most probability distributions, VaR is difficult to compute and is not a coherent risk measure (Artzner et al., 1999). CVaR, on the other hand, is a coherent measure of risk and is easier to calculate (Acerbi and Tasche, 2002), where it is called expected shortfall. More pertinently, Rockafellar and Uryasev presented a way to use CVaR in optimization problems via a portfolio optimization example (Rockafellar and Uryasev, 2002).

Intuitively, CVaR is the average of the upper tail of the distribution of losses due to uncertainty, which are quantified in a loss function. The loss function and the threshold for the upper tail are provided by the modeler. Figure 2.1 illustrates VaR and CVaR for a negative binomial distribution with the tail threshold set at the 90th percentile.

Currently CVaR is applied in several areas of finance such as insurance and credit risk evaluation (Rockafellar and Uryasev, 2000). In these applications, risk is usually specified in terms of monetary losses (Rockafellar and Uryasev, 2000) although utility can be used as well. In more general contexts, though, CVaR is defined in terms of a real-valued loss function $L(\mathbf{x}, \mathbf{Y})$, where \mathbf{Y} is a vector of random variables that model uncertainties in the underlying system, and \mathbf{x} is a vector of decisions. In portfolio optimization, \mathbf{Y} would be future share prices for all companies traded in the stock market and \mathbf{x} the portfolio allocation. For each decision \mathbf{x} , the loss function $L(\mathbf{x}, \mathbf{Y})$ is itself a random variable. Generally gains are modeled as negative losses. In portfolio optimization, the loss is often measured in terms of

Figure 2.1: An example of VaR and CVaR



return on investment (ROI), where the loss is negative if the ROI is positive.

A recent paper (Madadi et al., 2010) formulates a CVaR-based supply chain network model in order to minimize the risk of defective pharmaceuticals in a medical supply chain. Supplier failures and resulting defective pharmaceuticals are the sources of uncertainty in this network. The demand at the customer nodes is fixed. In the transportation network the arcs and their costs are deterministic. The objective function minimizes the CVaR of the loss from defective pharmaceuticals. The constraints trade off the cost of inspection versus the cost of defective products. They enumerate all possible scenarios and assign them a probability. This was computationally feasible because the number of suppliers is likely to be small compared to the number of customers. One strength of their model is that it allows for a state of partial failure. This paper uses CVaR for managing production-side risks. In contrast to this thesis, modeling uncertainty in the arcs of the underlying

transportation network is not within the scope of Madadi et al. (2010)'s paper.

2.5 Formal Definition of CVaR

Definition 2.5.1 (Rockafellar and Uryasev (2002)). *Let $\mathbf{x} \in X \subset \mathbb{R}^n$ be a vector of decision variables and $\mathbf{Y} : \Omega \rightarrow \mathbb{R}^m$ be a vector of random variables defined on the sample space Ω . Then the loss function $L(\mathbf{x}, \mathbf{Y})$ is a function of the decisions \mathbf{x} and the random variables \mathbf{Y} .*

The random variable $L(\mathbf{x}, \mathbf{Y})$ has a family of distributions, one for each $\mathbf{x} \in X$. The sample space of $L(\mathbf{x}, \mathbf{Y})$ is denoted by $\Omega_L(\mathbf{x})$. Given a decision \mathbf{x} , denote the probability density function or the probability mass function of $L(\mathbf{x}, \mathbf{Y})$ by $\phi_L(\ell; \mathbf{x}, \mathbf{Y})$ where $\ell \in \Omega_L(\mathbf{x})$ and the cumulative distribution function by $\Phi_L(\ell; \mathbf{x}, \mathbf{Y})$.

Note 2.5.2 (Finite Loss Functions). *For each loss function discussed in this thesis, $E\{L(\mathbf{x}, \mathbf{Y})\} < \infty$ for all $\mathbf{x} \in X$.*

The assumption that the uncertainties modeled by \mathbf{Y} are not affected by the choice of \mathbf{x} is important (Rockafellar and Uryasev, 2002). For instance, in a financial setting, this assumption implies that stock prices fluctuate randomly regardless of who owns how much of each stock. Throughout this thesis the distribution of \mathbf{Y} , the random vector of arc failures, is assumed to be unaffected by the choice of path \mathbf{x} . In addition, it is assumed that arc failures are independent of each other. In particular, for each arc $(ij) \in A$, Y_{ij} is a Bernoulli random variable, and $\mathbf{Y} : \Omega \rightarrow \{0, 1\}^{|A|}$ (see Section 3.1). Modeling of loss functions in the context of the SSPP-PAF is discussed extensively in Section 3.3 of this thesis.

The $\text{VaR}(\mathbf{x}, \beta)$ is the quantile associated with the threshold probability β of the distribution of $L(\mathbf{x}, \mathbf{Y})$ and is defined as follows:

Definition 2.5.3 (Rockafellar and Uryasev (2002)).

$$\text{VaR}(\mathbf{x}, \beta) = \min \{ \ell \mid \Phi_L(\ell; \mathbf{x}, \mathbf{Y}) \geq \beta \} \quad (2.1)$$

Definition 2.5.4 (Rockafellar and Uryasev (2002)). Given a threshold percentile β , the cumulative upper β -tail distribution, denoted by $\Phi_{\beta,L}(\ell; \mathbf{x}, \mathbf{Y})$, is given by:

$$\Phi_{\beta,L}(\ell; \mathbf{x}, \mathbf{Y}) = \begin{cases} 0 & \text{if } \ell < \text{VaR}(\mathbf{x}, \beta) \\ \frac{\Phi_L(\ell; \mathbf{x}, \mathbf{Y}) - \beta}{1 - \beta} & \text{if } \ell \geq \text{VaR}(\mathbf{x}, \beta) \end{cases} \quad (2.2)$$

Definition 2.5.5 (Rockafellar and Uryasev (2002)). For a given threshold percentile β , the $\text{CVaR}(\mathbf{x}, \beta)$ of the loss associated with decision \mathbf{x} is given by:

$$\text{CVaR}(\mathbf{x}, \beta) = \mathbb{E} \{L(\mathbf{x}, \mathbf{Y}) | L(\mathbf{x}, \mathbf{Y}) \geq \text{VaR}(\mathbf{x}, \beta)\} \quad (2.3)$$

In this thesis the random vector \mathbf{Y} represents arc failures, so the distribution of \mathbf{Y} is discrete. Hence, the loss functions induced are also discrete distributions. For loss functions with finite discrete distributions, the CVaR of the loss is defined below.

Proposition 2.5.6. (Proposition 8, Rockafellar and Uryasev (2002)) Assume that the sample space of \mathbf{Y} is finite, so that for each $\mathbf{x} \in X$ the distribution of $L(\mathbf{x}, \mathbf{Y})$ is likewise discrete and finite and $\Phi_L(\ell; \mathbf{x}, \mathbf{Y})$ is a step function with jumps at $\ell \in \Omega_L(\mathbf{x})$. For a fixed \mathbf{x} , order $\ell \in \Omega_L$ such that $\ell_0 < \ell_1 < \dots < \ell_{\max}$. Let ℓ_k be the unique element of Ω_L such that

$$\Phi_L(\ell_{k-1}; \mathbf{x}, \mathbf{Y}) < \beta \leq \Phi_L(\ell_k; \mathbf{x}, \mathbf{Y}). \quad (2.4)$$

The $\text{VaR}(\mathbf{x}, \beta)$ of the loss is given by

$$\text{VaR}(\mathbf{x}, \beta) = \ell_k. \quad (2.5)$$

In addition, the $\text{CVaR}(\mathbf{x}, \beta)$ is given by

$$\text{CVaR}(\mathbf{x}, \beta) = \frac{1}{1 - \beta} \left[(\Phi_L(\ell; \mathbf{x}, \mathbf{Y}) - \beta) \ell_k + \sum_{\ell=\ell_{k+1}}^{\ell_{\max}} \ell \cdot \phi_L(\ell; \mathbf{x}, \mathbf{Y}) \right]. \quad (2.6)$$

2.6 Using CVaR in Optimization

The minimization formulas below were developed by Rockafellar and Uryasev (2000, 2002) and rely on the convexity of CVaR.

Theorem 2.6.1 (Rockafellar and Uryasev (2002)). *Let*

$$F_\beta(\mathbf{x}, \zeta) = \zeta + \frac{1}{1-\beta} \mathbb{E} \left\{ [L(\mathbf{x}, \mathbf{Y}) - \zeta]^+ \right\}, \quad (2.7)$$

where $[z]^+ = \max\{0, z\}$. $F_\beta(\mathbf{x}, \zeta)$ is finite and convex as a function of $\zeta \in \mathbb{R}$ and

$$\text{CVaR}(\mathbf{x}, \beta) = \min_{\zeta} F_\beta(\mathbf{x}, \zeta). \quad (2.8)$$

Denote the set of minimizers of $F_\beta(\mathbf{x}, \zeta)$ by Z . This set is a nonempty, closed interval, reducing to a single point if the minimum ζ^* of $F_\beta(\mathbf{x}, \zeta)$ is unique, and

$$\text{VaR}(\mathbf{x}, \beta) = \min\{\zeta \in Z\}. \quad (2.9)$$

In addition,

$$\text{CVaR}(\mathbf{x}, \beta) = F_\beta(\text{VaR}(\mathbf{x}, \beta)). \quad (2.10)$$

Since $F_\beta(\mathbf{x}, \zeta)$ is convex, it may be used in optimization problems. Specifically, the next theorem and corollary show that $F_\beta(\mathbf{x}, \zeta)$ can be used to minimize $\text{CVaR}(\mathbf{x}, \beta)$ in the objective function or part of the constraints, which is how CVaR is used in the CVARSPP.

Theorem 2.6.2 (Rockafellar and Uryasev (2002)). *Minimizing $\text{CVaR}(\mathbf{x}, \beta)$ over $\mathbf{x} \in X$ is equivalent to minimizing $F_\beta(\mathbf{x}, \zeta)$ over $X \times \mathbb{R}$ since*

$$\min_{\mathbf{x} \in X} \text{CVaR}(\mathbf{x}, \beta) = \min_{(\mathbf{x}, \zeta) \in X \times \mathbb{R}} F_\beta(\mathbf{x}, \zeta). \quad (2.11)$$

In addition, $(\mathbf{x}^*, \zeta^*) \in X \times \mathbb{R}$ are minimizers of $F_\beta(\mathbf{x}, \zeta)$ (not necessarily unique) if and only if $\mathbf{x}^* \in X$ is a minimizer of $\text{CVaR}(\mathbf{x}, \beta)$ and ζ^* is a minimizer of $F_\beta(\mathbf{x}^*, \zeta)$.

Since $F_\beta(\mathbf{x}, \zeta)$ is a convex piecewise linear function of ζ , it can be used as a constraint or objective function in an optimization problem. This is helpful since the direct formula in Proposition 2.5.6 requires computing $\text{VaR}(\mathbf{x}, \beta)$ in advance, which is usually difficult (Rockafellar and Uryasev, 2002). Using $F_\beta(\mathbf{x}, \zeta)$ instead allows the use of CVaR in optimization without having to compute the $\text{VaR}(\mathbf{x}, \beta)$ in advance, as the corollary below mentions. The optimal solution found by minimizing $\text{CVaR}(\mathbf{x}, \beta)$ via the minimization formula $F_\beta(\mathbf{x}, \zeta)$ will contain information about $\text{VaR}(\mathbf{x}, \beta)$ as a byproduct.

Corollary 2.6.3 (Rockafellar and Uryasev (2002)). *If (\mathbf{x}^*, ζ^*) minimizes $F_\beta(\mathbf{x}, \zeta)$ over $X \times \mathbb{R}$, then \mathbf{x}^* minimizes $\text{CVaR}(\mathbf{x}, \beta)$ over X and*

$$\text{CVaR}(\mathbf{x}^*, \beta) = F_\beta(\mathbf{x}^*, \zeta^*) \quad (2.12)$$

and

$$\text{VaR}(\mathbf{x}^*, \beta) \leq \zeta^*, \quad (2.13)$$

where equality holds if the set of minimizers Z of $F_\beta(\mathbf{x}, \zeta)$ (see Theorem 2.6.1) reduces to a single point.

CHAPTER 3

Model Development and Analysis

3.1 Modeling Arc Failures

Throughout this thesis the simplifying assumption is made that failures are independent of each other. The probability that arc $(ij) \in A$ fails is p_{ij} where $0 \leq p_{ij} < 1$. Arcs for which $p_{ij} = 0$ never fail. The random vector of arc failures, \mathbf{Y} , is of length $|A|$. Each component of \mathbf{Y} is a random variable Y_{ij} . If arc (ij) fails for some $\mathbf{y}^s \in \Omega$, then $y_{ij}^s = 0$. In other words, Y_{ij} is a Bernoulli random variable.

$$Y_{ij} \sim \text{Bernoulli}(1 - p_{ij}) \quad \forall (ij) \in A \quad (3.1)$$

3.2 Formulation of the CVARSPP

This model is an extension of the deterministic LP model of the shortest path problem in Section 1.2.3. It simply adds a constraint on CVaR to the LP model. When solved to optimality, a shortest path that satisfies the CVaR constraint will be chosen. Unfortunately, adding a CVaR constraint generally destroys the integrality of the shortest path polytope, so it is necessary to solve the problem as a MILP, greatly increasing solution time.

3.2.1 Additional Parameters

In addition to the notation used in Formulation 1.2.2, the following notation is used throughout the rest of this thesis. Note that the set of arcs A is random. In the formulation, we denote by A the set of arcs such that $p_{ij} < 1$.

- C is the maximum acceptable CVaR of loss.
- β is the percentile for the upper tail (e.g. 0.95, 0.99)
- $S \subseteq \Omega$ is the set of scenarios in the model, indexed by s . S may be a random sample from Ω or the whole sample space if Ω is small.
- p^s is the probability of scenario s , normalized so that $\sum_{s \in S} p^s = 1$. For equiprobable scenarios, set $p^s = \frac{1}{|S|} \quad \forall s \in S$.
- \mathbf{y}^s is the realization of the random vector \mathbf{Y} corresponding to scenario $s \in S$.

3.2.2 CVARSPP Optimization Model

Formulation 3.2.1.

$$\text{Minimize } \sum_{(ij) \in A} c_{ij} x_{ij} \quad (3.2)$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (3.3)$$

$$\text{CVaR}(\mathbf{x}, \beta) \leq C \quad (3.4)$$

$$\mathbf{x} \in \{0, 1\}^{|A|} \quad (3.5)$$

Constraint 3.4 can be rewritten in terms of the minimization formula $F_\beta(\mathbf{x}, \zeta)$ from Theorem 2.6.1 as in the formulation below.

Additional Decision Variables

- $\zeta \in \mathbb{R}$ is $\text{VaR}(\mathbf{x}, \beta)$ at optimality.
- z^s linearizes the excess loss $[L(\mathbf{x}, \mathbf{y}^s) - \zeta]^+$ in scenario $s \in S$.

Formulation 3.2.2.

$$\text{Minimize } \sum_{(ij) \in A} c_{ij} x_{ij} \quad (3.6)$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (3.7)$$

$$F_\beta(\mathbf{x}, \zeta) \leq C \quad (3.8)$$

$$\mathbf{x} \in \{0, 1\}^{|A|} \quad (3.9)$$

$$\zeta \in \mathbb{R} \quad (3.10)$$

In Constraint 3.8,

$$F_\beta(\mathbf{x}, \zeta) = \zeta + \frac{1}{1 - \beta} \sum_{s \in S} p^s [L(\mathbf{x}, \mathbf{y}^s) - \zeta]^+. \quad (3.11)$$

Constraint 3.8 can be linearized using standard techniques and the distribution of $L(\mathbf{x}, \mathbf{Y})$ approximated by sampling to produce Formulation 3.2.3. As the loss functions are developed in the next section, the CVARSPP models for each loss function will be based on this formulation.

Formulation 3.2.3.

$$\text{Minimize } \sum_{(ij) \in A} c_{ij} x_{ij} \quad (3.12)$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (3.13)$$

$$\zeta + \frac{1}{1 - \beta} \sum_{s \in S} p^s z^s \leq C \quad (3.14)$$

$$z^s \geq L(\mathbf{x}, \mathbf{y}^s) - \zeta \quad \forall s \in S \quad (3.15)$$

$$z^s \geq 0 \quad \forall s \in S \quad (3.16)$$

$$\mathbf{x} \in \{0, 1\}^{|A|} \quad (3.17)$$

$$\zeta \in \mathbb{R} \quad (3.18)$$

3.3 Loss Functions

In this section three different loss function are introduced. Each loss function corresponds to a different effect of arc failures on the path. While the concepts behind these loss functions have appeared in the literature before, explicitly formulating and analyzing them as a part of a CVaR optimization model is the main contribution of this thesis. All the loss functions measure properties of an $s - t$ path \mathbf{x} . The first loss function is called the *Path Reliability Loss Function* and measures the reliability of the path. The second function is the *Arc Failure Loss Function*, which measures the number arc failures. The final loss function is the *Detours Loss Function*, which counts the number of detours around broken segments of a path.

3.3.1 Path Reliability

The first loss function to be considered derives from the basic physical interpretation of the problem. It asks “Does the path break in the current scenario?” In

other words, the loss is 0 if no arc fails on the current path and 1 otherwise. This loss function is worth looking at because it is simple and comparable to other SSPP approaches such as Ahuja (1988). This is the loss function implicit in the Most Reliable Path Problem, which finds the path with the lowest probability of failure (Ball et al., 1995b). In Figure 3.1 the path reliability loss function measures a loss of one.



Figure 3.1: Path Reliability Loss Function: The loss for this path is 1.

Definition 3.3.1 (Path Reliability Loss Function).

$$L^r(\mathbf{x}, \mathbf{Y}) = \max_{(ij) \in A} x_{ij}(1 - Y_{ij}) \quad (3.19)$$

Remark 3.3.2. $L^r(\mathbf{x}, \mathbf{Y}) = 0$ if no arcs fail and 1 otherwise.

$$L^r(\mathbf{x}, \mathbf{Y}) \sim \text{Bernoulli}(q_0) \quad (3.20)$$

where $q_0 = \prod_{(ij) \in A(\mathbf{x})} (1 - p_{ij})$ for all $(ij) \in A$.

Note that the path reliability loss function is piecewise linear and convex in \mathbf{x} , so Formulation 3.2.3 can be further linearized to give the MILP below.

Formulation 3.3.3 (Path Reliability Loss Function).

$$\text{Minimize } \sum_{(ij) \in A} c_{ij} x_{ij} \quad (3.21)$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (3.22)$$

$$\zeta + \frac{1}{1-\beta} \sum_{s \in S} p^s z^s \leq C \quad (3.23)$$

$$z^s \geq x_{ij}(1 - y_{ij}^s) - \zeta \quad \forall s \in S \quad (3.24)$$

$$z^s \geq 0 \quad \forall s \in S \quad (3.25)$$

$$\mathbf{x} \in \{0, 1\}^{|A|}; \quad \zeta \in \mathbb{R} \quad (3.26)$$

3.3.2 Number of Arc Failures

The second loss function counts the number of arcs that fail on the path. For a decision-maker who is responsible for repairing the path, this loss function gives an estimate of the amount of work to be done. In Figure 3.2, which is the same path as Figure 3.1, the arc failure loss function measures a loss of three. The drawback to this loss function is that not all arc failures, even on the path, may have an equal impact on the path length. For example, in reality it may be very easy to avoid one broken link.



Figure 3.2: Arc Failures Loss Function: The loss for this path is 3.

Definition 3.3.4 (Arc Failure Loss Function).

$$L^a(\mathbf{x}, \mathbf{Y}) = \sum_{(ij) \in A} x_{ij}(1 - Y_{ij}) \quad (3.27)$$

Remark 3.3.5. If $p_{ij} = p$ for all $(ij) \in A$, then $L^a(\mathbf{x}, \mathbf{Y})$ is the sum of Bernoulli random variables and

$$L^a(\mathbf{x}, \mathbf{Y}) \sim \text{Bin}(|A(\mathbf{x})|, 1 - p) \quad (3.28)$$

Since the arc failure loss function is linear, Formulation 3.2.3 reduces to the MILP below.

Formulation 3.3.6 (Arc Failure Loss Function).

$$\text{Minimize } \sum_{(ij) \in A} c_{ij} x_{ij} \quad (3.29)$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (3.30)$$

$$\zeta + \frac{1}{1-\beta} \sum_{s \in S} p^s z^s \leq C \quad (3.31)$$

$$z^s \geq \sum_{(ij) \in A} x_{ij} (1 - y_{ij}^s) - \zeta \quad \forall s \in S \quad (3.32)$$

$$\mathbf{x} \in \{0, 1\}^{|A|}; \quad \mathbf{z} \in \mathbb{R}_+^{|S|}; \quad \zeta \in \mathbb{R} \quad (3.33)$$

3.3.3 Number of Detours

The final loss function considered in this thesis counts the number of gaps in the current path. A gap is any set of consecutive failed arcs, such as arcs (2,3) and (3,4) in Figure 3.3. This loss function is related to ideas from Lagrangian relaxation and derives from the formulation of the problem. Mathematically the loss function measures total violation of flow conservation due to chance in the form of arc failures. This is a useful loss function to examine because it is common to measure the violation of the constraints in stochastic programming, particularly in terms of basic recourse. Figure 3.3 shows that on the same path as Figure 3.1 and Figure 3.2 the detours loss function measures a loss of 2.



Figure 3.3: Detours Loss Function: The loss for this path is 2.

Definition 3.3.7 (Detours Loss Function). *The loss function is*

$$L^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} \sum_{i \in N} \left| \sum_{j \in N^+(i)} x_{ij} Y_{ij} - \sum_{j \in N^-(i)} x_{ji} Y_{ji} - b_i \right| \quad (3.34)$$

where $b_s = 1$, $b_t = -1$ and $b_i = 0$ for all $i \in N \setminus \{s, t\}$. This loss function measures violation of the flow balance constraints or number of detours needed.

The loss at some node $i \in N$ is

$$L_i^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} \left| \sum_{j \in N^+(i)} x_{ij} Y_{ij} - \sum_{j \in N^-(i)} x_{ji} Y_{ji} - b_i \right|, \quad (3.35)$$

and

$$L^d(\mathbf{x}, \mathbf{Y}) = \sum_{i \in N} L_i^d(\mathbf{x}, \mathbf{Y}). \quad (3.36)$$

Since flow conservation will only be violated at the ends of each gap in the path, this loss function ends up counting the number of gaps. The loss function can be interpreted physically as the number of detours necessary to avoid all the gaps. Since one failed arc renders the following arc inaccessible, the failure of that second arc does not cause additional loss. Until all the consecutive failed arcs have all been repaired, that gap in the path must still be detoured around.

Lemma 3.3.8. *For any incidence vector \mathbf{x} of an $s - t$ path,*

$$L^d(\mathbf{x}, \mathbf{Y}) = \sum_{i \in N(\mathbf{x})} L_i^d(\mathbf{x}, \mathbf{Y})$$

Proof. A feasible $s - t$ path \mathbf{x} corresponds to a graph $G(\mathbf{x}) = (N(\mathbf{x}), A(\mathbf{x}))$. Nodes $i \notin N(\mathbf{x})$ have no arcs incident at them, and $s, t \in N(\mathbf{x})$. Hence,

$$L_i^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} \left| \sum_{j \in N^+(i)} x_{ij} Y_{ij} - \sum_{j \in N^-(i)} x_{ji} Y_{ji} - b_i \right| \quad \forall i \notin N(\mathbf{x}) \quad (3.37)$$

$$= \frac{1}{2} \left| \sum_{j \in N^+(i)} 0 \cdot Y_{ji} - \sum_{j \in N^-(i)} 0 \cdot Y_{ij} - 0 \right| \quad (3.38)$$

$$= 0 \quad (3.39)$$

So

$$L^d(\mathbf{x}, \mathbf{Y}) = \sum_{i \in N(\mathbf{x})} L_i^d(\mathbf{x}, \mathbf{Y}) \quad (3.40)$$

□

Lemma 3.3.9. *If \mathbf{x} is an incidence vector of an $s - t$ path, then*

$$L_s^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} |Y_{si} - 1| \quad \text{where } i \in N^+(s) \cap N(\mathbf{x}), \quad (3.41)$$

$$L_j^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} |Y_{ij} - Y_{jk}| \quad \forall j \in N(\mathbf{x}) \setminus \{s, t\}, \quad (3.42)$$

where $i \in N^-(j) \cap N(\mathbf{x})$ and $k \in N^+(j) \cap N(\mathbf{x})$, and

$$L_t^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} |1 - Y_{jt}| \quad \text{where } j \in N^-(t) \cap N(\mathbf{x}). \quad (3.43)$$

Proof. A feasible $s - t$ path \mathbf{x} corresponds to a graph $G(\mathbf{x}) = (N(\mathbf{x}), A(\mathbf{x}))$ such that one arc incident must be incident to s and t and two arcs to all other nodes $i \in N(\mathbf{x}) \setminus \{s, t\}$.

At s , $N^-(s) = \emptyset$ and $b_s = 1$. Also, $x_{si} = 0$ if $i \notin N(\mathbf{x})$.

$$L_s^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} \left| \sum_{j \in N^+(s)} x_{sj} Y_{sj} - 0 - b_s \right| \quad (3.44)$$

$$= \frac{1}{2} |1 \cdot Y_{sk} + 0 - 1| \quad \text{for some } k \in N^+(s) \quad (3.45)$$

$$= \frac{1}{2} |Y_{sk} - 1| \quad (3.46)$$

Similarly, at t , there exists some $k \in N^-(t) \cap N(\mathbf{x})$ such that,

$$L_i^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} |0 - 1 \cdot Y_{kt} - (-1)| \quad (3.47)$$

$$= \frac{1}{2} |1 - Y_{kt}| \quad (3.48)$$

For $j \in N(\mathbf{x}) \setminus \{s, t\}$, let $N^-(j) \cap N(\mathbf{x}) = \{i\}$, and $N^+(j) \cap N(\mathbf{x}) = \{k\}$.

$$L_j^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} \left| \sum_{u \in N^-(j)} x_{uj} Y_{uj} - \sum_{v \in N^+(j)} x_{jv} Y_{jv} - b_j \right| \quad (3.49)$$

$$= \frac{1}{2} |1 \cdot Y_{ij} - 1 \cdot Y_{jk} - 0| \quad (3.50)$$

$$= \frac{1}{2} |Y_{ij} - Y_{jk}| \quad (3.51)$$

□

Corollary 3.3.10. For all $j \in N(\mathbf{x}) \setminus \{s, t\}$, $L_j^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2}$ if $Y_{ij} \neq Y_{jk}$; that is, if one arc fails and the other does not. If $Y_{ij} = Y_{jk}$, then $L_j^d(\mathbf{x}, \mathbf{Y}) = 0$. $L_s^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2}$ if $Y_{si} = 0$, and similarly for t .

Each gap consisting of consecutive nodes $v_1 \dots v_k \in N(\mathbf{x})$ results in a loss of

$$\sum_{i=1}^k L_{v_i}^d(\mathbf{x}, \mathbf{Y}) = 1. \quad (3.52)$$

Corollary 3.3.11. The sample space for $L^d(\mathbf{x}, \mathbf{Y})$ is $\Omega_d = \{0, 1, \dots, \lfloor \frac{|N(\mathbf{x})|}{2} \rfloor\}$

Distribution of Detours Loss Function

Theorem 3.3.12. Suppose that all arcs have an equal probability of failure; that is, $p_{ij} = p$ for all $(ij) \in A$. In that case, the pmf of the detours loss function $L^d(\mathbf{x}, \mathbf{Y})$ is given by

$$\phi_L(\ell; \mathbf{x}, \mathbf{Y}) = \begin{cases} q^{|A(\mathbf{x})|} & \text{when } \ell = 0, \\ \sum_{i=\ell}^{|A(\mathbf{x})|-\ell+1} \sum_{j=\ell-1}^{|A(\mathbf{x})|-i} p^i q^{|A(\mathbf{x})|-i} \binom{i-1}{\ell-1} \binom{j-1}{\ell-2} (|A(\mathbf{x})| - i - j + 1) & \\ & \text{when } \ell = 1 \dots \lfloor \frac{|N(\mathbf{x})|}{2} \rfloor, \text{ and} \\ 0 & \text{otherwise,} \end{cases} \quad (3.53)$$

where $q = 1 - p$.

Proof. From Corollary 3.3.11, we know that there are between 0 and $\lfloor \frac{|N(\mathbf{x})|}{2} \rfloor$ gaps in the path subgraph $G(\mathbf{x}) = (N(\mathbf{x}), A(\mathbf{x}))$. Let us then consider the number of ways to distribute the failed arcs into gaps and the surviving arcs into “fragments”.

Let $L^d(\mathbf{x}, \mathbf{Y}) = \ell \in \Omega_d$ be the loss. Suppose $\ell > 0$. Denote by i the number of failed arcs. By Corollary 3.3.10, the failed arcs must be distributed into exactly ℓ gaps with at least one arc in each gap, which implies that $i \geq \ell$. The number of ways to distribute the i failed arcs into ℓ gaps with at least one arc in each gap is $\binom{i-1}{\ell-1}$.

Denote by j the total number of surviving arcs between the first and last failed arcs. Between each of the ℓ gaps are $\ell - 1$ “fragments” of at least one surviving arc each, which implies that $\ell - 1 \leq j \leq |A(\mathbf{x})| - i$. The number of ways to distribute the j surviving arcs into $\ell - 1$ fragments with at least one arc in each fragment is $\binom{j-1}{\ell-2}$.

The length of the entire path segment from the first to the last failed arc is $i + j \leq |A(\mathbf{x})|$. This segment may start anywhere in the first $|A(\mathbf{x})| - i - j + 1$ arcs. The total number of ways to arrange i failed arcs with $j \leq |A(\mathbf{x})| - i$ surviving arcs between the failed arcs into ℓ gaps with $\ell - 1$ fragments between them is

$$\binom{i-1}{\ell-1} \binom{j-1}{\ell-2} (|A(\mathbf{x})| - i - j + 1). \quad (3.54)$$

The probability of i arcs failing and a loss of ℓ is then

$$P(L(\mathbf{x}, \mathbf{Y}) = \ell \text{ and } \sum_{(ij) \in A(\mathbf{x})} Y_{ij} = i) = p^i q^{|A(\mathbf{x})| - i} \sum_{j=\ell-1}^{|A(\mathbf{x})| - i} \binom{i-1}{\ell-1} \binom{j-1}{\ell-2} (|A(\mathbf{x})| - i - j + 1). \quad (3.55)$$

Hence the total probability of a loss of ℓ is

$$P(L(\mathbf{x}, \mathbf{Y}) = \ell) = \sum_{i=\ell}^{|A(\mathbf{x})| - \ell + 1} \sum_{j=\ell-1}^{|A(\mathbf{x})| - i} p^i q^{|A(\mathbf{x})| - i} \binom{i-1}{\ell-1} \binom{j-1}{\ell-2} (|A(\mathbf{x})| - i - j + 1). \quad (3.56)$$

The case of $L(\mathbf{x}, \mathbf{Y}) = 0$ is a special case. The only possible number of arc failures is $i = 0$, and all arcs must fall into a single fragment. The probability of $\ell = 0$ is therefore

$$\phi_L(0; \mathbf{x}, \mathbf{Y}) = q^{|\mathcal{A}(\mathbf{x})|} \quad (3.57)$$

□

As was the case with the path reliability loss function, the detours loss function is piecewise linear and convex.

Formulation 3.3.13 (Detours Loss Function).

$$\text{Minimize } \sum_{(ij) \in \mathcal{A}} c_{ij} x_{ij} \quad (3.58)$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (3.59)$$

$$\zeta + \frac{1}{1 - \beta} \sum_{s \in S} p^s z^s \leq C \quad (3.60)$$

$$z^s \geq \sum_{i \in N} \ell_i^s - \zeta \quad \forall s \in S \quad (3.61)$$

$$\ell_i^s \geq \sum_{j \in N^-(i)} x_{ji} y_{ji}^s - \sum_{j \in N^+(i)} x_{ij} y_{ij}^s - b_i \quad (3.62)$$

$$\ell_i^s \geq b_i + \sum_{j \in N^+(i)} x_{ij} y_{ij}^s - \sum_{j \in N^-(i)} x_{ji} y_{ji}^s \quad (3.63)$$

$$z^s \geq 0 \quad \forall s \in S \quad (3.64)$$

$$\mathbf{x} \in \{0, 1\}^{|\mathcal{A}|}; \quad \ell \in \mathbb{R}^{|S|} \times \mathbb{R}^{|N|}; \quad \zeta \in \mathbb{R} \quad (3.65)$$

Table 3.1 summarizes the different loss functions.

Table 3.1: Summary of Loss Functions (LF)

Name	Formulation	Sample Space
Path Reliability LF	$L^r(\mathbf{x}, \mathbf{Y}) = \max_{(ij) \in A} x_{ij}(1 - Y_{ij})$	$\{0, 1\}$
Arc Failure LF	$L^a(\mathbf{x}, \mathbf{Y}) = \sum_{(ij) \in A} x_{ij}(1 - Y_{ij})$	$\{0, \dots, A(\mathbf{x}) \}$
Detours LF	$L^d(\mathbf{x}, \mathbf{Y}) = \frac{1}{2} \sum_{i \in N} \left \sum_{j \in N^-(i)} x_{ji} Y_{ji} - \sum_{j \in N^+(i)} x_{ij} Y_{ij} - b_i \right $	$\{0, \dots, \lfloor \frac{ N(\mathbf{x}) }{2} \rfloor\}$

3.4 Comparison of Loss Functions

3.4.1 Magnitude of Losses

As shown in Table 3.1, the sample space for the path reliability loss function is a subset of the sample space for the detours loss function, which is in turn a subset of the sample space for the arc failure loss function. Furthermore, the loss as measured by the arc failure loss function will always be greater than or equal to the loss measured by the detours loss function, which will in turn be greater than or equal to the loss measured by the path reliability loss function for the same realization $\mathbf{y}^s \in \Omega$.

Theorem 3.4.1. *For all feasible paths \mathbf{x} and realizations of the random vector of arc failures $\mathbf{y}^s \in \Omega$,*

$$L^r(\mathbf{x}, \mathbf{y}^s) \leq L^d(\mathbf{x}, \mathbf{y}^s) \leq L^a(\mathbf{x}, \mathbf{y}^s). \quad (3.66)$$

Proof. Consider an arbitrary feasible $s - t$ path, corresponding to the incidence vector \mathbf{x} , and \mathbf{y}^s , a realization of \mathbf{Y} . Let $\ell_r = L^r(\mathbf{x}, \mathbf{y}^s)$, $\ell_d = L^d(\mathbf{x}, \mathbf{y}^s)$, and $\ell_a = L^a(\mathbf{x}, \mathbf{y}^s)$. If no two arcs fail consecutively, then each arc failure is its own gap in the path, and $\ell_d = \ell_a$. Otherwise at least two arcs will fail in a row, and $\ell_d < \ell_a$. Therefore $L^d(\mathbf{x}, \mathbf{y}^s) \leq L^a(\mathbf{x}, \mathbf{y}^s)$.

If $\ell_d = 0$, no arcs have failed, so $\ell_r = 0$ and $\ell_d = \ell_r$. If $\ell_d \geq 1$, then at least one arc has failed, and $\ell_r = 1$. When $\ell_d \geq 2$, therefore, $\ell_d > \ell_m$. That is, $L^r(\mathbf{x}, \mathbf{y}^s) \leq L^d(\mathbf{x}, \mathbf{y}^s)$. \square

3.4.2 CVaR as a function of β

Before beginning the next chapter, it is useful to examine how the choice of β affects $\text{CVaR}(\mathbf{x}, \beta)$. This section gives some guidance that will be used to check the results in the next chapter. Theorem 3.4.2 shows that $\text{CVaR}(\mathbf{x}, \beta)$ is piecewise hyperbolic in β but not necessarily convex in β .

Theorem 3.4.2. *Let $L(\mathbf{x}, \mathbf{Y})$ be a loss function with a finite discrete distribution. Given a fixed feasible path \mathbf{x} , the corresponding loss distribution is $\phi_L(\ell; \mathbf{x}, \mathbf{Y})$ with sample space $\Omega_L(\mathbf{x}) = \{0, 1 \dots \ell_{\max}\}$. For each $\ell \in \Omega_L(\mathbf{x})$, $\text{CVaR}(\mathbf{x}, \beta)$ satisfies:*

$$\text{CVaR}(\mathbf{x}, \beta) = \ell + \frac{\mathbb{E} \left\{ [L(\mathbf{x}, \mathbf{Y}) - \ell]^+ \right\}}{1 - \beta} \quad (3.67)$$

when $P(L(\mathbf{x}, \mathbf{Y}) < \ell) < \beta \leq P(L(\mathbf{x}, \mathbf{Y}) \leq \ell)$.

Proof. For each $\ell \in \Omega_L(\mathbf{x})$, let $\beta_\ell = P(L(\mathbf{x}, \mathbf{Y}) \leq \ell)$, let $\beta_{-1} = 0$, and consider $\text{CVaR}(\mathbf{x}, \beta)$ for each $\beta \in (\beta_{\ell-1}, \beta_\ell]$. Applying Proposition 2.5.6 gives:

$$\text{CVaR}(\mathbf{x}, \beta) = \frac{1}{1 - \beta} \left[\ell(\Phi_L(\ell; \mathbf{x}, \mathbf{Y}) - \beta) + \sum_{i=\ell+1}^{\ell_{\max}} i \cdot \phi_L(i; \mathbf{x}, \mathbf{Y}) \right] \quad (3.68)$$

$$= \frac{1}{1 - \beta} \left[\ell\Phi_L(\ell; \mathbf{x}, \mathbf{Y}) - \ell\beta + \sum_{i=\ell+1}^{\ell_{\max}} (i - \ell + \ell)\phi_L(i; \mathbf{x}, \mathbf{Y}) \right] \quad (3.69)$$

$$= \frac{1}{1 - \beta} \left[\ell\Phi_L(\ell; \mathbf{x}, \mathbf{Y}) - \ell\beta + \ell \sum_{i=\ell+1}^{\ell_{\max}} \phi_L(i; \mathbf{x}, \mathbf{Y}) + \sum_{i=\ell+1}^{\ell_{\max}} (i - \ell)\phi_L(i; \mathbf{x}, \mathbf{Y}) \right] \quad (3.70)$$

$$= \frac{1}{1 - \beta} \left[\ell(1 - \beta) + \sum_{i=\ell+1}^{\ell_{\max}} (i - \ell)\phi_L(i; \mathbf{x}, \mathbf{Y}) \right] \quad (3.71)$$

$$= \ell + \frac{1}{1 - \beta} \left[\sum_{i=\ell+1}^{\ell_{\max}} (i - \ell)\phi_L(i; \mathbf{x}, \mathbf{Y}) \right] \quad (3.72)$$

\square

CHAPTER 4

Computational Experiments

The experiments in this chapter are exploratory, meant to illustrate the use and effects of CVaR. The purpose of these computational experiments are to explore the behavior of the three loss functions as C and β vary, to compare the loss functions to each other, and to illustrate the difference between using CVaR and expected value to choose a path through an uncertain network.

4.1 Implementation Details

The experiments were performed on two HP xw4600 workstations with a quad-core 2.50 GHz Intel Core 2 CPU and 8 GB of RAM. The operating system on the first computer was Windows XP and on the other Windows Vista. The experiments were programmed using Python 2.7 (Van Rossum, 2003) and Gurobi 4.0.2 (Gu et al., 2010). The networks were implemented using the Networkx graph package (Hagberg et al., 2008). The Numpy matrix package (Oliphant, 2006) and the Scipy scientific computing package (Jones et al., 2001) were used for some calculations and for generating scenarios. Data was collected and stored in a Sqlite3 database using the SQLAlchemy object-relationship manager package (ORM) (Bayer et al., 2011). Finally, results were analyzed and visualized with a mix of Microsoft Access, Microsoft Excel, and Matplotlib (Hunter, 2007), a Python plotting package.

4.2 Experiments

Test instances were implemented as a directed network with source and sink attributes, a list of integer arc weights, a list of arc failure probabilities, and a set of scenarios. The network edges, arc weights, arc failure probabilities, and scenarios were all stored in the database using Python’s built-in pickle module. Since these experiments are exploratory, smaller instances were chosen. Preliminary experiments had found that instances of several hundred nodes took a long time to solve. Some of the larger and denser networks ran out of memory as well. The final test bed consisted of seven instances divided into three groups. The instances are summarized in Table 4.1.

Table 4.1: Summary of Test Bed

Name	Nodes	Arcs	Density	Scenarios	Notes
exp0002	6	9	30.0%	512	Complete enumeration
branch02	27	30	4.27%	100	Only one arc per branch fails.
branch03	27	30	4.27%	100	Less expensive paths are less reliable.
branch04	27	30	4.27%	100	$p_{ij} \sim Unif(0, 1), c_{ij} \sim DU(1, 100)$
gnp01	25	59	9.83%	100	$p_{ij} \sim Unif(0, 1), c_{ij} \sim DU(1, 100)$
gnp02	25	138	23.0%	100	$p_{ij} \sim Unif(0, 1), c_{ij} \sim DU(1, 100)$
gnp03	25	243	40.5%	100	$p_{ij} \sim Unif(0, 1), c_{ij} \sim DU(1, 100)$

For each instance, the deterministic shortest path (DSP), the most reliable path (MRP, the path with the lowest probability of failure), and the most direct path (MDP, the path with the fewest arcs from source to sink) were calculated and recorded. All the sets of scenarios were generated as random samples of size 100 and treated as equiprobable except exp0002. All scenarios for that instance were

enumerated and their exact probability used in optimization. Descriptions of each component of a test instance were also stored in the database (see Appendix B for details).

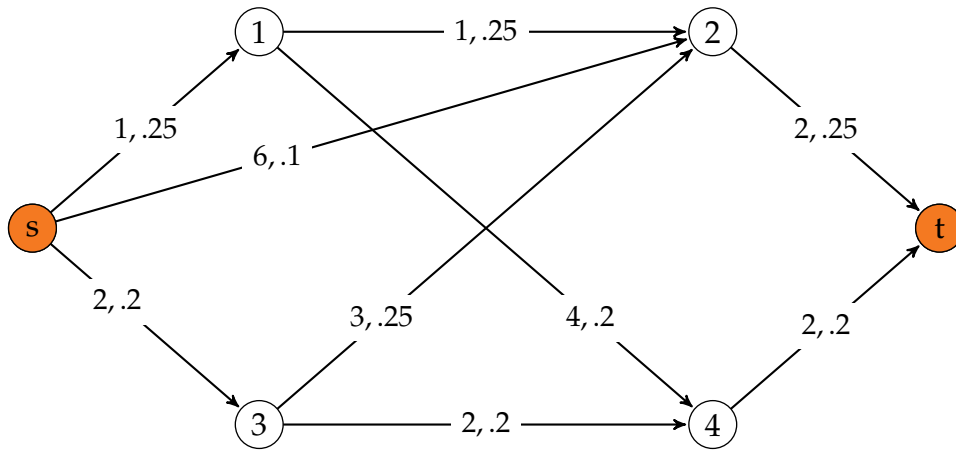
The main experiment mapped the level sets of the optimal cost as a function of β and C for each of the three loss functions. The full CVARSPP formulation for each loss function are in Chapter 3 and Appendix A. The parameter β was varied from 0 to 0.99 in increments of 0.01. The parameter C was varied from 0 to ℓ_{\max} in increments of $0.01\ell_{\max}$. The optimization status (infeasible, optimal, or unfinished), setup time, solve time, number of nodes, and number of integer feasible solutions found during optimization were recorded for each optimization run. If the run solved to optimality, the objective value, the values of each decision variable, and the arcs, cost, and reliability of the optimal path were recorded as well. For full details of what was recorded, see Appendix B.

4.2.1 Group 1: Complete Enumeration

The purpose of the first test network was to enable direct comparison with the analytical results in Section 3.4. The CVARSPP models were still small enough to solve in $<1s$ on average, despite containing all 512 scenarios. Since the network only contained 6 nodes and 9 arcs, it was possible to completely enumerate all $s - t$ paths. The arc costs and failure probabilities for each path were chosen to force the MDP to be the MRP and have the highest cost and the DSP to be the least reliable. Figure 4.1 contains a diagram of the resulting test network.

The lower edges of the level sets of the optimal path length are $\text{CVaR}(\mathbf{x}, \beta)$ for each path \mathbf{x} that was optimal for the region. The values predicted by Theorem 3.4.2 are plotted over the level sets in Appendix C and agree with the optimal solutions found during the experiment. The level sets are piecewise hyperbolic but not convex in general. The plot for the detours loss function in Figure C.2(b) shows that

Figure 4.1: Network exp0002



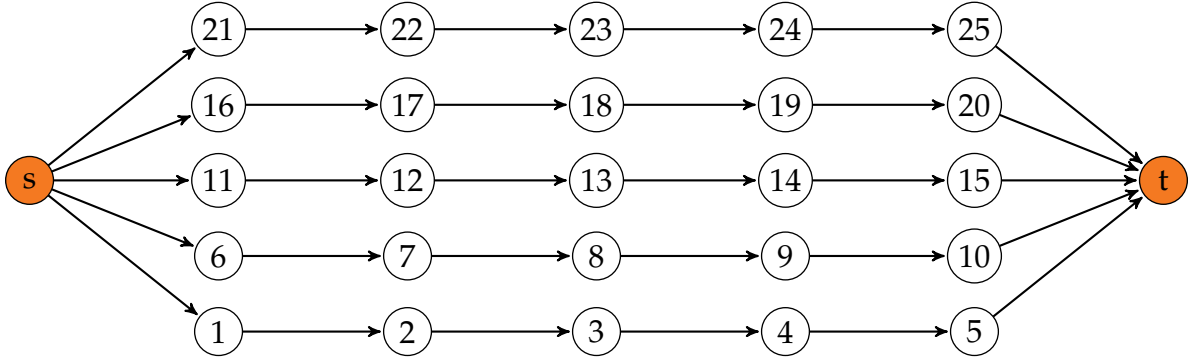
the lower edge of the MRP contour is flat at $\ell_{\max} = 1$ for $\beta \geq 0.675$. The other two paths represented on the plot have three arcs, so the lower edge of their level sets keep increasing toward $\ell_{\max} = 2$.

4.2.2 Group 2: Branch Networks

The effect of varying the arc failure probabilities on equal-length paths was examined in a branch network consisting of a source and sink node connected by five internally node-disjoint paths of five nodes each, as shown in Figure 4.2. In the first test network, branch02, only one arc was allowed to fail in each branch. As indicated by Theorem 3.4.1, all three loss functions gave the same results for each combination of β and C (Figures C.4(a), (b), and (c)).

In the second test network, branch03, the arc costs and arc failure probabilities were set so that more reliable paths also had higher cost (Figure D.2). For instance, each arc on the MRP had $c_{ij} = 5$ and $p_{ij} = 0.05$, each arc on the next most reliable path had $c_{ij} = 4$ and $p_{ij} = .1$, and so on down to the DSP, where each arc had $c_{ij} = 1$ and $p_{ij} = 0.25$. The goal of this setup was to see if each path would be optimal in some region of the (β, C) plane. This was indeed the case under all

Figure 4.2: Branch Network Structure



three loss functions.

In the final branch network, branch04, arc costs and arc failure probabilities were generated randomly for all arcs in order to observe how the CVARSPP models would respond to a wider range of arc reliabilities and path costs. The arc failure probabilities p_{ij} were drawn from a $\text{Unif}(0, 1)$ distribution and the costs c_{ij} from a $\text{DU}(1, 100)$ distribution. Since each branch is six arcs long, the expected reliability of each path is low. Even the MRP of branch04 failed in 98 out of 100 scenarios, giving an expected loss of 0.98 under the path failure loss function. When $\beta > 0.98$, the minimum choice of C that resulted in a feasible solution to the CVARSPP model under the path reliability loss function is $C = 1$ as shown in Figure C.8(a). The other two loss functions discriminated among paths, as $\mathbb{E}\{L^d(\mathbf{x}_{MRP}, \mathbf{y}^s)\} = 1.55$ and $\mathbb{E}\{L^a(\mathbf{x}_{MRP}, \mathbf{y}^s)\} = 2.05$. This result suggests that the path reliability loss function is better suited for networks with highly reliable arcs or shorter $s - t$ paths and should potentially be used with smaller β values.

4.2.3 Group 3: Uniform Random Networks

The final group of test networks are uniform random networks (Erdős and Rényi, 1959) of 25 nodes with different densities. The arc costs and failure probabilities were randomly generated from $p_{ij} \sim \text{Unif}(0, 1)$ and $c_{ij} \sim \text{DU}(1, 100)$. The purpose

of this group of test networks was to observe the effect of increasing densities and the behavior of the CVARSPP models on less controlled test networks. As the density increased from 0.0983 for gnp01 to 0.230 for gnp02 to 0.405 for gnp03, the number of paths that were feasible in some region of the (β, C) plane increased as well, from 2 to 4 to 6 (Figures C.10(c), C.12(c), and C.14(c)).

Network gnp02 also provided an illustration of the effects of sampling. The path corresponding to the region labeled 53 had two arcs, one of which failed in all 100 scenarios. That arc had a probability of failure of 0.96, so its failure in all scenarios is an artifact of sampling. This resulted in the path being infeasible for $C < 1$ under the path reliability loss function but being chosen under the detours and arc failure loss functions, as in Figure C.11(d). Under the detours loss function, though, $\text{CVaR}(\mathbf{x}_{53}, \beta) = 1$ for all $\beta \leq 1$. If the scenarios had been fully enumerated, the path would be feasible in some narrow region under the path reliability loss function.

4.3 Comparison of Loss Functions

To expand on Section 4.2.2 above, another feature of the the path reliability loss function is that, unless the network contains a path with no arcs that can fail, above $\beta = 1 - P(\mathbf{Y}_{\text{MRP}} = \mathbf{0})$ the instance will be infeasible for all $C < 1$. The CVARSPP model under the detours loss function has a feasible solution with $C < \ell_{\max}$ for all $\beta \leq 1$ in five out of seven networks. Under the arc failure loss function the CVARSPP model has a feasible solution with $C < \ell_{\max}$ for all $\beta \leq 1$ for all seven test networks. This suggests that β should be chosen somewhat lower and C somewhat higher within the range $(0, \ell_{\max})$, when using the path reliability loss function instead of the detours or arc failure loss functions. The detours loss function can use the same values of β as the arc failure loss function for most networks. Only in networks with higher arc failure probabilities does β need to be adjusted

slightly downward to preserve feasibility for $C < \ell_{\max}$. The CVaR limit C , on the other hand, should be chosen lower in the range $(0, \ell_{\max})$ than under the arc failure loss function, although precise guidance must wait on further computational experiments.

Figure C.2 shows how the average optimal path length, average optimal path reliability, average CVaR loss varied as β increased for the test instances using network exp0002. For each loss function, the instances which solved to optimality were grouped by β . Figure C.3(d) records the percentage of instances in the experiment that were feasible for each β and loss function. Figures C.3(a)-(c) record the average optimal path reliability, optimal path length, and optimal path CVaR for the instances that solved to optimality. The results all test networks are recorded in Appendix C.

CHAPTER 5

Conclusion and Future Work

This thesis formulates CVaR-based models for the SSPP-PAF. The literature contains models for the SSPP-PAF using robust optimization and expected value objectives in stochastic programming (Chapter 2). There has also been some work attempting to trade off robustness and path length (see Ahuja (1988)). Although CVaR has recently been used for supply-chain optimization (Madadi et al., 2010), to our knowledge this is the first attempt to use CVaR for a SSPP. The major contributions of this thesis are formulating the CVARSPP and developing three ways of modeling loss.

The results found in Chapter 3 develop the loss functions. The three loss functions measure whether the path breaks, the number of gaps, and the number of arc failures with corresponding physical interpretations. Distributions for the each loss function were found for networks with uniform arc failure probabilities. The distribution of the path failure loss function was identified for any path with probabilistic arc failures. Moreover, an analytical characterization of the shape of $\text{CVaR}(\mathbf{x}, \beta)$ in terms of β was derived. Finally, it was shown that, for each possible scenario the arc failure loss function is always at least as large as the detours loss function, which is in turn always at least as large as the path reliability loss function.

The lower edge of the level sets of the the fully-enumerated test network exp0002 agreed with the values calculated analytically. The computational experiments suggest that the path reliability loss function be used for networks with more reli-

able paths and the tail threshold β should be set lower than when using the other loss functions. In general, the CVaR-based formulations find more reliable paths than when using expected value.

This thesis has focused on modeling. Future research is needed to investigate computational issues. Preliminary experiments have indicated that scaling is likely to prove challenging. A decomposition approach such as the one proposed by Künzi-Bay and Mayer (2006) may be needed to handle larger instances. Another scaling issue is the number of scenarios. Even the small networks in this study showed the effects of sampling. As the network grows, the number of scenarios needed to approximate the actual loss distribution increases rapidly. Some investigation into a better sampling scheme might also help with scaling.

Another area for further investigation lies in extending single-stage CVaR-based formulations to two-stage recourse models of the SSPP-PAF. Finally, more investigation into the behavior of the CVaR-based formulation over a wider and more representative collection of networks is needed. It would be particularly interesting to test the models against some real-world networks, which include their own patterns of arc failures. Relaxing the assumption of independent arc failures and extending the analysis of the loss functions is another avenue to explore.

BIBLIOGRAPHY

- Acerbi, C. and Tasche, D. (2002). On the coherence of expected shortfall. *Journal of Banking & Finance*, 26(7):1487–1503.
- Ahuja, R. (1988). Minimum cost-reliability ratio path problem. *Computers & Operations Research*, 15(1):83–89.
- Ahuja, R. K., Magnanti, T. L., and Orlin, J. B. (1993). *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1 edition.
- Andreatta, G. and Romeo, L. (1988). Stochastic shortest paths with recourse. *Networks*, 18(3):193–204.
- Artzner, P., Delbaen, F., Eber, J.-M., and Heath, D. (1999). Coherent Measures of Risk. *Mathematical Finance*, 9(3):203–228.
- Ball, M. O., Magnanti, T. L., Monma, C. L., and Nemhauser, G. L., editors (1995a). *Network Models*, volume 7 of *Handbooks in Operations Research and Management Science*. Elsevier Science.
- Ball, M. O., Monma, C. L., and Magnanti, T. L., editors (1995b). *Network Routing*, volume 8 of *Handbooks in Operations Research and Management Science*. Elsevier Science Pub Co.
- Bayer, M., Kirtland, J., de Menten, G., Trier, M., Jenvey, P., Aasma, A., Johnston, P., Ellis, J., and Others (2011). SQLAlchemy 0.7.1. software.
- Bellman, R. E. (1958). On a Routing Problem. *Quarterly of Applied Mathematics*, 16:87–90.

- Bertsimas, D., Brown, D. B., and Caramanis, C. (2008). Theory and applications of robust optimization.
- Canada, J. R., Sullivan, W. G., Kulonda, D. J., and White, J. A. (2004). *Capital Investment Analysis for Engineering and Management*. Prentice Hall, 3 edition.
- Croucher, J. S. (1978). A note on the stochastic shortest-route problem. *Naval Research Logistics*, 25(4):729–732.
- Dantzig, G. B. (1957). Discrete-Variable Extremum Problems. *Operations Research*, 5(2).
- Deo, N. and Pang, C.-Y. (1984). Shortest-path algorithms: Taxonomy and annotation. *Networks*, 14(2):275–323.
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik*, 1(1):269–271.
- Dreyfus, S. E. (1969). An Appraisal of Some Shortest-Path Algorithms. *Operations Research*, 17(3).
- Erdős, P. and Rényi, A. (1959). On random graphs, I. *Publicationes Mathematicae (Debrecen)*, 6:290–297.
- Floyd, R. W. (1962). Algorithm 97: Shortest path. *Commun. ACM*, 5(6):345+.
- Gu, Z., Rothberg, E., and Bixby, R. (2010). Gurobi 4.0.2. software.
- Hagberg, A. A., Schult, D. A., and Swart, P. J. (2008). Exploring network structure, dynamics, and function using NetworkX. In *Proceedings of the 7th Python in Science Conference (SciPy2008)*, pages 11–15, Pasadena, CA USA.
- Hunter, J. D. (2007). Matplotlib: A 2D Graphics Environment. *Computing in Science and Engineering*, 9(3):90–95.

- Jaillet, P. (1992). Shortest path problems with node failures. *Networks*, 22(6):589–605.
- Jones, E., Oliphant, T., Peterson, P., and Others (2001). SciPy: Open source scientific tools for Python. software.
- Kendall, K. E. and Kendall, J. E. (2008). *Systems analysis and design*. Pearson/Prentice Hall.
- Künzi-Bay, A. and Mayer, J. (2006). Computational aspects of minimizing conditional value-at-risk. *Computational Management Science*, 3(1):3–27.
- Madadi, A. R., Kurz, M. B., Taafe, K., Mason, S., Pohl, E., Root, S., and Sir, M. (2010). Managing Disruptions in Pharmaceutical Supply Chain Networks. In Johnson, A. and Miller, J., editors, *Industrial Engineering Research Conference*.
- Oliphant, T. E. (2006). *Guide to Numpy*. Tregol Publishing, Provo, UT.
- Papadimitriou, C. (1991). Shortest paths without a map. *Theoretical Computer Science*, 84(1):127–150.
- Pollack, M. and Wiebenson, W. (1960). Solutions of the Shortest-Route Problem-A Review. *Operations Research*, 8(2).
- Rockafellar, R. and Uryasev, S. (2002). Conditional value-at-risk for general loss distributions. *Journal of Banking & Finance*, 26(7):1443–1471.
- Rockafellar, R. T. and Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of Risk*, 2:21–41.
- Sen, S. and Higle, J. L. (1999). An Introductory Tutorial on Stochastic Linear Programming Models. *Interfaces*, 29(2).

Van Rossum, G. (2003). *The Python Language Reference Manual*. Network Theory Ltd.

Yu, G. and Yang, J. (1998). On the Robust Shortest Path Problem. *Computers & Operations Research*, 25(6):457–468.

APPENDIX A

Optimization Models

A.1 Parameters

These parameters are used in all three optimization models.

- Graph $G = (N, A)$ is a directed network.
- $S \subseteq \Omega$ is the set of scenarios in the model, indexed by s . S may be a random sample from Ω or the whole sample space if Ω is small.
- c_{ij} is the cost of arc $(ij) \in A$.
- \mathbf{y}^s is the realization of the random vector \mathbf{Y} corresponding to scenario $s \in S$.
- p^s is the probability of scenario s , normalized so that $\sum_{s=1}^S p^s = 1$.
- C is the maximum acceptable CVaR loss.
- β is the percentile for the upper tail (e. g. 0.50, 0.95, 0.99).

A.2 Decision Variables

- x_{ij} indicates which arcs are chosen for the shortest path. $x_{ij} = 1$ if arc (ij) is chosen, and 0 otherwise.
- $\zeta \in \mathbb{R}$ is $\text{VaR}(\mathbf{x}, \beta)$ at optimality.
- z^s linearizes the excess loss $[L(\mathbf{x}, \mathbf{y}^s) - \zeta]^+$ in scenario $s \in S$.
- ℓ_i^s is the loss at node $i \in N$ in scenario $s \in S$ and linearizes the loss function.

A.3 Path Reliability Loss Function

$$L^r(\mathbf{x}, \mathbf{Y}) = \max_{(ij) \in A} x_{ij}(1 - Y_{ij}) \quad (\text{A.1})$$

Formulation A.3.1 (Path Reliability Loss Function).

$$\text{Minimize } \sum_{(ij) \in A} c_{ij} x_{ij} \quad (\text{A.2})$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (\text{A.3})$$

$$\zeta + \frac{1}{1 - \beta} \sum_{s \in S} p^s z^s \leq C \quad (\text{A.4})$$

$$z^s \geq x_{ij}(1 - y_{ij}^s) - \zeta \quad \forall s \in S, (ij) \in A \quad (\text{A.5})$$

$$\mathbf{x} \in \{0, 1\}^{|A|}; \quad z \in \mathbb{R}_+^{|S|}; \quad \zeta \in \mathbb{R} \quad (\text{A.6})$$

A.4 Arc Failure Loss Function

$$L^a(\mathbf{x}, \mathbf{Y}) = \sum_{(ij) \in A} x_{ij}(1 - Y_{ij}) \quad (\text{A.7})$$

Formulation A.4.1 (Arc Failure Loss Function).

$$\text{Minimize } \sum_{(ij) \in A} c_{ij}x_{ij} \quad (\text{A.8})$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (\text{A.9})$$

$$\zeta + \frac{1}{1 - \beta} \sum_{s \in S} p^s z^s \leq C \quad (\text{A.10})$$

$$z^s \geq \sum_{(ij) \in A} x_{ij}(1 - y_{ij}^s) - \zeta \quad \forall s \in S \quad (\text{A.11})$$

$$\mathbf{x} \in \{0, 1\}^{|A|}; \quad \mathbf{z} \in \mathbb{R}_+^{|S|}; \quad \zeta \in \mathbb{R} \quad (\text{A.12})$$

A.5 Detours Loss Function

$$L^d(\mathbf{x}, \mathbf{Y}) = \sum_{i \in N} \left| \sum_{j \in N^-(i)} x_{ji} Y_{ji} - \sum_{j \in N^+(i)} x_{ij} Y_{ij} - b_i \right| \quad (\text{A.13})$$

Formulation A.5.1 (Detours Loss Function).

$$\text{Minimize } \sum_{(ij) \in A} c_{ij} x_{ij} \quad (\text{A.14})$$

subject to:

$$\sum_{j \in N^+(i)} x_{ij} - \sum_{j \in N^-(i)} x_{ji} = \begin{cases} 1 & \text{if } i = s \\ -1 & \text{if } i = t \\ 0 & \text{if } i \in N \setminus \{s, t\} \end{cases} \quad (\text{A.15})$$

$$\zeta + \frac{1}{1 - \beta} \sum_{s \in S} p^s z^s \leq C \quad (\text{A.16})$$

$$z^s \geq \sum_{i \in N} \ell_i^s - \zeta \quad \forall s \in S \quad (\text{A.17})$$

$$\ell_i^s \geq \sum_{j \in N^-(i)} x_{ji} y_{ji}^s - \sum_{j \in N^+(i)} x_{ij} y_{ij}^s - b_i \quad \forall s \in S, i \in N \quad (\text{A.18})$$

$$\ell_i^s \geq b_i + \sum_{j \in N^+(i)} x_{ij} y_{ij}^s - \sum_{j \in N^-(i)} x_{ji} y_{ji}^s \quad \forall s \in S, i \in N \quad (\text{A.19})$$

$$\mathbf{x} \in \{0, 1\}^{|A|}; \quad \mathbf{z} \in \mathbb{R}_+^{|S|}; \quad \boldsymbol{\ell} \in \mathbb{R}^{|S|} \times \mathbb{R}^{|N|}; \quad \zeta \in \mathbb{R} \quad (\text{A.20})$$

APPENDIX B

Schema of Experiment Database

This appendix describes the database used in the computational experiments. The database contains all data generated during the experiments. The database schema is diagrammed in Figure B.2 using crow's foot notation. Each table in the database and its fields are described below. The database was normalized to the Backus-Naur Third Normal Form (Kendall and Kendall, 2008) before being selectively denormalized for ease of use.

B.1 Database Tables

Graph The table for the directed networks. This does not contain arc weights, arc failure probabilities, or scenarios.

name The name of the directed network.

descr A sentence or two describing the network to a human reader.

n The number of nodes.

a The number of arcs.

source The source node of the network.

sink The sink node of the network.

density The density of the network

obj The Python object containing the arc list.

Weights The table for arc weights (c_{ij}).

name A short name for the arc weights.

descr A sentence or two describing the arc weights or their generation method to a human reader.

obj The Python object containing the list of arc weights.

length The number of weights in the list of arc weights.

graph_id Foreign key to Graphs table.

AFPs The table for arc failure probabilities (p_{ij}).

name A short name for the arc failure probabilities.

descr A sentence or two describing the arc failure probabilities and their generation method to a human reader.

obj The Python object containing the arc failure probabilities.

length The number of arc failure probabilities in the list.

graph_id Foreign key to Graphs table.

Scenarios The table for scenarios (\mathbf{y}^s and p^s for all $s \in S$)

name A short name for this set of scenarios.

descr A sentence or two describing the scenarios and their generation method to a human reader.

s The number of scenarios.

scen_array The Python object containing an $|A| \times s$ Numpy matrix of the scenarios. Each row is \mathbf{y}^s , the incidence vector of arc failures.

ps The Python object containing a Numpy vector of scenario probabilities. Each element corresponds to p^s .

empafp The Python object containing a Numpy vector of $|A|$ entries. Each entry is the percent of scenarios in which that arc fails.

afp_id Foreign key to AFPs table.

TestInstance The table for instances. This is a full network scenarios and with weights and failure probabilities attached to each arc.

name A short name for this instance.

graph_id Foreign key to Graphs table.

wts_id Foreign key to Weights table.

scen_id Foreign key to Scenarios table.

Path The table for information about paths of interest through a network.

rel The a-priori reliability (i.e. $\prod_{(ij) \in A(x)} (1 - p_{ij})$).

cost The cost of the path (i.e. $\sum_{(ij) \in A(x)} c_{ij}$).

arcs Number of arcs in this path.

type Classification — DSP, MRP, MDP, or an optimal solution to some CVAR-SPP instance.

inst_id Foreign key to the TestInstance table.

sol_id Optional: Foreign key to the Solution table.

PathArcs The table that stores the arcs in a path.

head The head of an arc in a path.

tail The tail of an arc in a path.

path_id Foreign key to the Path table indicating which path contains these arcs.

Exp The table of summary information about a computational experiment. The information in this table is mostly intended as notes for a human reader.

name A short name for this set of experiment.

descr A summary of the experiment.

goal The objective of the experiment.

beta_method A summary of how the tail threshold probabilities β are to be generated.

c_method A summary of how the maximum acceptable losses C are to be generated.

loss_fns The loss functions involved in this experiment.

computer The machine used to run the experiment.

solver The program used to solve the CVARSPP models.

Param The table storing solver control parameters such as time limit.

filename The name of the file with the parameters for the solver to read.

folder The path to the file.

suffix The extension of the file indicates format.

plist The Python object containing the parameters, a duplicate of the file.

Run Each record represents a unique instance of the CVARSPP.

name A short name for this CVARSPP instance.

beta The tail threshold probability β .

C The Python object representing the arc failure probabilities (a BLOB in SQL terms).

loss_fn Which loss function to use; i.e., which specific model to choose (see Appendix A).

solved Flag indicating whether the run has been sent to the solver or not.

exp_id Foreign key to Exp table.

param_id Foreign key to the Param table.

inst_id Foreign key to TestInstance table.

OptInfo The table for information reported by the solver about an instance. If an optimal solution is found, it is stored in the Solution table.

optimal Did the instance solve to optimality?

grb_status The status code reported by the solver.

setup The time to set up each instance.

sol_time The solution time for each instance.

iter The number of simplex iterations performed for all the nodes explored.

nodes The number of branch-and-bound nodes explored.

best_bound If the solver ran out of time, the best bound found during optimization.

num_sol The number of integer feasible solutions during optimization.

obj The Python object containing all the information about running the optimization which was returned by the solver.

date Timestamp for the optimization run.

run_id Foreign key for the Run table.

Solution The table for information about the optimal solution to a run.

obj_cost The cost of the optimal path.

var The VaR for this instance.

cvar The CVaR for the instance, calculated C and the slack in the CVaR constraint.

cvar_slack The slack in the CVaR constraint.

dvs The Python object containing the values of all decision variables.

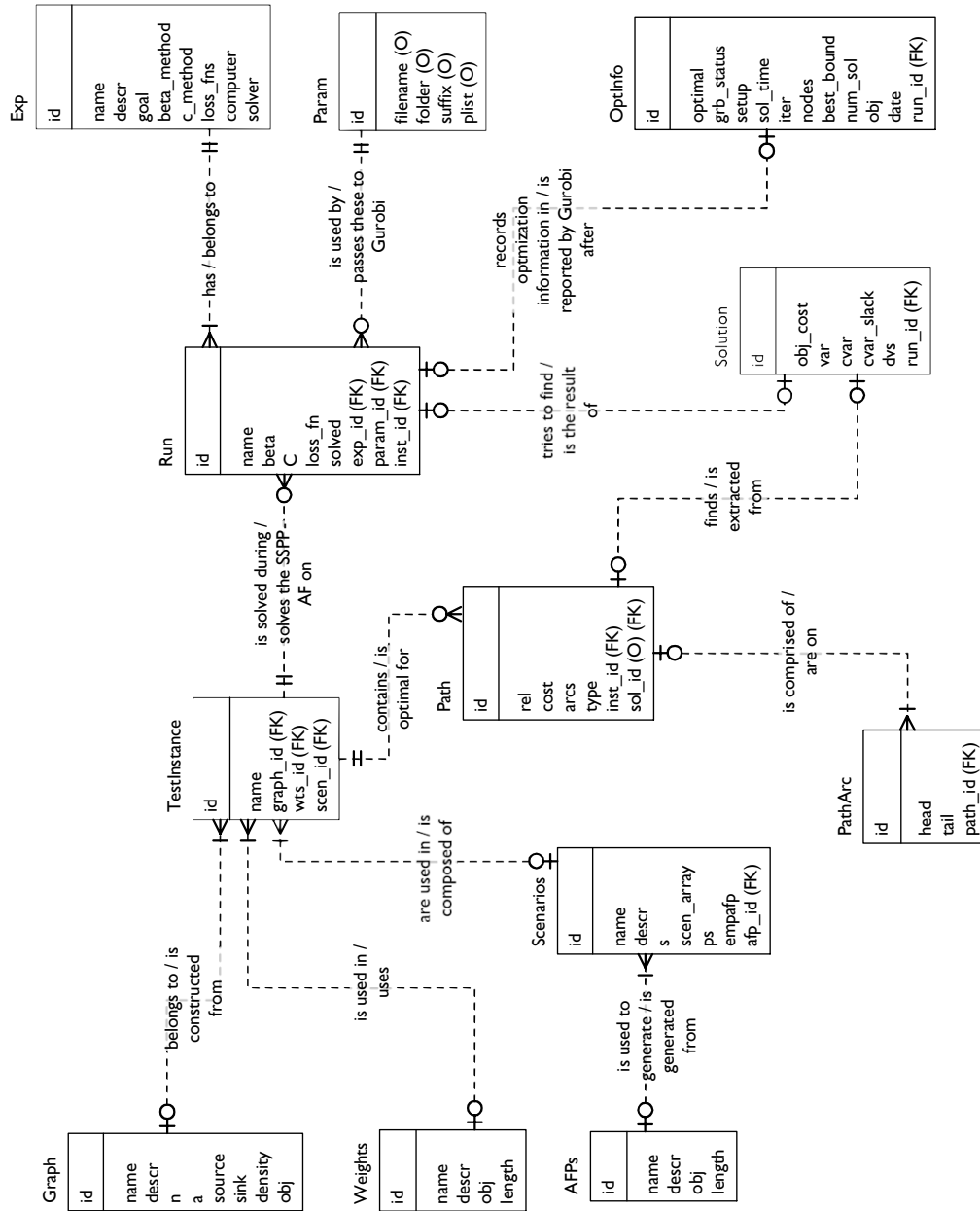
run_id Foreign key for the Run table.

B.2 Database Schema

Figure B.1: Legend: Crow's Foot Notation

Table Name	Relationship	Symbol
Primary Key		
Attribute	Zero or one-to-zero or one	+○-----○+
(FK): Foreign Key	Zero or one-to-many	+○----- <
Key	One-to-zero or many	H-----○<
(O): Optional	One-to-many	-H----- <

Figure B.2: Database Schema for Computational Experiments after Kendall and Kendall (2008)



APPENDIX C

Objective Function Contour Graphs by Loss Function

This appendix contains graphs of the objective function values for each test instance and loss function.

Figure C.1: exp0002: Level sets of the optimal path length as a function of β and C

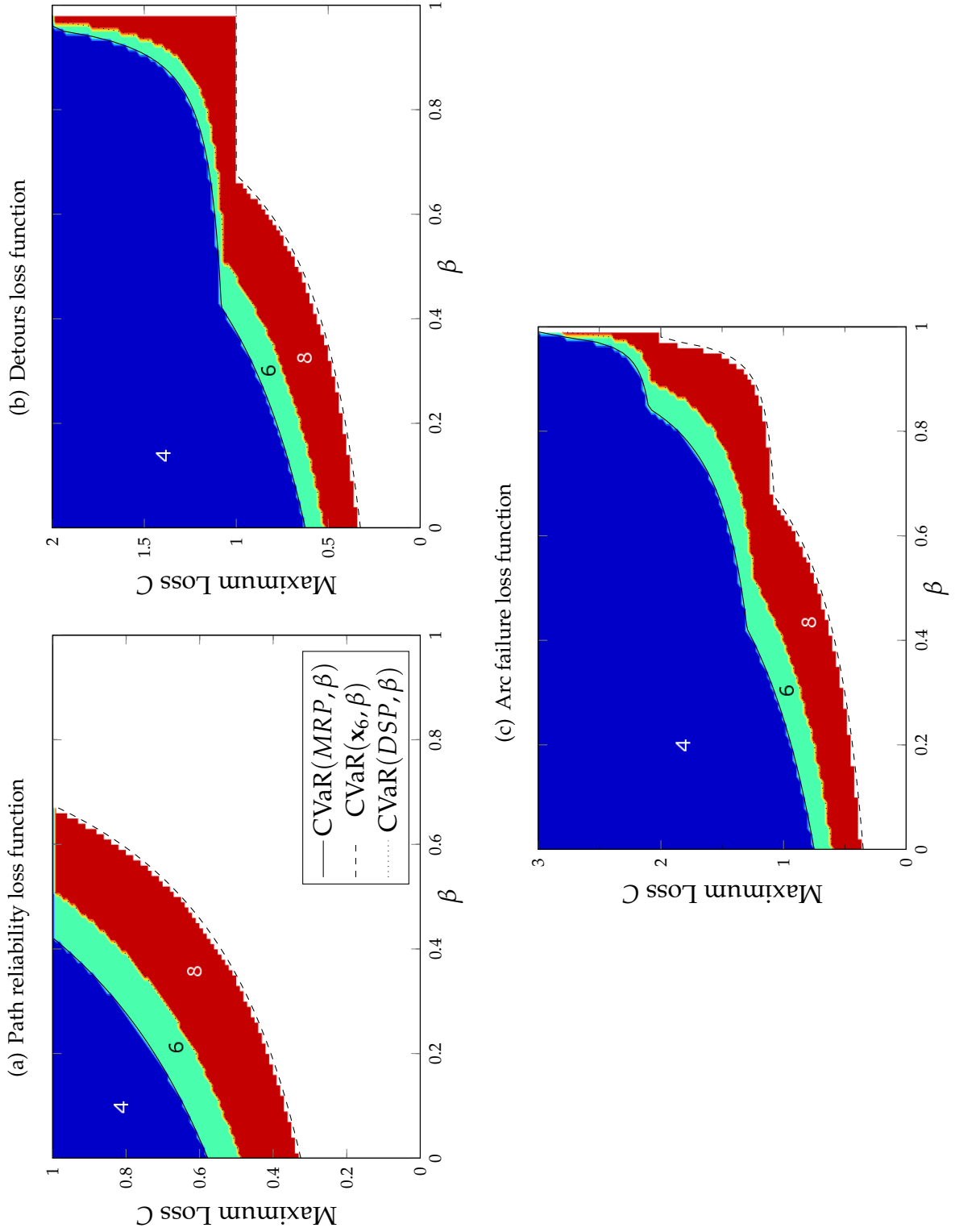


Figure C.2: exp0002: Change in properties of the average optimal path vs. β

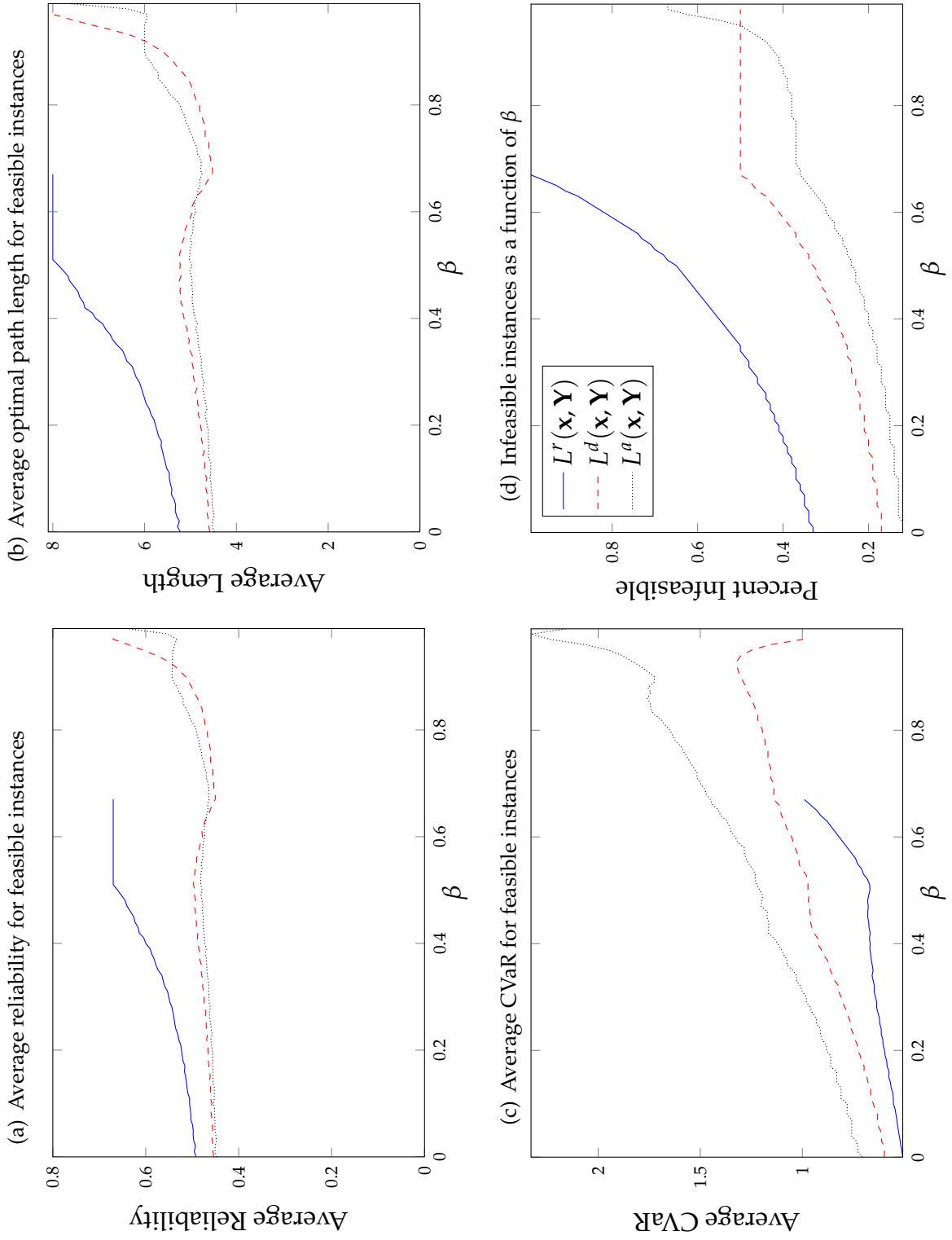


Figure C.3: branch02: Level sets of the optimal path length as a function of β and C

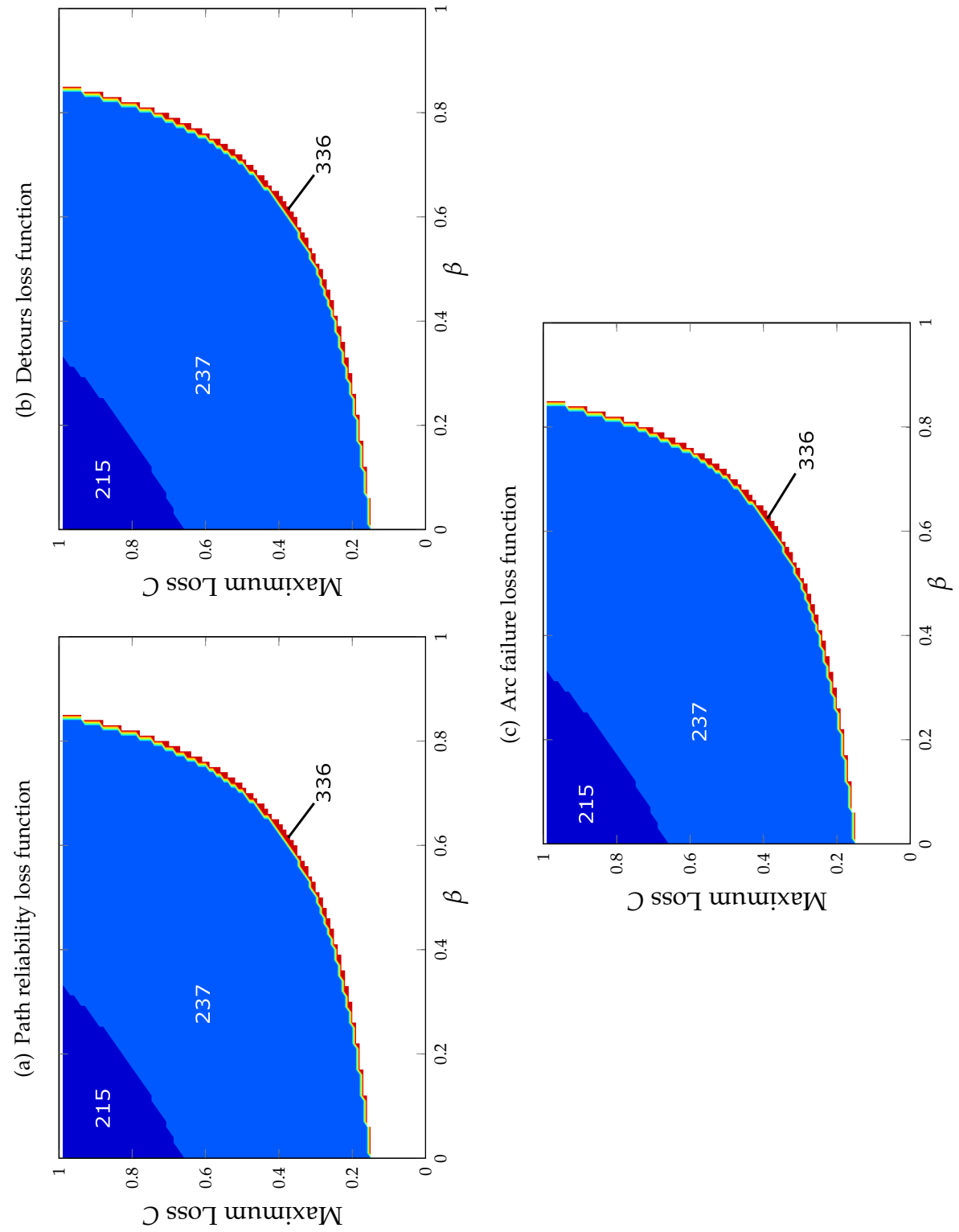


Figure C.4: branch02: Change in properties of the average optimal path vs. β

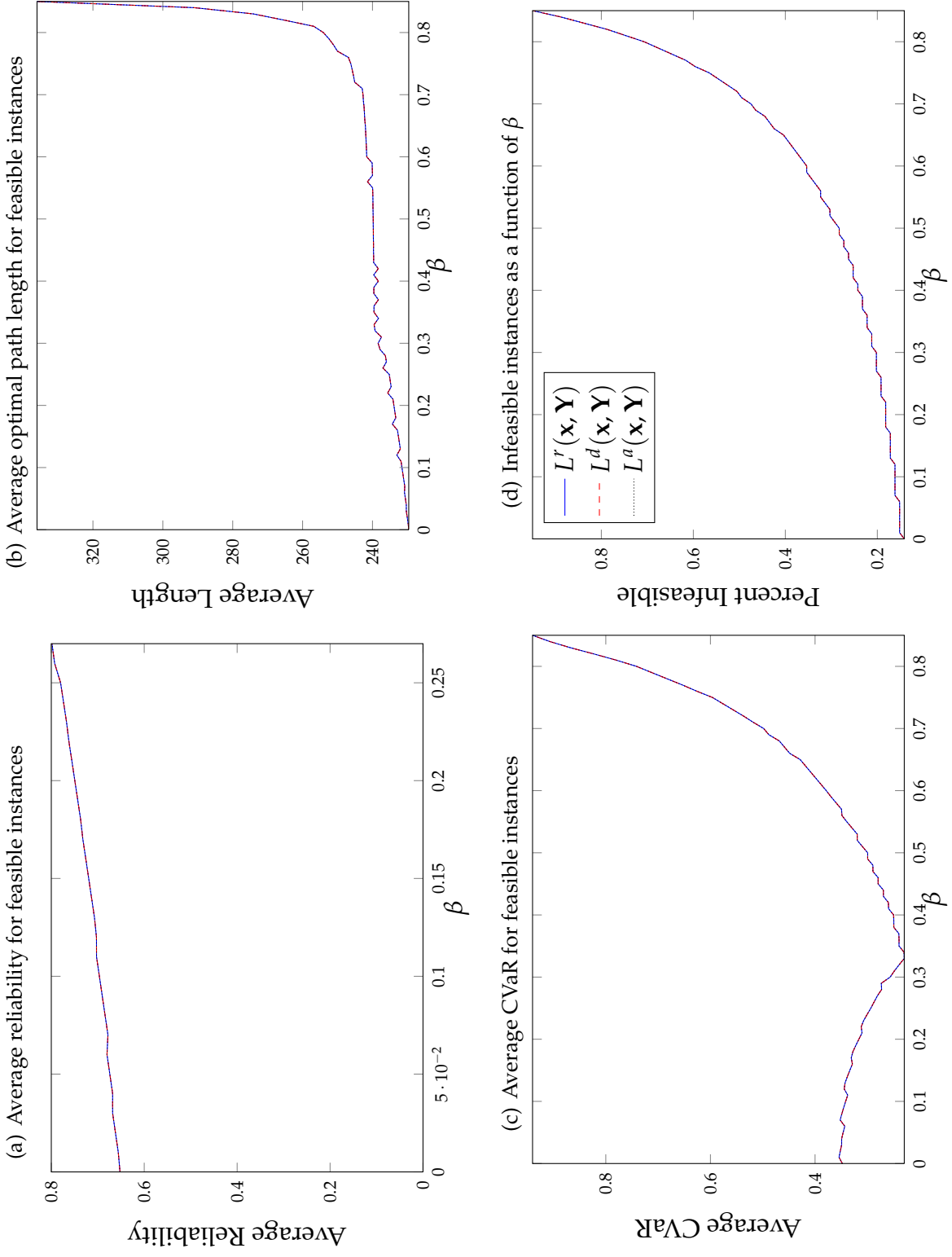


Figure C.5: branch03: Level sets of the optimal path length as a function of β and C

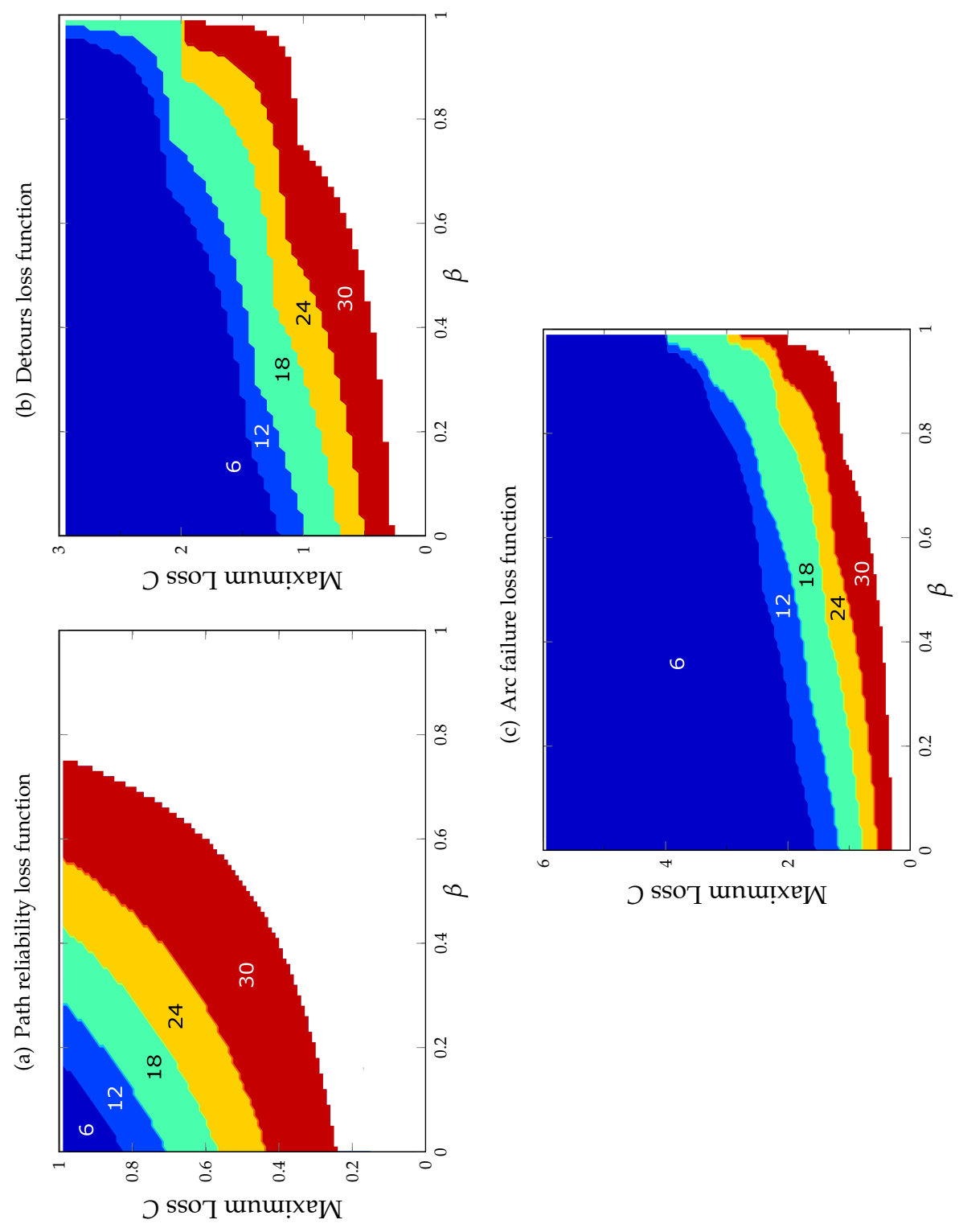


Figure C.6: branch03: Change in properties of the average optimal path vs. β

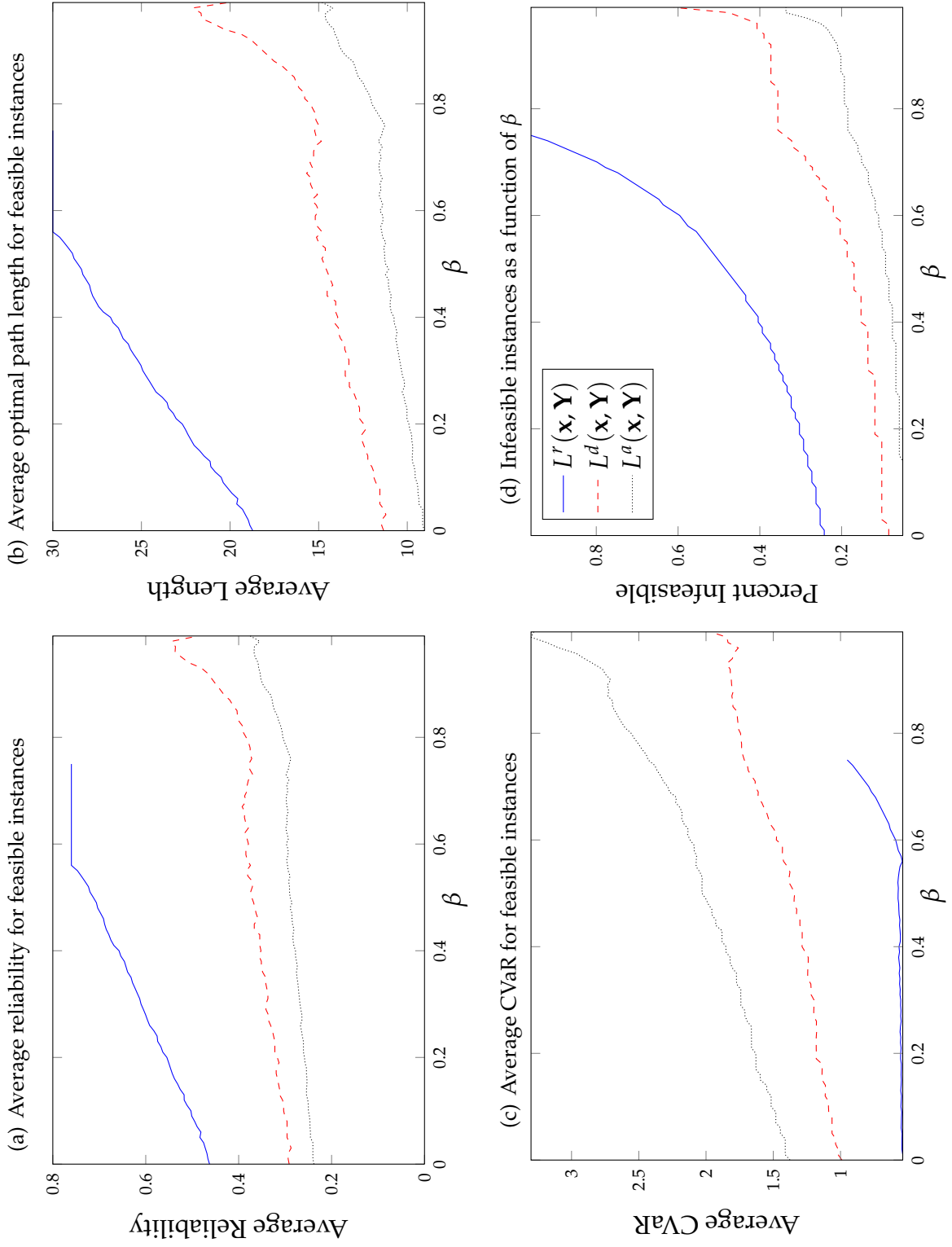


Figure C.7: branch04: Level sets of the optimal path length as a function of β and C

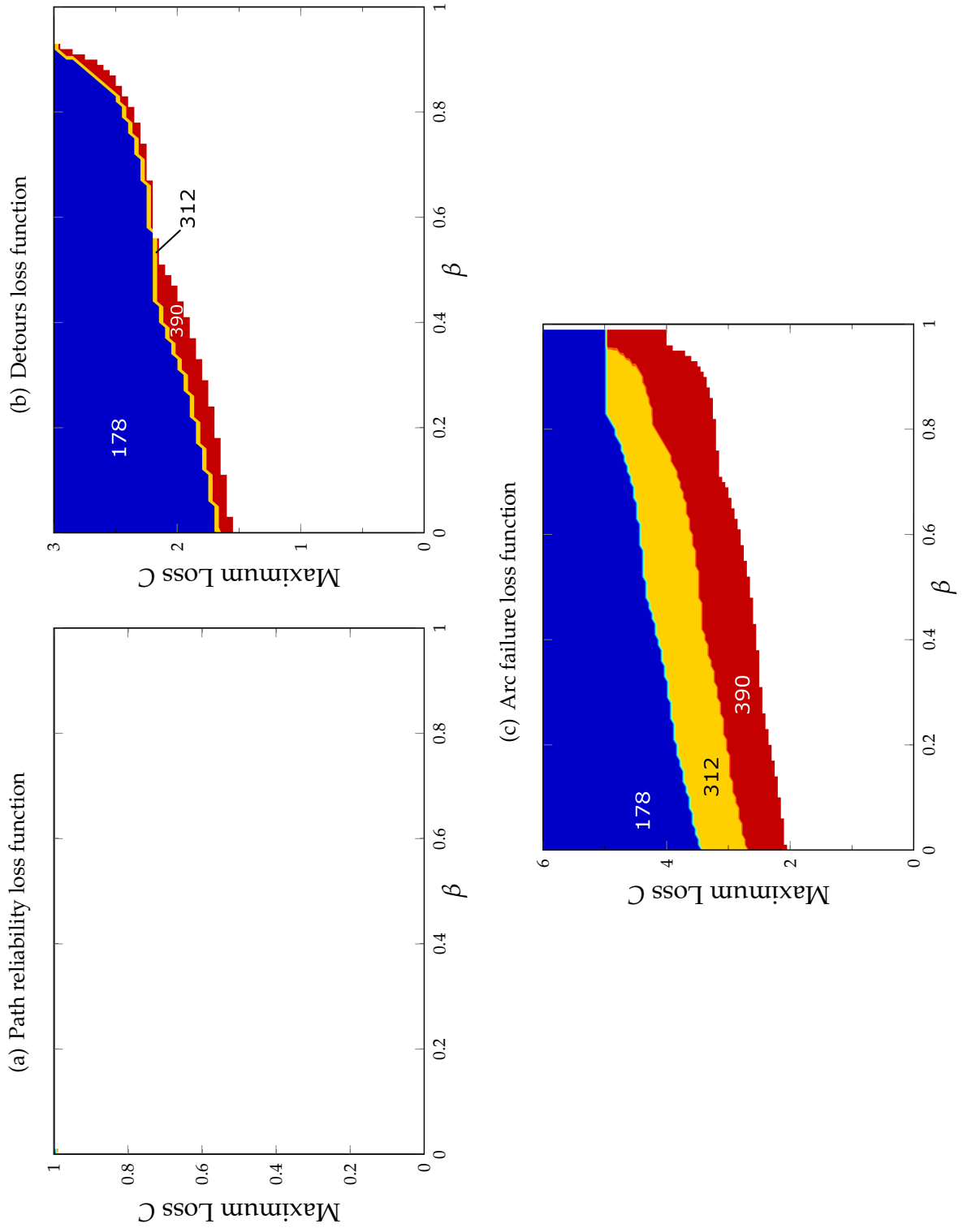


Figure C.8: branch04: Change in properties of the average optimal path vs. β

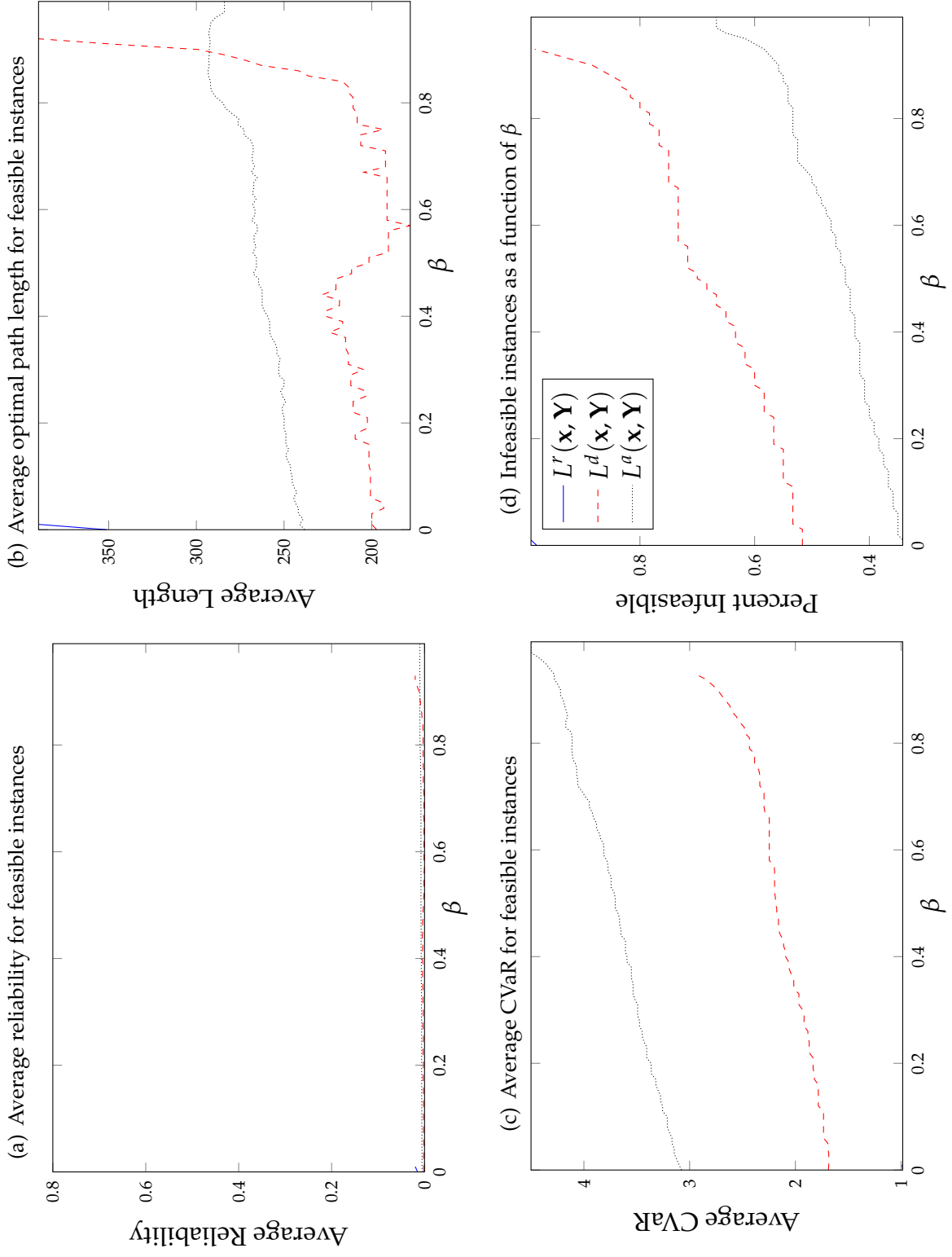


Figure C.9: gnp01: Level sets of the optimal path length as a function of β and C

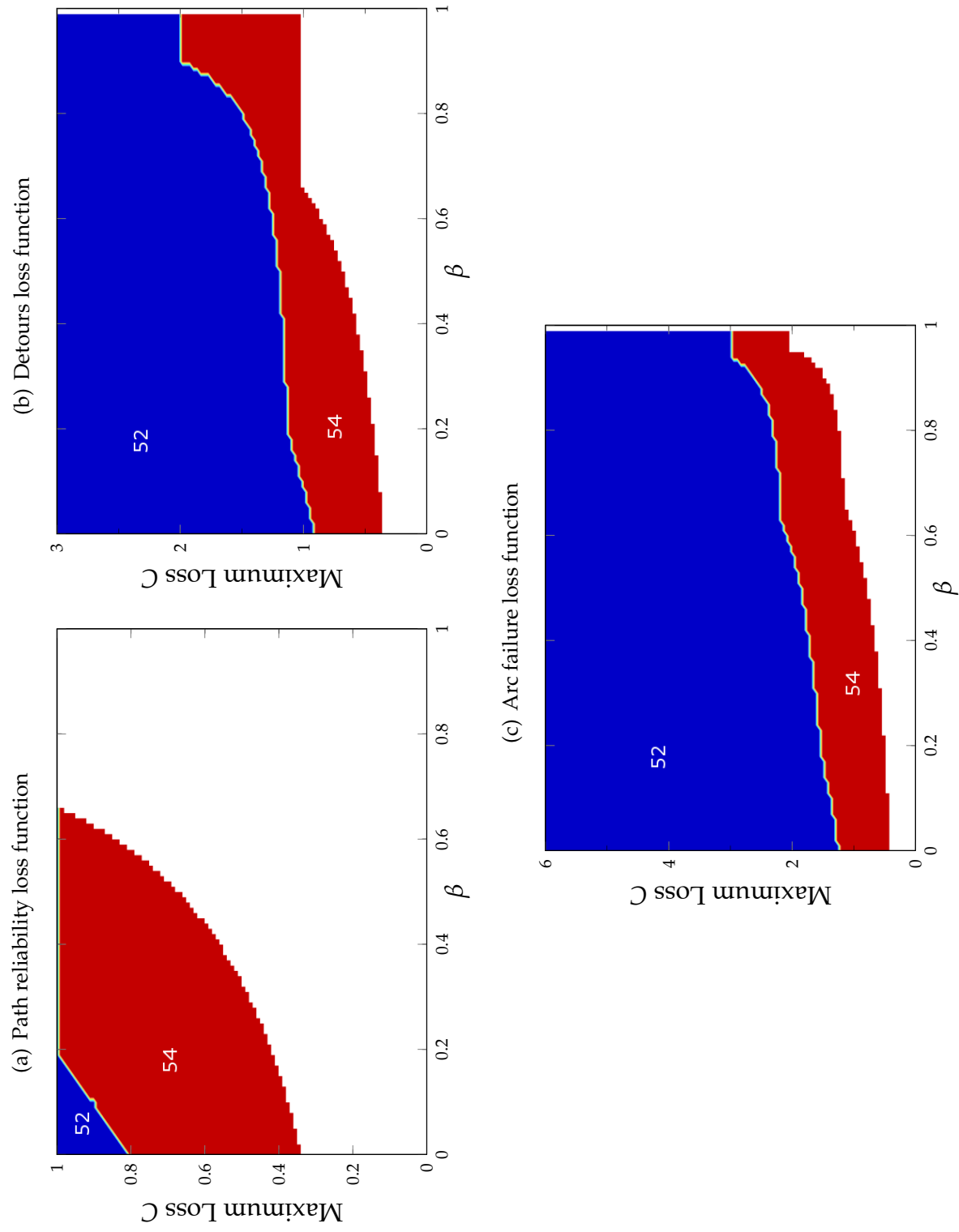


Figure C.10: gnp01: Change in properties of the average optimal path vs. β

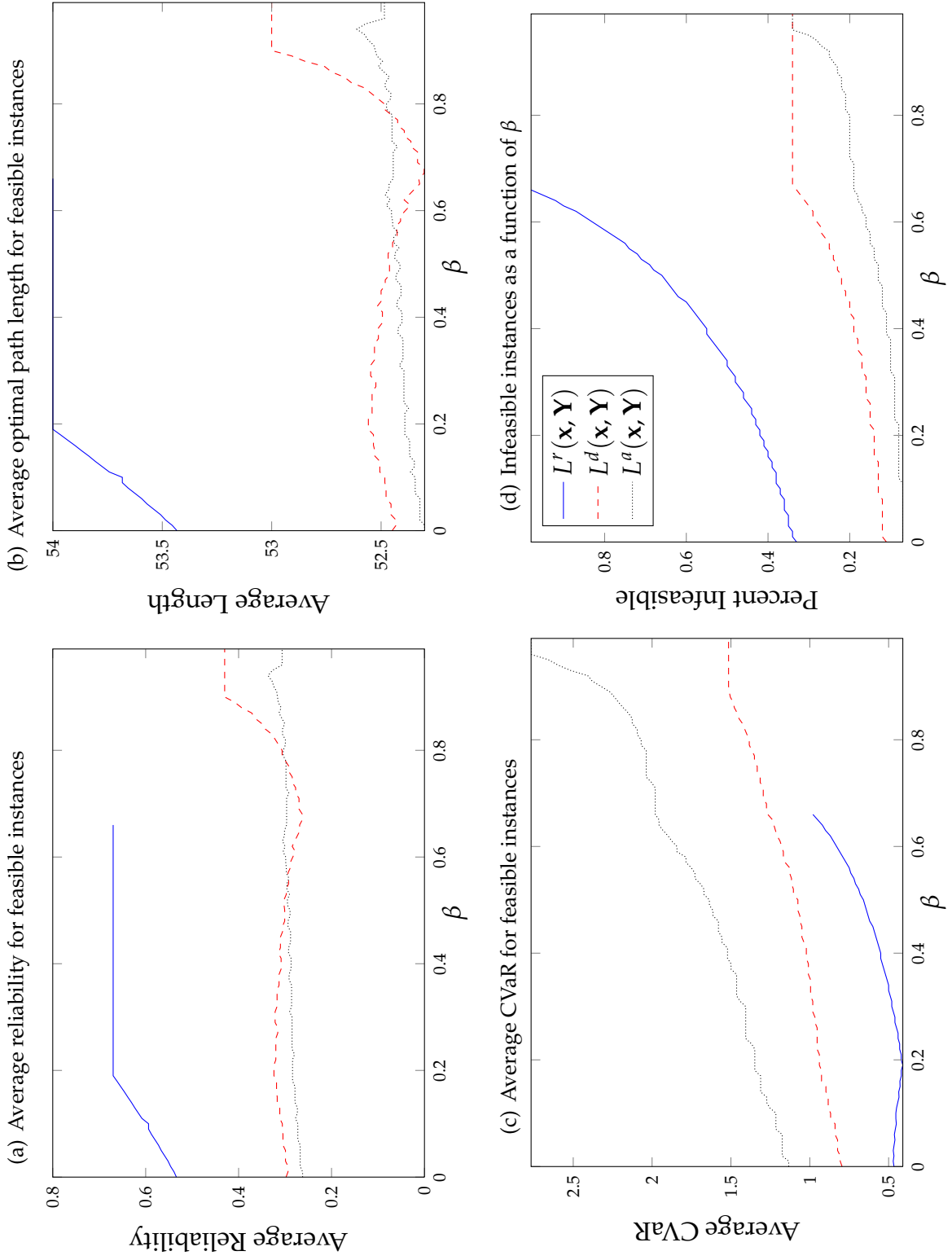


Figure C.11: gnp02: Level sets of the optimal path length as a function of β and C

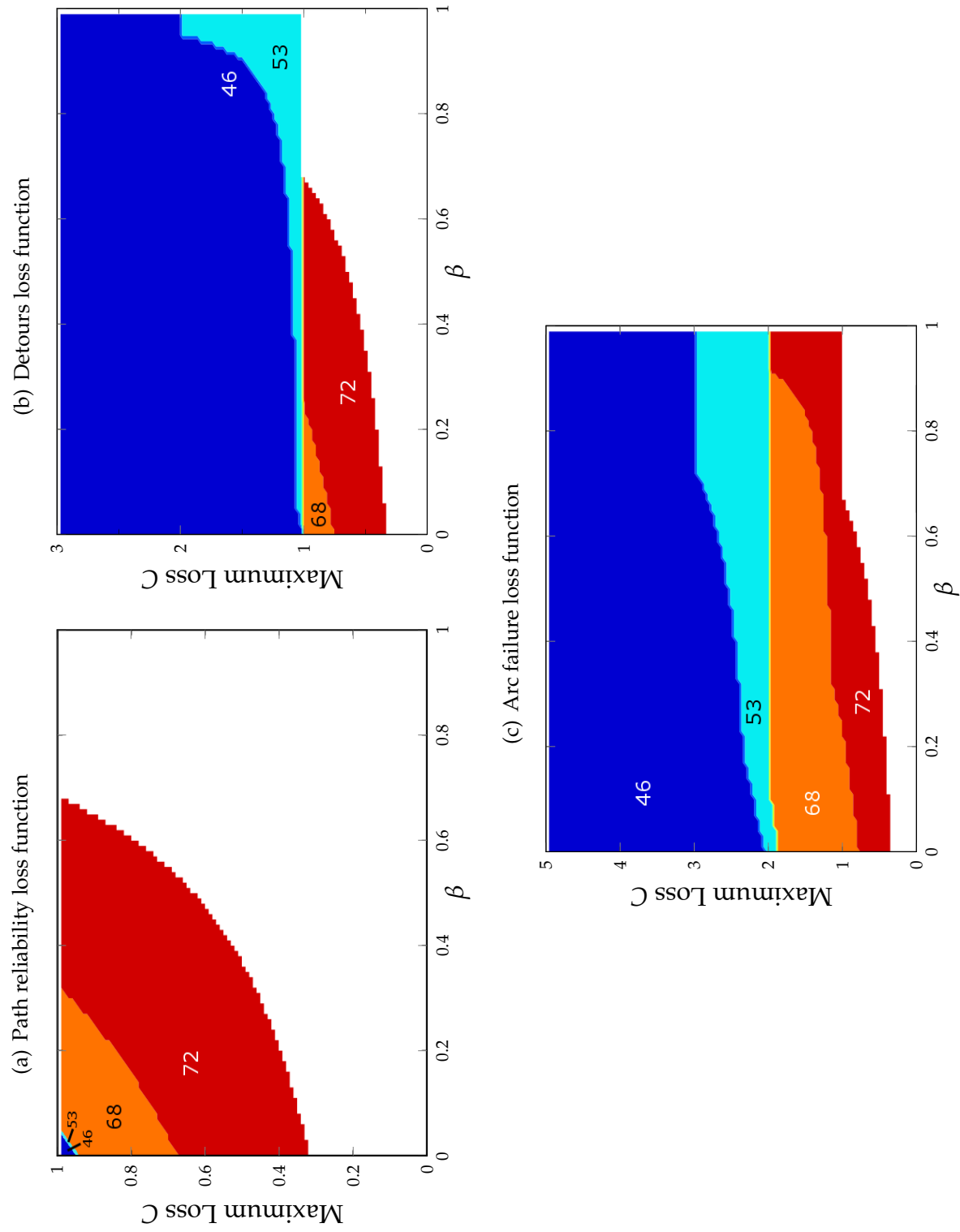


Figure C.12: gnp02: Change in properties of the average optimal path vs. β

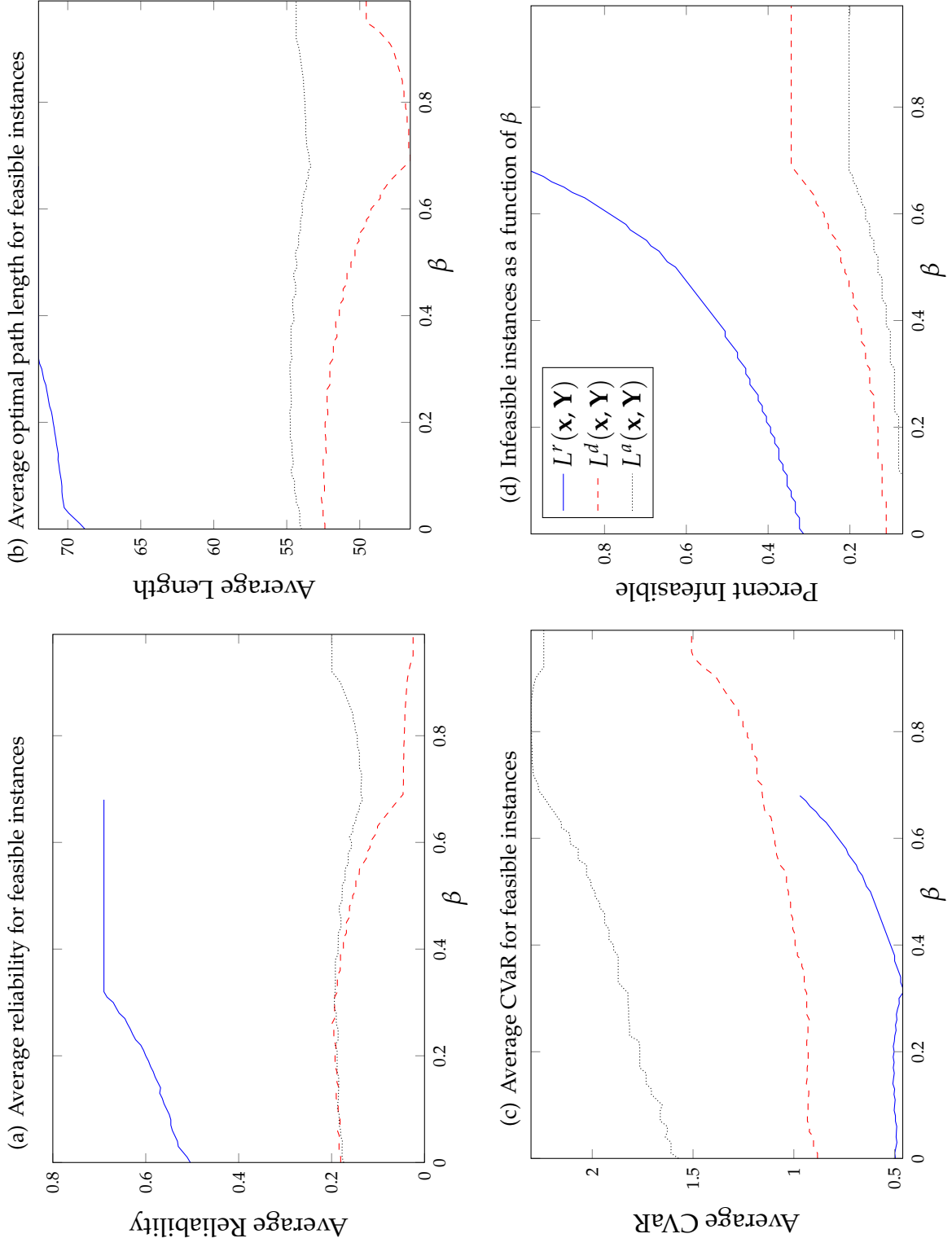


Figure C.13: gnp03: Level sets of the optimal path length as a function of β and C

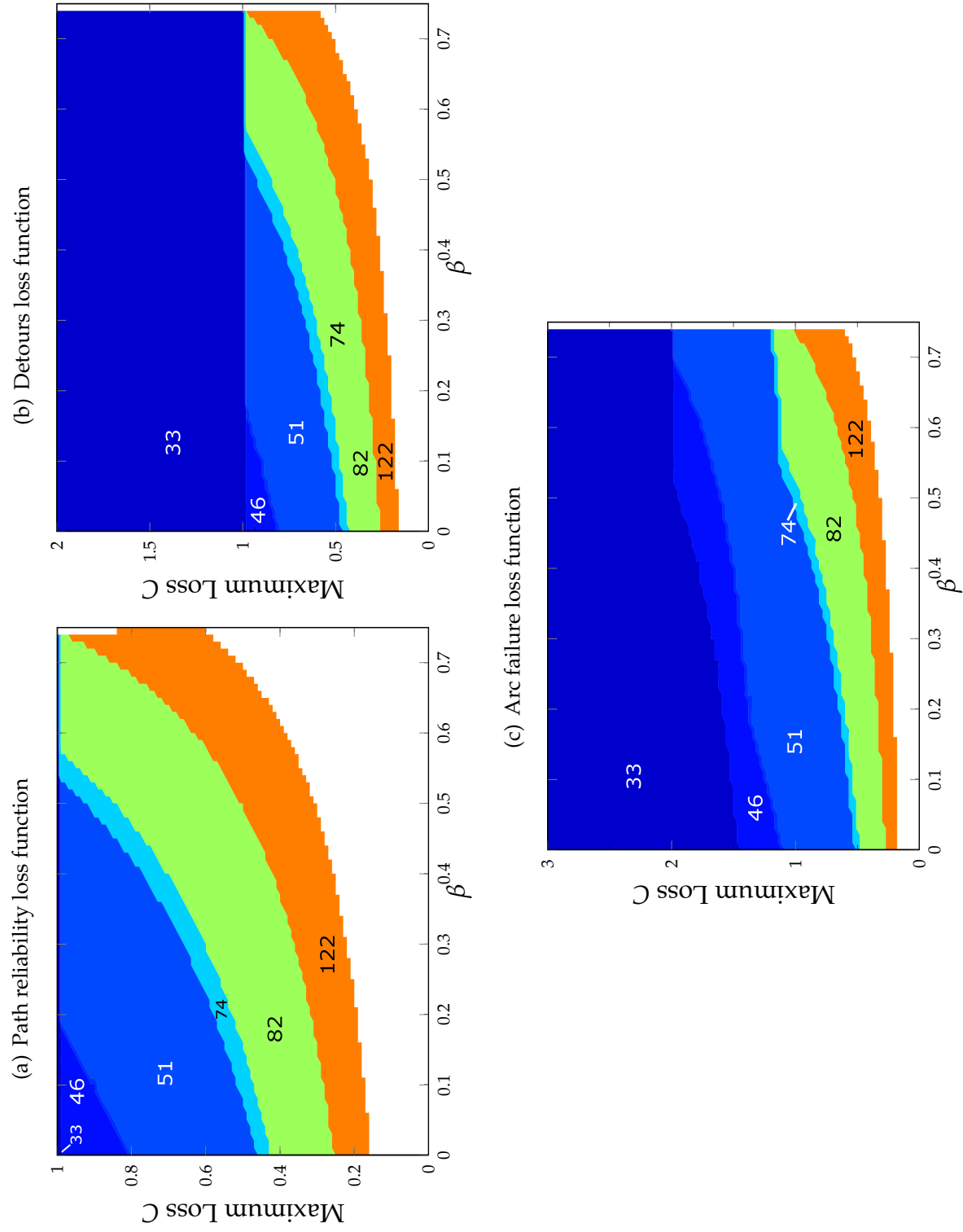
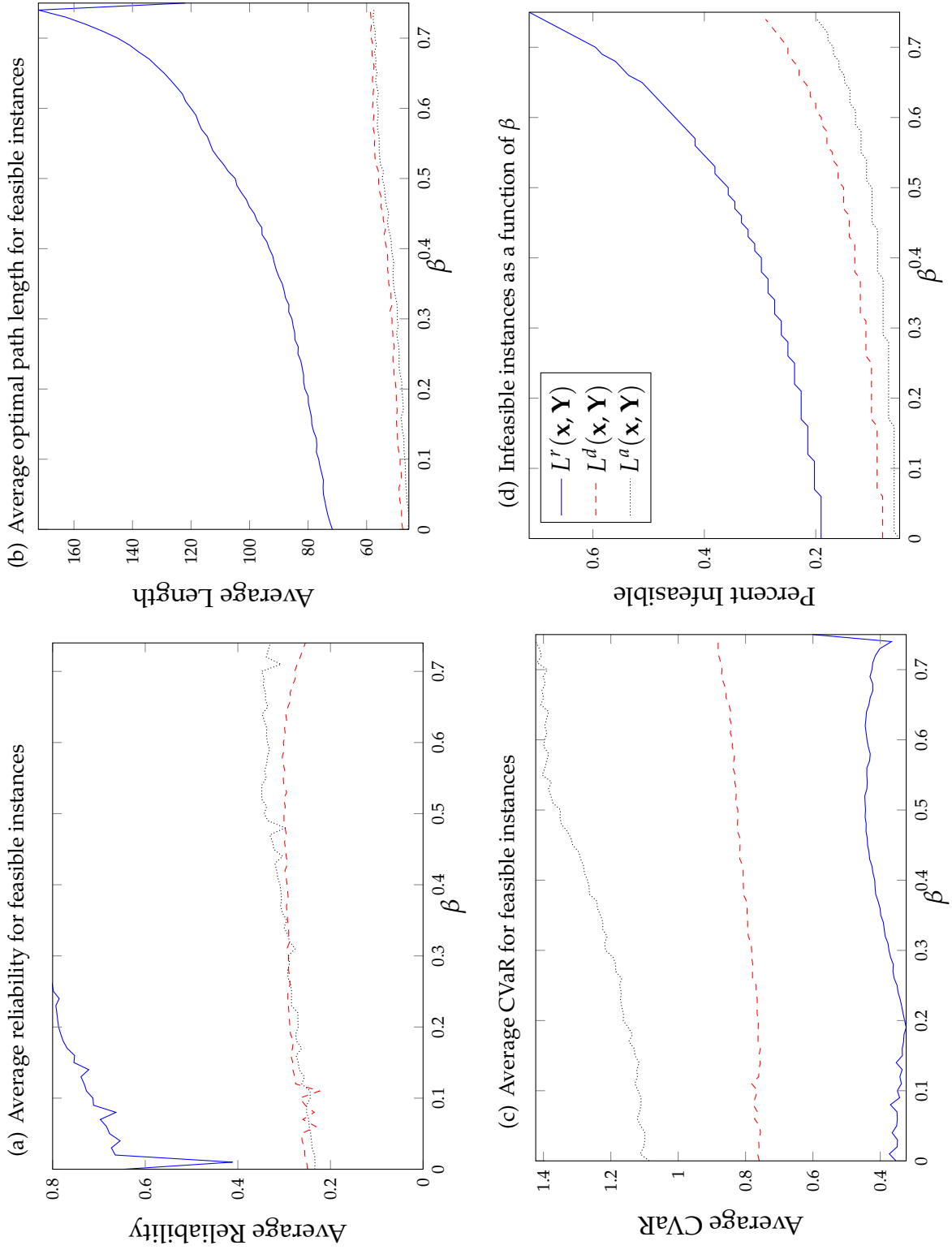


Figure C.14: gnp03: Change in properties of the average optimal path vs. β



APPENDIX D

Test Network Diagrams

D.1 Small Network: exp0002

The scenarios for this small network are completely enumerated. Arc failure probabilities are chosen to force the MDP to be the MRP and have the highest cost while the DSP is the least reliable path.

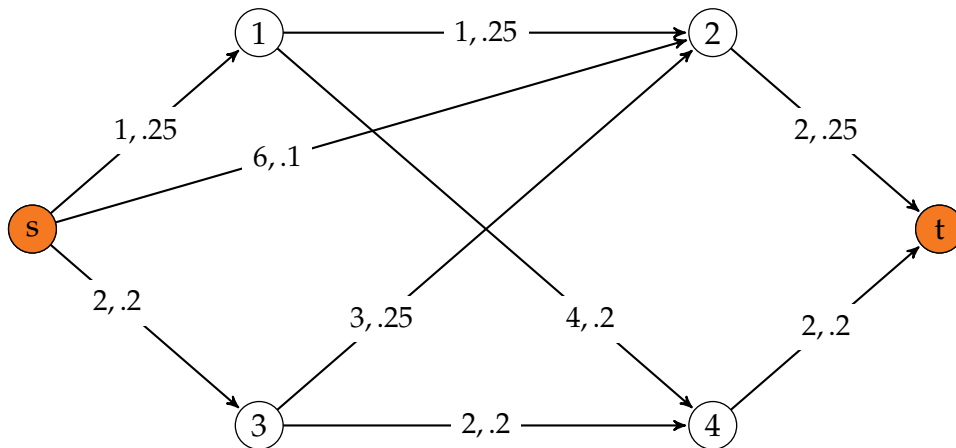


Figure D.1: Network exp0002

D.2 Branch Networks

The three following networks have five disjoint paths, or branches, of six arcs connecting the source s to sink t .

Figure D.2: Network branch02

The arc weights c_{ij} are drawn from $DU(1, 100)$. One arc is allowed to fail on each branch with probability p_{ij} drawn from $Unif(0, 1)$.

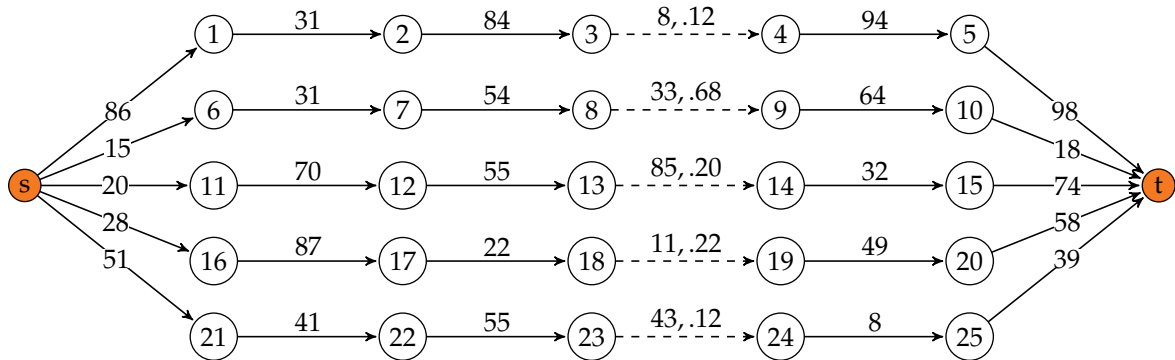


Figure D.3: Network branch03

The arc failure probabilities and arc costs are uniform across each branch. As the cost of the path increases from the DSP to the most expensive path, so does the reliability.

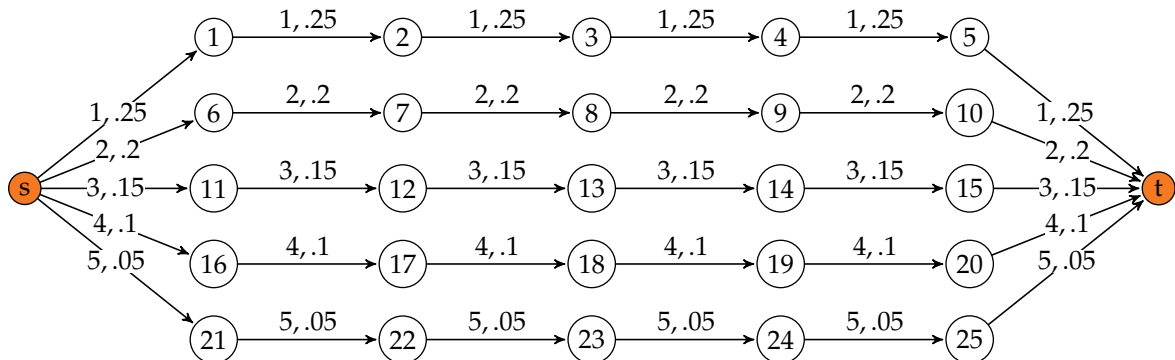
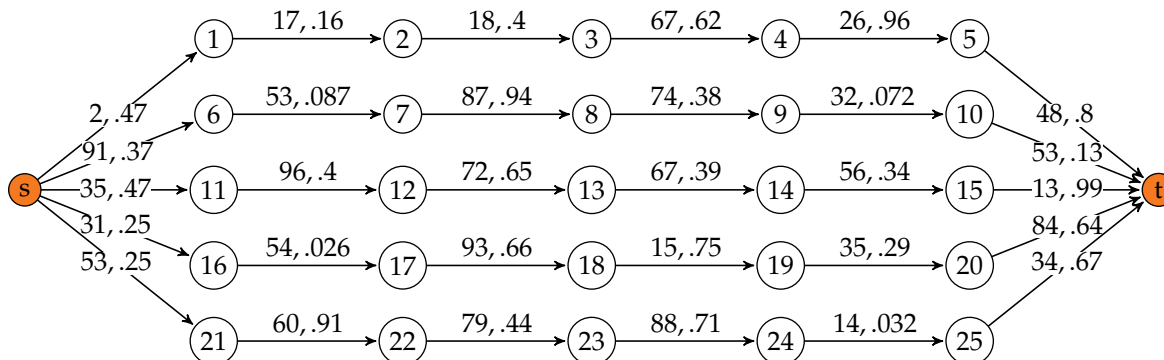


Figure D.4: Network branch04

The arc failure probabilities p_{ij} are drawn from $\text{Unif}(0, 1)$, and the arc weights c_{ij} are drawn from $\text{DU}(1, 100)$.



D.3 Uniform Random Networks

The final three networks are $\text{GNP}(n, p)$ random networks (Erdős and Rényi, 1959).

All three networks have $n = 25$ nodes. The arc failure probabilities p_{ij} are drawn from $\text{Unif}(0, 1)$, and arc weights c_{ij} from $\text{DU}(1, 100)$.

Figure D.5: Network gnp01 — $\text{GNP}(25, 0.1)$

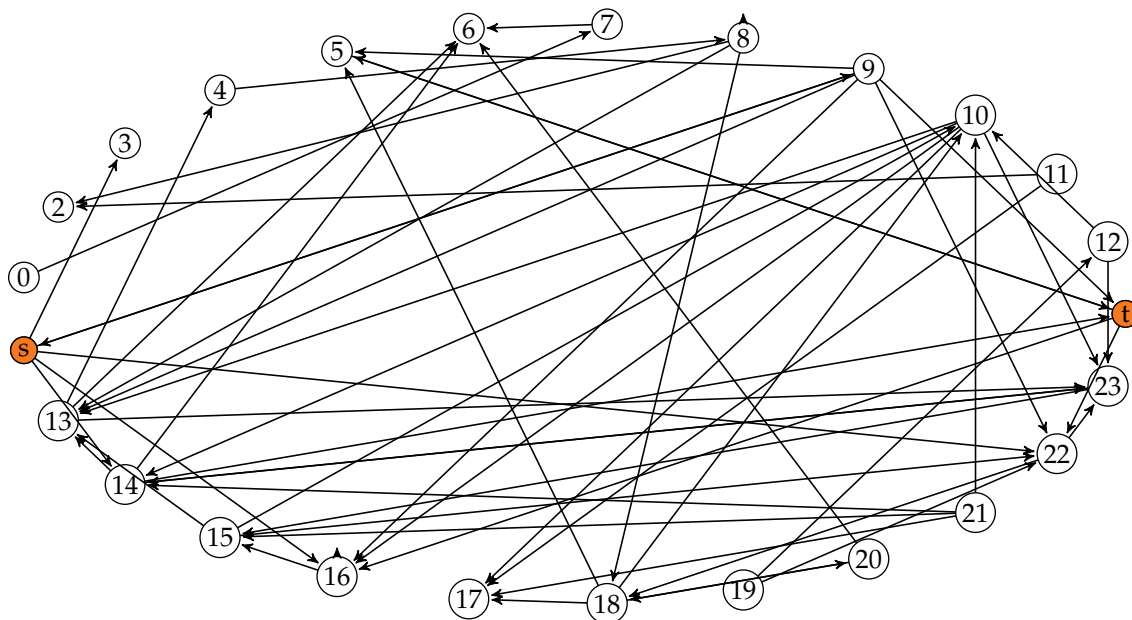


Figure D.6: Network gnp02 — $GNP(25, 0.25)$

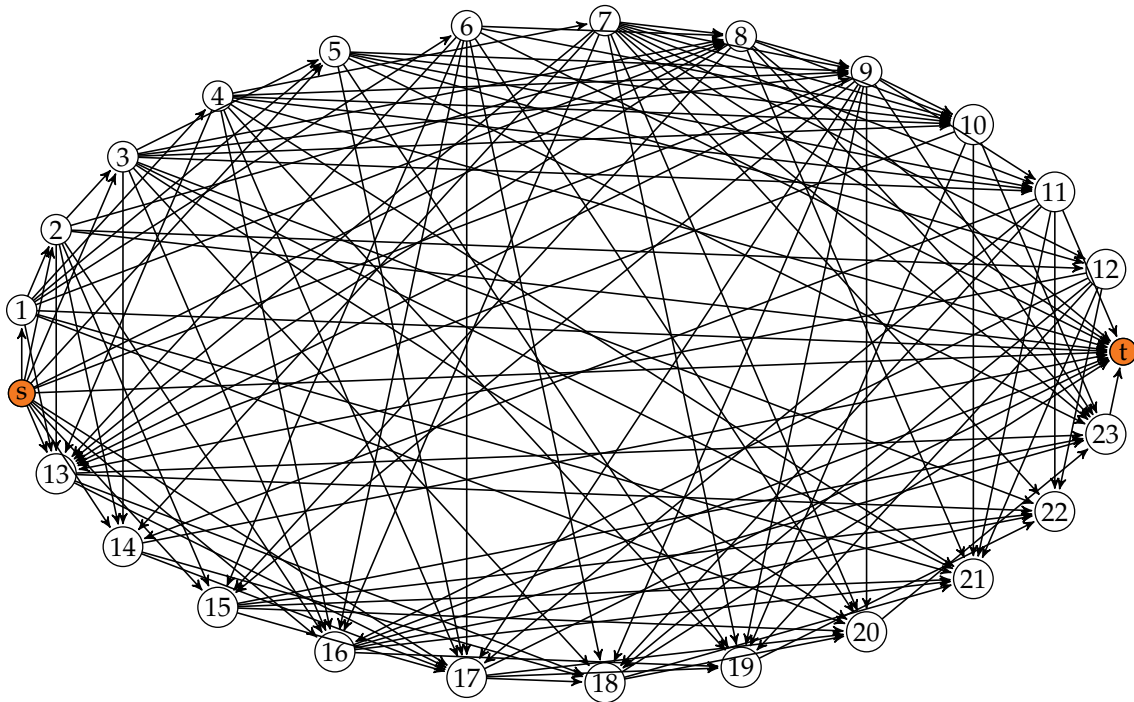
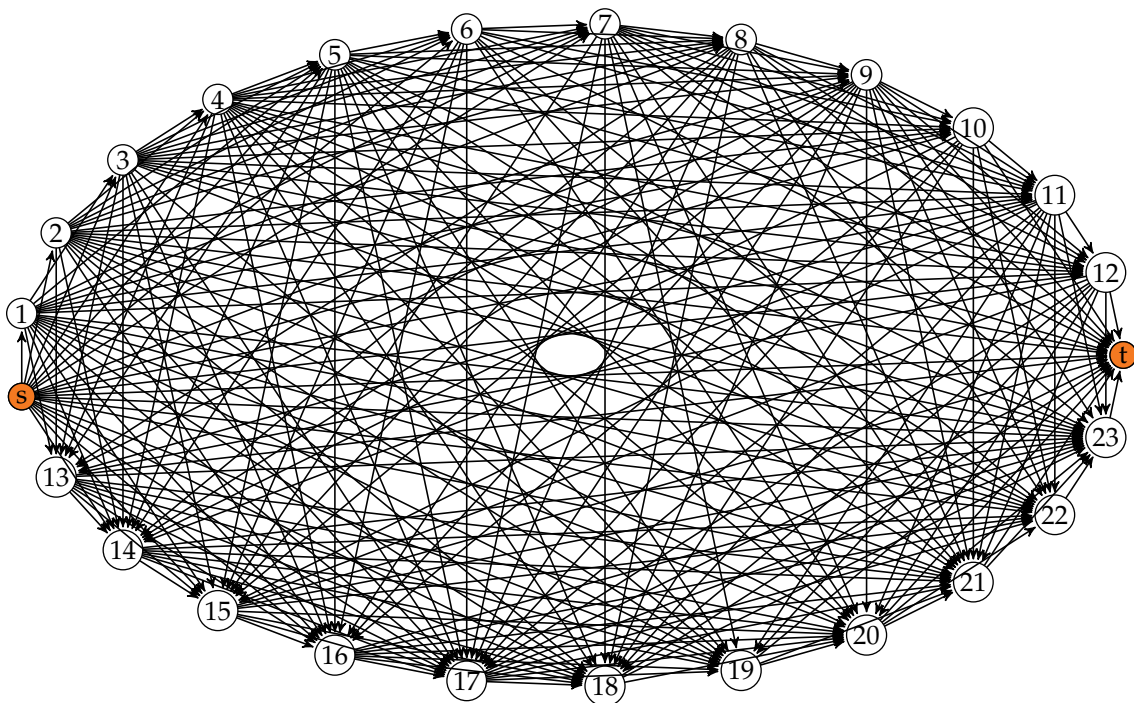


Figure D.7: Network gnp03 — $GNP(25, 0.4)$



VITA

Juliana Bright

Candidate for the Degree of

Master of Science

Thesis: ROBUST SHORTEST PATHS UNDER UNCERTAINTY USING CONDITIONAL VALUE-AT-RISK

Major Field: Industrial Engineering and Management

Biographical:

Personal Data: Born in London, UK on April 22, 1980.

Education: Bachelor of Arts in German from the University of Oklahoma, December 2005. Bachelor of Science in Industrial Engineering and Management from Oklahoma State University in Stillwater, Oklahoma in August 2009. Completed the Requirements for the Master of Science degree at Oklahoma State University in December 2011 with a major in Industrial Engineering and Management.

Experience: Graduate research assistant, Industrial Engineering and Management, Oklahoma State University, Fall 2010-Spring 2011; undergraduate and graduate teaching assistant, CEATLabs, Oklahoma State University, January 2008-August 2010; staff assistant, Dollar-Thriftly Automotive Group, Tulsa, OK, June 2004-September 2006.

Professional Memberships: Engineering Intern, Institute for Operations Research and Management Science (INFORMS), Institute of Industrial Engineers (IIE)

Awards and Honors: Doctoral Academy Fellowship, University of Arkansas; First place in the 2010 IIE Undergraduate Student Technical Paper Competition.

Name: Julie Anna Bright

Date of Degree: December 2011

Institution: Oklahoma State University

Location: Stillwater, Oklahoma

Title of Study: ROBUST SHORTEST PATHS UNDER UNCERTAINTY USING
CONDITIONAL VALUE-AT-RISK

Pages in Study: 73

Candidate for the Degree of Master of Science

Major Field: Industrial Engineering and Management

Finding a shortest path in a network is a classical problem in discrete optimization. The systems underlying the network models are subjects to a variety of sources of uncertainty. The purpose of this thesis is to model the shortest path problem with probabilistic arc failures with the aim of finding short but reliable paths through the network. This thesis proposes to use Conditional Value-at-Risk (CVaR) to find a path that is robust under probabilistic arc failures. CVaR, a quantitative risk measure, is roughly the mean excess loss associated with a decision. This thesis also develops three models of losses due to arc failures. Mixed integer linear programming models for the stochastic shortest path problem with probabilistic arc failures are formulated and implemented. The optimization models limit the CVaR of the loss due to arc failures as measured by the models of loss developed in the thesis.

ADVISOR'S APPROVAL: Dr. Balabhaskar Balasundaram