# Strategic Network Planning Under Uncertainty with Two-Stage Stochastic Integer Programming

by

Zhihao Chen

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Industrial and Operations Engineering)
in the University of Michigan
2016

Doctoral Committee:

Assistant Professor Siqian M. Shen, Chair
Associate Professor Marina A. Epelman
Assistant Professor Kwai Hung Henry Lam
Assistant Professor Cong Shi
Assistant Professor Ming Xu

---

2016

To Mum and Dad
To my sister and my two adorable nieces
To Joy
And to anyone else who believed in me

# ACKNOWLEDGMENTS

This thesis would not have been possible without the love and support from my family – my parents who taught me kindness and humility, and my sister who always reminds me to believe in myself. They are, and always will be, my pillar of light and my source of strength.

My most sincere gratitude goes to my advisor, Siqian Shen. She took a chance with me four years ago, despite my lack of research experience; for the four years following that, she guided me through my PhD with her tireless passion. Her counsel was vital to me academically, but it was her endless exuberance that made my PhD far more enjoyable, and I am glad to call her both a mentor and a friend.

I would also like to thank my dissertation committee, whose invaluable input helped refine my research. Despite my typically hastily scheduled meetings with them, they have been nothing but patient in offering advice to improve my dissertation. Their suggestions will be the fuel for my future research, especially those with regards to the carsharing model.

Finally, my time in Ann Arbor would have been far less entertaining without my friends: Joy, whose countless text messages gave me the determination to keep going; Yan, who shared the same PhD journey; Bryan, Darren and Chor Seng, who helped pull me through the inevitable late nights; Hai Bin, Ziyang, Cheerin, Christopher, Arlena, Feiya, Ryan, and Amos, who gave me something to look forward to on weekends.

# TABLE OF CONTENTS

# LIST OF FIGURES

# LIST OF TABLES

# LIST OF ALGORITHMS

# LIST OF APPENDICES

# ABSTRACT

**Strategic Network Planning Under Uncertainty with Two-Stage Stochastic Integer Programming**

**by**

**Zhihao Chen**

**Chair: Siqian Shen**

This thesis proposes three risk-averse models: a chance-constrained approach to network design problems (NDPs), a distributionally robust approach to NDPs, and a car-sharing model to maximize profitability. These two-stage models are applied under demand uncertainty – the first stage makes strategic design decisions for the network, namely arc capacities and supply volume, and the second stage utilizes risk-averse approaches to ensure high levels of demand satisfaction while minimizing costs – network design costs, commodity flow costs, and potential penalty costs for not satisfying all units of demanded commodity.

The first model optimizes the probabilistic network design problem (PNDP), where we maintain quality of service (QoS) through chance constraints. The second-stage problem differentiates the corresponding QoS for demand satisfaction, and prioritizes for customers and/or commodities. We consider PNDP variants that have either continuous or binary arc decision variables (in the first stage) according to different applications. Each probabilistic model is reformulated as a mixed-integer lin-

ear program (MILP), and we develop polynomial-time algorithms for special cases with single-row chance constraints. The PNDP is benchmarked against stochastic programming models that either enforce meeting all demands or penalize any unmet demand via a linear penalty function. We compare different models and approaches by testing randomly generated network instances and the Sioux-Falls network. The numerical results suggest potential cost savings in arc capacity design and commodity flow for customized QoS parameters and also showed a huge improvement in computation efficiency when using the polynomial-time algorithm over the MILP formulation.

The second model proposes a distributionally robust NDP (DR-NDP) with a marginal moment-based ambiguity set, to obtain arc capacity solutions that optimize the worst-case total cost over all candidate distributions. By approximating polynomials with piecewise-linear functions, we estimate the optimal value of DR-NDP with an MILP, optimized via a cutting-plane algorithm that iteratively generates valid cuts for the approximate problem. We compare the performance of DR-NDP solution against those of a benchmark sample average approximation-based model, testing on grid networks of various sizes and a network based on major roads in the United States. Our results show the robustness of DR-NDP solutions and how they respond to varying demand levels in the observed realizations, highlighting potential niche uses of DR-NDP in data-scarce contexts.

The third model is formulated as a two-stage stochastic integer programming model with the aim of allocating a carsharing fleet to service zones with contracted parking lots under demand uncertainty. The first-stage problem determines the allocation of shared car fleet to pre-designated zones, and the number of contracted parking lots to rent or free-float permits to obtain in these zones. In the second stage, we solve a stochastic minimum cost flow problem on a spatial-temporal network, which considers random demand of one-way and round-trip rentals, as well as ad-

hoc vehicle relocation, to optimize total profits less any penalties from unsatisfied demand. We minimize the expected penalty cost of unserved customers to encourage higher QoS, and also consider a risk-averse variant of the second-stage model that penalizes the conditional value-at-risk of unsatisfied demand. To solve both models, we develop a branch-and-cut algorithm with mixed-integer rounding-enhanced Benders cuts. We provide insights of stochastic carsharing system management by testing instances generated from real data reported by Zipcar in the Boston area. We find that profitability and QoS decrease with increased proportions of one-way rental demand, and the QoS can be significantly improved by lowering relocation costs.

# CHAPTER 1

# Introduction

We propose three novel models in the area of network design and flow optimization under demand uncertainty. Each model is a two-stage model in which the first stage makes strategic, long-term decisions on the design of the network (e.g. arc capacities, node supplies). These strategic decisions are intended to be risk-averse, such that lower costs and higher quality of service (QoS) may be achieved in the second stage. The second stage is a minimum cost flow problem that maintains high QoS through penalties in the objective or chance constraints, depending on the model used. The first two models address a network design problem that determines arc capacities in the first stage, and apply a chance-constrained and a distributionally robust approach in the second stage. The third model is an application to carsharing systems, that determines the initial allocation of the fleet in the first stage, and maximizes operational profit in the second stage, using a spatial-temporal network to determine revenues and costs and a conditional value-at-risk (CVaR) penalty in the objective to maintain high QoS.

## 1.1 Network Design Problems

Network design problems (NDPs) refer to an important class of problems that frequently arise in the modern connected world. Their applications include traffic planning, energy distribution, telecommunication, and supply chain management, all of which are essential to our daily lives (see Section 1.3 for more detailed examples). In this thesis, we focus on NDPs under demand uncertainty, and aim to optimally determine the arc capacities of a given network to satisfy demand realizations through ad-hoc flows. Chapters 2 and 3 propose risk-averse models to solving the NDP – Chapter 2 proposes a chance-constrained model to limit the probability of demand loss, while Chapter 3 proposes a distributionally robust model that optimizes the worst-case objective value over a set of candidate distributions.

The types of design variables for arcs fall into two broad categories: (i) continuous variables, for which the value of the variable equals to the capacity allocated to the arc, and (ii) 0–1 binary variables, which indicate whether the arc is built or not with a fixed, predetermined capacity. The former type, which is used in both Chapters 2 and 3, can be used in NDPs in which there can more flexibility in the capacity of arcs built. The latter type, addressed only in Chapter 2, is more likely to be employed if there are prior restrictions on arc capacity, and the arcs need only be selected. Regardless of the type of design variable used, constructing an arc incurs a cost (per-unit cost for continuous, fixed cost for binary), and the arcs are designed in the first stage of the NDP.

In the second stage, the amount of commodity flow on the arcs is determined. Flow values are typically recourse decisions made after demand is realized, with the exception of the probabilistic NDP (PNDP, see Chapter 2), in which flow is determined before demand is realized. Flow are always continuous and incur a cost per unit of flow. QoS is an important concept in this research, and is measured by the amount of demand that is satisfied. To ensure a high QoS level, any unmet demand is proportionally penalized in the objective function, to discourage not satisfying demand to save (arc construction and flow) cost. This, again, is with the exception of the PNDP, which ensures QoS by limiting the probability of demand not being met. Overall, the aim of the NDP is to minimize the total cost of arc construction, commodity flow, and penalty on unmet demand.

## 1.2 Previous Work on NDPs

NDPs are commonly used in network planning and operation problems, so it is unsurprising that there is much literature studying its theory and applications. In generic network planning contexts, Magnanti and Wong (1984) described a unifying framework for deterministic single and multi-commodity NDPs, and provided specializations of the framework to solving specific problems such as facility location and traffic network design problems. These specific classes of NDPs are also studied in depth in the literature, for example, facility location design in Daskin (2011), road network design in Yang and H. Bell (1998), and telecommunications and computer network design in Pióro and Medhi (2004).

Parameter uncertainty in networks is a major concern in the study of NDPs. This is especially so when studying real-world applications, since the practical design of networks is usually in anticipation of future uncertain needs and circumstances. Lium et al. (2009) studied demand uncertainty in stochastic NDPs comprehensively, focusing particularly on applications to service network design. Network design under demand uncertainty has also been studied in other applications, such as in supply-chain network design (Santoso

et al., 2005; Tsiakis et al., 2001; MirHassani et al., 2000), in transportation network design (Ukkusuri and Patil, 2009; Patil and Ukkusuri, 2007) and in emergency response (Chang et al., 2007; Sheu, 2007; Oh and Haghani, 1997).

The literature on stochastic NDPs typically assume fully known distributional information on the demand uncertainty, allowing the formulation of large-scale stochastic programs with a finitely large number of realizations. The L-shaped method (Laporte and Louveaux, 1993) is usually used to derive cuts to iteratively optimize such large-scale stochastic programs. Patil and Ukkusuri (2007) applied the L-shaped method to transportation NDPs, although Crainic et al. (2011) more recently proposed a scenario decomposition metaheuristic for similar problems. However, the L-shaped method requires sufficiently many realizations to accurately represent the underlying distribution, and may not produce meaningful results if realizations cannot be easily or cheaply generated from the true distribution.

In cases where more conservative NDP solutions are needed, or where the demand distribution is not fully known, robust optimization is often used to solve NDPs. Robust optimization utilizes bounds on the uncertain parameters to construct an uncertainty set, and optimize the worst-case objective value for any parameter realization in the uncertainty set. Since only the bounds of the uncertainty are required for robust optimization, it is well-suited to conservatively solve problems without full information on the distribution of uncertainty. Ukkusuri et al. (2007) robustly designed a transportation network under discrete demand uncertainty. Similarly, Mudchanatongsuk et al. (2008) proposed a robust transportation NDP under demand and transportation cost uncertainty, representing the uncertainty in both cases with box-shaped, polyhedral and ellipsoidal uncertainty sets. In supply chain network design, Pishvaee et al. (2011) applied robust optimization to minimize the cost of the supply chain network under box-shaped uncertainty sets for demand, returns, and transportation costs.

## 1.3 Applications of Risk-Averse NDP Models

Similar to robust NDPs, the chance-constrained PNDP and distributionally robust NDP (DR-NDP) models that are proposed in Chapters 2 and 3 seek conservative solutions to the NDP. These models are risk-averse – they aim to achieve high QoS levels through high demand satisfaction, often at the expense of constructing more arc capacity to ensure sufficient commodity can be flowed from supply to demand nodes. Due to the risk-aversion of these two proposed models, this research specifically targets NDP applications in which high QoS solutions are of importance to the network planner, or in which there is insufficient data to accurately determine the true underlying distribution for a meaningful

risk-neutral solution. The following are several examples of such applications.

- **Humanitarian relief supply network design:** When providing humanitarian relief immediately after the occurrence of disaster, the effects of the disaster may have yet to be determined. Demand for aid is unknown and little prior data is likely to be known due to the rare occurrence of disaster, yet high QoS must still be maintained to provide aid to as many people in the affected region as possible.

- **Telecommunications network design:** Telecommunications companies frequently have to expand their networks in response to the increased need for connectivity in modern society. As the network expansion in response to insufficient capacity, data on the true demand (not restricted by the current capacity) in the network cannot be gathered from past usage data. To obtain this data, companies might instead survey their customers on their usage habits. Since this data is costly, companies may opt to obtain only a small amount of such data and employ risk-averse models to maintain high QoS to its customers.

- **Supply chain network for new consumer products:** Releasing new products, or pushing out products to new regions requires careful planning of the supply network for the products. Due to the youth of the product, data regarding consumer demand is likely to be uncertain and scarce. A risk-averse model could ensure products are launched with relatively high levels of customer satisfaction at the cost of a higher expense on constructing the supply chain network.

Apart from the two models proposed in Chapters 2 and 3, a third risk-averse approach can be taken – penalizing the CVaR of unmet demand in the objective function. This approach is applied to a carsharing system in Chapter 4, to maximize profitability and QoS.

## 1.4 Carsharing

High vehicle ownership costs and increased awareness of climate change due to pollution has resulted in a greater desire for cheaper yet more environmentally friendly ways to move people. In particular, the latter reason, together with frequent traffic congestion in large cities, have prompted many governments to explore other transportation options to alleviate environmental and traffic concerns. Public transportation has traditionally been a common alternative to private vehicle ownership. However, public transport is far from being a perfect replacement due to its limited accessibly, fixed schedules, and the fact that

users have to share a common space, posing problems to individuals with needs that may inconvenience their fellow riders.

In recent years, carsharing has become a popular means of alternative transportation, serving as a middle ground between public transport and private ownership. Carsharing companies provide car rental services to customers on a short term basis, as opposed to long term rentals provided by typical car rental companies. The popularity of carsharing can be attributed to users benefiting from private use of vehicles without having to shoulder the costs or responsibilities associated with car ownership. Carsharing also provides other benefits by reducing vehicle ownership in cities, namely reduced traffic congestion and reduced total vehicle miles traveled. Furthermore, fewer vehicle miles traveled means lower fuel consumption, and consequently lower vehicle emissions (Fan et al., 2008). In North America, studies suggest that carsharing has reduced vehicle mileage by 44% on average per carsharing user, with each carsharing vehicle replacing between 6 to as many as 23 vehicles (Shaheen and Cohen, 2007). The benefit of carsharing is further expanded by the use of electric vehicles – advances in electric vehicle charging technology will lower carsharing fleet sizes and operational costs, and bring greater reduction in $CO_2$ emissions (He et al., 2015). Hence, it is not surprising that almost 1,000 cities worldwide have adopted carsharing, with over 1,337,000 individuals sharing almost 20,000 vehicles through carsharing programs in the United States alone as of July 2014 (Carsharing, 2009).

To carsharing companies, optimally locating their vehicle fleet in response to demand is important to their profitability and QoS. As more carsharing companies begin to offer one-way rentals in addition to traditional round-trip rentals, such as Zipcar through its ONE>WAY beta program, the problem of optimal fleet allocation and ad-hoc relocation of vehicles becomes increasingly complex. We address the aforementioned operational concerns in Chapter 4, with the aim of providing a mechanism to determine the impact of one-way rentals on the profitability and QoS of carsharing systems.

## 1.5 Contributions of Research

The main contributions of this research are the three aforementioned models – the PNDP, the DR-NDP, and the carsharing model. Chapter 2 formulates PNDPs with various forms of chance constraints to satisfy different needs for the QoS levels. The PNDPs generalize the chance-constrained models in recent transportation research literature that analyze the NDP under various transportation settings. We reformulate and solve the PNDPs as mixed-integer linear programs. For special cases of the PNDP, we develop an algorithm that reformulates them in polynomial time to a deterministic linear program, which can be

solved far quicker than the MILP formulation.

Chapter 3 formulates the DR-NDP as a novel approach to solving NDPs. The majority of the literature on NDPs focuses on problems with fully known distributional information; when distributional information is sparse, robust optimization is used instead. The DR-NDP provides a means to solve NDPs that is less conservative than using robust optimization, yet is robust enough for its solutions to be less sensitive to the distribution of the observations, when compared to using a risk-neutral stochastic optimization approach. We reformulate the DR-NDP as an approximate MILP and develop a cutting-plane algorithm to solve the MILP.

Chapter 4 models a carsharing system as a spatial-temporal network that can approximate the profitability and QoS provided to customers through a minimum cost flow problem. Strategic decisions can be made with this framework – we focus on the decisions of purchasing parking lots and free-float permits together with car fleet allocation in zones. We integrate both one-way and round-trip rentals, and reservation-based and free-floating carsharing decisions into a generalized model, which has not been done in carsharing literature. We also develop a branch-and-cut algorithm strengthened by mixed-integer rounding (MIR) to solve the carsharing model, which shows promising computational performance when implemented in parallel computing.

# CHAPTER 2

# Probabilistic Network Design

## 2.1 Introductory Remarks

In this chapter, (related to the work published in Shen and Chen, 2013), we study PNDPs under demand uncertainty and with multi-commodity flows, which can be interpreted as shipments of multiple products, or other types of heterogeneous flows involved in a wide class of applications. The flow variables obey balance constraints and knapsack constraints that limit the summation of all flows within a shared capacity on every arc (e.g. Ahuja et al., 1993). Instead of an expectation-based stochastic programming approach, we use probabilistic constraints, or chance constraints (Charnes et al., 1958), to differentiate demand satisfaction rates of shipping multiple commodities to different nodes. This chapter formulates four types of chance-constrained models to allow the flexibility in differentiating QoS levels with respect to commodity and node-wise demand.

Recent literature has used chance-constrained programming to analyze NDP variants under diverse settings (e.g. Chen et al., 2011). To name a few, Waller and Ziliaskopoulos (2001) formulate a chance-constrained model for studying continuous NDPs with traffic dynamics and random time-dependent demands. Lo and Tung (2003) study the tradeoff between the maximum flow in a network and the extent of satisfying chance constraints of the probabilistic user equilibrium, given that link capacities are subject to stochastic degradations. Chen and Yang (2004) account for both spatial equity and demand uncertainty, and formulate the equity constraint as a chance constraint. Chen et al. (2007) develop a "alpha reliable NDP model" which is a variant of the chance-constrained NDP to optimize network design decisions under demand uncertainty with different risk aversion levels.

Following the order of PNDP variants discussed in this chapter, we determine both flow variables and continuous capacity expansion variables of existing links at the upper level, and evaluate demand satisfaction rates at the lower level. The aim is to minimize the total cost of capacity expansion and flow assignment subject to various forms of chance

constraints for bounding the demand losses. We justify this approach by the application of supply chain design, where a flow scheduler needs to be decided before knowing the uncertainty and cannot be easily adjusted in different scenarios. For instance, due to high contracting fees and the ease of maintaining a relationship with a stable set of suppliers and customers, the scheduler may prefer a fixed shipping schedule regardless of actual daily demand when the demand fluctuation is not significant. In some emergency response applications, reaction time for changing the delivery plan may be too short so that fixed flows are more favorable.

We formulate benchmark risk-neutral stochastic network design problems (SNDPs) by letting flows be recourse variables, whose values are determined *after* knowing the demand. The modification results in the flexibility of having different flow decisions in each scenario, but enforces a hard constraint on the flow decisions. We consider two types of SNDPs, one does not penalize unmet demand, the other adds a penalty cost proportional to unmet demand. The latter is typically used in the existing literature and will provide benchmarks in our numerical results. Furthermore, we consider PNDPs and SNDPs with binary design variables, representing the addition of new links, and provide the corresponding variants under this assumption.

We transform all probabilistic models into equivalent MILP formulations under the assumption of finitely distributed random demand. For some special cases of the PNDP, we present alternative polynomial-time algorithms. We demonstrate the relationship of PNDP variants via their risk parameters such as the reliability levels associated with various chance constraints. For large-scale PNDP/SNDP models, we describe a general Benders decomposition approach to improve the computational efficiency. In our results, we present managerial insights by computing on randomly generated network instances and the Sioux-Falls network.

The remainder of the chapter is organized as follows. Section 2.2 assumes continuous network design variables and formulates various PNDPs with fixed flow variables and SNDPs with recourse flow variables. Section 2.3 describes NDP formulations with discrete network design variables. Section 2.4 develops solution methodologies, and demonstrates polynomial-time algorithms for special cases of the PNDP with single-row chance constraints. Two sets of numerical examples are given in Section 2.5: the first consists of randomized instances to test various PNDP formulations and their algorithms; the second simulates a practical setting with the commonly used Sioux Falls network, in which we compare PNDPs with SNDPs. Section 2.6 concludes the chapter and describes future research directions.

## 2.2 NDPs with Continuous Capacity Design Variables

We begin with the assumption that all upper-level decision variables are continuous, corresponding to the case where we can smoothly increase the capacity of existing links. This section describes the two cases where the multi-commodity flow variables are fixed before and decided after realizing the demand uncertainty.

### 2.2.1 Notation and Assumptions

We formulate the NDPs on a directed connected graph $G(N, A)$, where $N$ is the set of nodes and $A \subset N \times N$ denotes the set of arcs. The set of commodities are denoted by $|W|$, with the sets of supply nodes $S_w$ and demand nodes $D_w$ for each commodity $w \in W$. The supply $s_{iw} > 0$ for each $i \in S_w$, $w \in W$ is deterministic; the demand $d_{iw} \geq 0$ for each $i \in D_w$, $w \in W$ is random, with all $d_{iw}$ jointly distributed such that the demand vector $d = [d_{iw} : i \in D_w, w \in W]^\mathsf{T}$ follows a known joint discrete distribution. The set of $|K|$ realizations of $d$ is denoted by $\{d^k\}_{k \in K}$, with $p^k$ being the probability of realizing scenario $k \in K$. When the first-stage design variables are continuous, we denote the arc capacity variables by $x_{ij} \in \mathbb{R}_+$, and each unit of arc capacity incurs a cost of $c_{ij} > 0$ for all arcs $(i, j) \in A$.

We name the models in this chapter in the form "`Problem-Capacity-Constraint`," where the types of `Problem` are either "PNDP" (flow fixed at the upper level) or "SNDP" (flow recourse available at the lower level), and the types of `Capacity` are either "-cont" (continuous values of capacity decisions at existing links) or "-bin" (binary decisions of adding new links). We refer to "-joint," "-nc," "-c," and "-n" as the types of `Constraint` for PNDPs, representing the cases where the chance constraints are formulated with respect to the overall joint, node-commodity-wise, commodity-wise, and node-wise probabilistic restrictions of unsatisfied demand, respectively; it also refers to "-wop" and "-wp" as the types of `Constraint` for SNDPs, representing the models with penalty and without penalty in the objective, respectively.

### 2.2.2 Fixed Flow Variables at the Upper Level

We first analyze PNDP-cont and give its four variations, where the flow decisions are fixed *before* the realization of demands. The variations are determined by the manner in which the uncertain demand constraints are joined by the chance constraints: by joining all of them (-joint), node-commodity-wise (-nc), commodity-wise (-c), and node-wise (-n).

We first focus on the joint case, then present the remaining variants as multiple chance-constrained models.

#### 2.2.2.1 Joint Chance Constraints

This variant involves a joint chance constraint for measuring demand satisfaction of all commodities at all nodes. The aim is to minimize the total cost of capacity design and network flows, while the probability of no demand loss of every commodity at every node is bounded from below by a given reliability level $1-\epsilon$. The joint chance constraint is given by

$$\mathbb{P}\left(\sum_{j:(j,i)\in A} y_{jiw} - \sum_{j:(i,j)\in A} y_{ijw} \geq d_{iw},\ \forall i \in D_w,\ w \in W\right) \geq 1-\epsilon, \tag{2.1}$$

where $y_{ijw} \in \mathbb{R}_+$ is the variable determining the amount of commodity $w \in W$ to flow from node $i$ to node $j$. The notation $\mathbb{P}(\cdot)$ denotes the probability of uncertain event $\cdot$ occurring under the given distribution for $d$. The expression $\sum_{j:(j,i)\in A} y_{jiw} - \sum_{j:(i,j)\in A} y_{ijw}$ represents the overall amount of commodity $w$ received at demand node $i$, which must be at least the demand $d_{iw}$ for all $i \in D_w$ and $w \in W$ with probability $1-\epsilon$.

By assuming discretely distributed demand, we transform all chance-constrained models in this chapter as equivalent MILP formulations. Define binary variables $z^k$ indicating whether constraint (2.1) is violated in scenario $k$, such that $z^k = 1$ if it is, and $z^k = 0$ otherwise for all $k \in K$. Then we have

**[PNDP-cont-joint]:**

$$\min_{x,y,z} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{w\in W}\sum_{(i,j)\in A} a_{ijw}y_{ijw} \tag{2.2}$$

$$\text{s.t.} \quad \sum_{w\in W} y_{ijw} \leq x_{ij} \quad \forall (i,j) \in A \tag{2.3}$$

$$\sum_{j:(i,j)\in A} y_{ijw} - \sum_{j:(j,i)\in A} y_{jiw} \leq s_{iw} \quad \forall i \in S_w,\ w \in W \tag{2.4}$$

$$\sum_{j:(i,j)\in A} y_{ijw} - \sum_{j:(j,i)\in A} y_{jiw} = 0 \quad \forall i \in N\backslash(S_w \cup D_w),\ w \in W \tag{2.5}$$

$$-\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} - d^k_{iw} + M^k z^k \geq 0 \quad \forall i \in D_w,\ w \in W,\ k \in K \tag{2.6}$$

$$\sum_{k\in K} p^k z^k \leq \epsilon, \tag{2.7}$$

$$x \geq 0,\ y \geq 0,\ z \in \{0,1\}^{|K|}, \tag{2.8}$$

where $a_{ijw} > 0$ is the unit cost of flow of commodity $w \in W$ on arc $(i,j) \in A$, and $M^k > 0$ is an arbitrarily large positive number. Constraint (2.3) ensures that the total flow of commodities at arc $(i,j)$ does not exceed allocated capacities, for all $(i,j) \in A$; (2.4) and (2.5) impose commodity-wise flow balances at supply and transshipment nodes, respectively; (2.6) enforces $z^k = 1$ for some scenario $k \in K$, if at least one commodity at one demand node has a positive demand loss in that scenario. Constraints (2.6) and (2.7) together yield a deterministic equivalence of the joint chance constraint (2.1).

For convenience of comparison with different models, we rewrite (2.6) as

$$\left[ -\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} - d^k_{iw},\ \forall i \in D_w,\ w \in W \right] + M^k z^k \geq 0 \quad \forall k \in K, \tag{2.9}$$

emphasizing that each row $-\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} - d^k_{iw}$, $w \in W,\ i \in D_w$ shares a common binary variable $z^k$ with a big-$M$ coefficient, for all scenarios $k \in K$. Constraint (2.7) guarantees that the probability of violation is no more than $\epsilon$. By having an integer capacity cost vector $c$, flow cost vector $a$, supply vector $s$, demand vector realizations $d^k$ for all $k \in K$, and large number vector $M$, there exists an optimal solution having integer capacity vector $x$ and flow vector $y$.

#### 2.2.2.2 Multiple Chance Constraints

In the remaining three variants of PNDP-cont, the joint chance constraint (2.1) is split into multiple smaller-scale chance constraints. In the first variant PNDP-cont-nc, one single-row chance constraint is imposed for each commodity $w$ and each demand node $i \in D_w$ with an individual risk tolerance $\epsilon_{iw}$, giving

$$\mathbb{P}\left( \sum_{j:(j,i)\in A} y_{jiw} - \sum_{j:(i,j)\in A} y_{ijw} \geq d_{iw} \right) \geq 1 - \epsilon_{iw},\ \forall i \in D_w,\ w \in W. \tag{2.10}$$

We motivate the study of (2.10) as follows. First, it allows the customization of risk tolerances for different node-commodity pairs, rather than assume homogeneous demand-satisfaction guarantees everywhere. In other words, we allow the differentiation of the importance of products and customers by appropriately choosing $\epsilon_{iw}$ for each combination of commodity $w$ and node $i$. Second, a decision maker can vary values of $\epsilon_{iw},\ \forall w \in W,\ i \in D_w$, and derive corresponding optimal risk-and-cost tradeoffs – this value-varying process can be implemented via sensitivity analysis. Alternatively, by treating all $\epsilon_{iw}$ as decision variables within certain preferable ranges, we can consider $x$, $y$, and $\epsilon$ as decision variables,

and seek solutions that simultaneously trade off between risk and cost.

According to similar motivations, the other two variants involve

- A joint chance constraint for each commodity with risk tolerance $\epsilon_w$ (PNDP-cont-c):

$$\mathbb{P}\left(\sum_{j:(j,i)\in A} y_{jiw} - \sum_{j:(i,j)\in A} y_{ijw} \geq d_{iw},\ \forall i \in D_w\right) \geq 1-\epsilon_w,\ \forall w \in W. \tag{2.11}$$

- A joint chance constraint for each node with risk tolerance $\epsilon_i$ (PNDP-cont-n):

$$\mathbb{P}\left(\sum_{j:(j,i)\in A} y_{jiw} - \sum_{j:(i,j)\in A} y_{ijw} \geq d_{iw},\ \forall w \in W\right) \geq 1-\epsilon_i,\ \forall i \in D_w. \tag{2.12}$$

Constraints (2.11) and (2.12) differentiate risk perceptions with respect to commodities and demand nodes, respectively. The corresponding models involve multiple joint chance constraints. By assuming a discrete joint demand distribution, we reformulate each model variant as an equivalent MILP, with binary variables indicating violation status of the chance constraints in each scenario. Analogously to the construction of PNDP-cont-joint, the reformulation procedures yield

**[PNDP-cont-nc]:**

$$\min_{x,y,z} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{w\in W}\sum_{(i,j)\in A} a_{ijw}y_{ijw}$$

$$\text{s.t.} \quad (2.3)\text{–}(2.5)$$

$$-\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} - d^k_{iw} + M^k_{iw}z^k_{iw} \geq 0 \quad \forall i \in D_w,\ w \in W,\ k \in K \tag{2.13}$$

$$\sum_{k\in K} p^k z^k_{iw} \leq \epsilon_{iw} \quad \forall w \in W,\ i \in D_w \tag{2.14}$$

$$x \geq 0,\ y \geq 0,\ z_{iw} \in \{0,1\}^{|K|},\ \forall w \in W,\ i \in D_w, \tag{2.15}$$

where $M^k_{iw} > 0$ is an arbitrary large positive number for each commodity $w$ and node $i \in D_w$, and the binary variable $z^k_{iw}$ takes the value 1 if there exists a demand loss of commodity $w$ at demand node $i$ in scenario $k$, and 0 otherwise, for all $i \in D_w,\ w \in W, k \in K$.

By defining binary variables

- $z^k_w$: $= 1$ if there exists a demand loss of commodity $w$ on at least one demand node $i \in D_w$, and $= 0$ otherwise, for all scenarios $k \in K$, and

- $z_i^k$: $= 1$ if there exists at least one commodity demand loss at node $i$, and $= 0$ otherwise, for all scenarios $k \in K$,

and their corresponding big-$M$ coefficients $M_w^k$ and $M_i^k$, we can formulate the other two PNDP-cont models as:

**[PNDP-cont-c]:**

$$\min_{x,y,z} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{w\in W}\sum_{(i,j)\in A} a_{ijw}y_{ijw}$$

s.t. (2.3)–(2.5)

$$\left[-\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} - d_{iw}^k,\ \forall i \in D_w\right] + M_w^k z_w^k \geq 0 \quad \forall w \in W,\ k \in K \tag{2.16}$$

$$\sum_{k\in K} p^k z_w^k \leq \epsilon_w \quad \forall w \in W \tag{2.17}$$

$$x \geq 0,\ y \geq 0,\ z_w \in \{0,1\}^{|K|},\ \forall w \in W, \tag{2.18}$$

and,

**[PNDP-cont-n]:**

$$\min_{x,y,z} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{w\in W}\sum_{(i,j)\in A} a_{ijw}y_{ijw}$$

s.t. (2.3)–(2.5)

$$\left[-\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} - d_{iw}^k,\ \forall w \in W\right] + M_i^k z_i^k \geq 0 \quad \forall i \in \bigcup_{w\in W} D_w,\ k \in K \tag{2.19}$$

$$\sum_{k\in K} p^k z_i^k \leq \epsilon_i \quad \forall i \in \cup_{w\in W} D_w \tag{2.20}$$

$$x \geq 0,\ y \geq 0,\ z_i \in \{0,1\}^{|K|},\ \forall i \in \bigcup_{w\in W} D_w. \tag{2.21}$$

**Remark 2.1.** PNDP-cont-c and PNDP-cont-n can be viewed as hybrid versions of PNDP-cont-joint and PNDP-cont-nc. PNDP-cont-joint measures unsatisfied demands simultaneously and homogeneously for all commodities and nodes, being more conservative than PNDP-cont-nc if subject to the same magnitude of risk tolerances. A scenario in PNDP-cont-joint is considered as "failed" when there exists one location with demand shortage

for one product. PNDP-cont-nc "frees" the feasible region to some extent (we also need to compare specific values of $\epsilon$ and $\epsilon_{iw}$, for each $i$ and $w$ to draw a more precise statement), and separates probabilistic measures for satisfying each individual $d_{iw}$. If a decision maker is only interested in evaluating the chance of positive demand loss for every commodity, or for every demand location, we formulate PNDP-cont-c and PNDP-cont-n, respectively. Both models are more conservative than PNDP-cont-nc, and less conservative than PNDP-cont-joint in general. □

We describe in Section 2.4 the methodological details for optimizing the four MILP models and algorithms in polynomial time for optimizing special cases of PNDP-cont in different forms.

### 2.2.3 Recourse Flow Variables at the Lower Level

Here we consider the case where capacity design decisions are still continuous, but flow decisions are made *after* the realization of demands, or in other words, SNDP-cont. We present two variations of SNDP-cont, one which does not impose a penalty, and the other which imposes a penalty on unmet demand in any of the possible scenarios.

#### 2.2.3.1 Without Penalty on Unmet Demand

We first highlight the differences between the formulations of SNDP-cont and those of its PNDP-cont counterparts. As flow decisions are made after demand realization, each scenario $k \in K$ has a flow decision variable $y^k_{ijw}$, as opposed to a common flow decision variable $y_{ijw}$ across all scenarios in the fixed flow case. Furthermore, since the demands are already known when the flow decision is to be made, the formulation no longer requires the binary variable $z$ to indicate whether the flow is feasible in a particular scenario – the flow vector $y^k$ must now satisfy the constraints with the demand realization $d^k$, for each scenario $k$. In addition, the objective value considers the *expected* cost of the flow decisions, weighted by the probability of each scenario occurring, as there are now multiple flow costs associated with the demand realization in each scenario. We can write this expectation explicitly as the sum of the flow costs in each scenario weighted by the probability of the scenario occurring.

The following shows the full formulation of the SNDP-cont-wop problem, which holds many similarities to the PNDP-cont-nc problem. We note here that the demand constraints are always node-commodity-wise constraints (no joint constraints), unlike in the PNDP-cont case, where demand constraints could be joined in various ways.

**[SNDP-cont-wop]:**

$$\min_{x,y} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{k\in K} p^k \left( \sum_{w\in W} \sum_{(i,j)\in A} a_{ijw} y^k_{ijw} \right) \tag{2.22}$$

$$\text{s.t.} \quad \sum_{w\in W} y^k_{ijw} \leq x_{ij} \quad \forall (i,j) \in A,\ k \in K \tag{2.23}$$

$$\sum_{j:(i,j)\in A} y^k_{ijw} - \sum_{j:(j,i)\in A} y^k_{jiw} \leq s_{iw} \quad \forall i \in S_w,\ w \in W,\ k \in K \tag{2.24}$$

$$\sum_{j:(i,j)\in A} y^k_{ijw} - \sum_{j:(j,i)\in A} y^k_{jiw} = 0 \quad \forall i \in N \backslash S_w \cup D_w,\ w \in W,\ k \in K \tag{2.25}$$

$$- \sum_{j:(i,j)\in A} y^k_{ijw} + \sum_{j:(j,i)\in A} y^k_{jiw} \geq d^k_{iw} \quad \forall i \in D_w,\ w \in W,\ k \in K \tag{2.26}$$

$$x \geq 0,\ y^k \geq 0,\ \forall k \in K \tag{2.27}$$

#### 2.2.3.2 With penalty on unmet demand

Here we present the formulation of SNDP which penalizes unfulfilled demands. A variable $t^k_{iw} \in \mathbb{R}_+$ is introduced for every $i \in D_w, w \in W, k \in K$, to represent the amount of demand for commodity $w$ at demand node $i$ that is not fulfilled in scenario $k$ by the solution. This unmet demand is penalized per unit by $G_{iw}$, a parameter for the model which varies according to how strictly a demand constraint should be followed. We illustrate the use of the penalty term $G_{iw}t^k_{iw}$ in the formulation of SNDP-cont-wp below.

**[SNDP-cont-wp]:**

$$\min_{x,y,t} \quad \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{k\in K} p^k \sum_{w\in W} \left( \sum_{(i,j)\in A} a_{ijw} y^k_{ijw} + \sum_{i\in D_w} G_{iw} t^k_{iw} \right) \tag{2.28}$$

$$\text{s.t.} \quad \text{(2.23), (2.24), (2.25)}$$

$$- \sum_{j:(i,j)\in A} y^k_{ijw} + \sum_{j:(j,i)\in A} y^k_{jiw} + t^k_{iw} \geq d^k_{iw} \quad \forall i \in D_w,\ w \in W,\ k \in K \tag{2.29}$$

$$x \geq 0,\ y^k \geq 0,\ t^k \geq 0,\ \forall k \in K \tag{2.30}$$

We compute the unmet demand $t^k_{iw}$ in scenario $k \in K$ of commodity $w \in W$ at demand node $i \in D_w$ according to (2.29), and penalize positive demand losses at a unit cost $G_{iw}$ in the objective function (2.28). The model is typically used to formulate cost-based NDPs. We later use SNDP-cont-wp as a benchmark against which we compare our PNDP-cont reformulations in Section 2.5.

## 2.3 NDPs with Binary Design Variables

Sometimes, decisions need to be made on whether to build a link, instead of on the capacity of existing links in the network. In this section, we consider such problems by focusing on NDPs with binary capacity design variables, denoted by $\beta_{ij}$ for all $(i,j) \in A$. If $\beta_{ij} = 1$ we add the link $(i,j)$ with a fixed capacity $u_{ij} > 0$ and incur a cost of $q_{ij} > 0$ in the objective; otherwise arc $(i,j)$ is not constructed and does not exist in the network. We modify the previous formulations with fixed flow variables according to this new definition.

### 2.3.1 Fixed Flow Variables at the Upper Level

We modify the variants of PNDP-cont to reflect the restriction of capacity design variable to binary variables only. Here use PNDP-cont-joint as an example – the other models can be reformulated in the same manner. The aim is to minimize both arc construction cost and flow cost, subject to a certain probability guarantee for satisfying the overall demand. To modify PNDP-cont-joint to PNDP-bin-joint, we replace $c_{ij}x_{ij}$ in the objective by $q_{ij}\beta_{ij}$ to reflect the fixed cost of building arc $(i,j)$, and replace constraint (2.3) with $\sum_{w\in W} y_{ijw} \leq u_{ij}\beta_{ij}$ to reflect the fixed capacity allocated to arc $(i,j)$ if it is built, for all $(i,j) \in A$. All other constraints are kept the same in all MILP models involving binary capacity design variables.

**[PNDP-bin-joint]:**

$$\min_{\beta,y,z} \quad \sum_{(i,j)\in A} q_{ij}\beta_{ij} + \sum_{w\in W}\sum_{(i,j)\in A} a_{ijw}y_{ijw} \tag{2.31}$$

$$\text{s.t.} \quad \sum_{w\in W} y_{ijw} \leq u_{ij}\beta_{ij} \quad \forall (i,j) \in A \tag{2.32}$$

(2.4)–(2.8)

$$\beta \in \{0,1\}^{|A|},\ y \geq 0,\ z \in \{0,1\}^{|K|} \tag{2.33}$$

### 2.3.2 Recourse Flow Variables at the Lower Level

The formulations of SNDP-bin are again very similar to that of their SNDP-cont counterparts. The only differences are that $q_{ij}\beta_{ij}$ replaces $c_{ij}x_{ij}$ in the objective function, and that the recourse flow variables are bounded above by $u_{ij}\beta_{ij}$ instead of $x_{ij}$. Here we present the formulation of SNDP-bin-wp as an example for comparison with SNDP-cont-wp.

**[SNDP-bin-wp]:**

$$\min_{\beta,y,t} \quad \sum_{(i,j)\in A} q_{ij}\beta_{ij} + \sum_{k\in K} p^k \left( \sum_{w\in W} \sum_{(i,j)\in A} a_{ijw} y^k_{ijw} + \sum_{i\in D_w} G_{iw} t^k_{iw} \right) \tag{2.34}$$

$$\text{s.t.} \quad \sum_{w\in W} y^k_{ijw} \le u_{ij}\beta_{ij} \quad \forall (i,j)\in A,\ k\in K \tag{2.35}$$

$$(2.24)\text{–}(2.25)\ ;\ (2.29)$$

$$\beta \in \{0,1\}^{|A|},\ y^k \ge 0,\ t^k \ge 0,\ \forall (i,j)\in A,\ k\in K \tag{2.36}$$

### 2.3.3 Comparison of PNDP/SNDP-bin with PNDP/SNDP-cont

When comparing the PNDP/SNDP-bin formulations with their associated PNDP/SNDP-cont formulations, it can be seen that their only difference is in the network design decision variables. However, having binary variables makes PNDP/SNDP-bin formulations MILP, which are much harder to solve compared to the linear programs of the PNDP/SNDP-cont.

Binary variables are easy to branch on, allowing us to use a branch-and-bound approach for optimizing PNDP/SNDP-bin. Furthermore, only $\beta$ contains binary variables, so by using a Benders decomposition approach, we isolate the binary variables in the master problem, and solve the subproblem as a linear program. More algorithmic details of solving PNDP/SNDP-bin formulations are presented in Appendix A.

## 2.4 Algorithms for Optimizing NDPs

We first derive valid inequalities for solving PNDP-cont-joint, and then describe how to solve PNDP-cont-nc as well as special cases of PNDP-cont-c/n in polynomial time. We discuss the relationship between joint chance constraints and single-row chance constraints, which can be used to derive valid bounds in the branch-and-bound scheme. Finally, we describe a Benders decomposition algorithm to optimize the benchmark SNDP formulations.

### 2.4.1 Valid Inequalities for Optimizing MILP Reformulations of PNDPs

We first propose valid inequalities that can be generated to strengthen PNDP-cont-joint, which can be considered to be a more general case of its -nc/-c/-n variants. The concepts of *partial orders* and *induced cover sets* are given as follows based on (2.1) in PNDP-cont-joint. We refer the readers to Ruszczyński (2002) for general definitions of the two concepts.

**Definition 2.1.** Given any feasible $y$, a partial order $\preceq$ defined on scenarios $K$ satisfies

$$a \preceq b$$
$$\Leftrightarrow\ d^a_{iw} + \sum_{j:(i,j)\in A} y_{ijw} - \sum_{j:(j,i)\in A} y_{jiw} \leq d^b_{iw} + \sum_{j:(i,j)\in A} y_{ijw} - \sum_{j:(j,i)\in A} y_{jiw},\ \forall i \in D_w,\ w \in W.$$

□

Moreover, given that the node-arc incidence matrix is deterministic, and the uncertainty only exists in $d$, there exists a partial order $\preceq$ defined on scenarios $a, b \in K$, satisfying

$$a \preceq b \ \Leftrightarrow\ d^a_{iw} \leq d^b_{iw},\ \forall w \in W,\ i \in D_w.$$

For PNDP-cont-joint, there exists an optimal solution $(\hat{x}, \hat{y}, \hat{z})$, in which

$$a \preceq b \ \Rightarrow\ \hat{z}^a \leq \hat{z}^b \quad \forall a,\ b \in K,\ a \neq b.$$

This indicates that if the chance constraint is violated in scenario $a$, then it will necessarily be violated in scenario $b$ (i.e. $z^a = 1 \Rightarrow z^b = 1$). For all scenarios $k \in K$, define the set

$$L^k := \{k' \in K\ :\ k \preceq k'\},$$

as the collection of scenarios in which the chance constraint will be violated if it is violated in scenario $k$.

**Definition 2.2.** A set $C \subseteq K$ is called an induced cover if

$$\mathbb{P}\left\{\bigcup_{k\in C} L^k\right\} > \epsilon. \tag{2.37}$$

An induced cover set $C$ is minimal if for *every* $k' \in C$,

$$\mathbb{P}\left\{\bigcup_{k\in C} L^k \setminus \{k'\}\right\} \leq \epsilon. \tag{2.38}$$

□

For any induced cover set $C$, there exists an inequality of the form

$$\sum_{k\in C} z^k \leq |C| - 1, \tag{2.39}$$

which is valid to PNDP-cont-joint (see Nemhauser and Wolsey, 1988; Wolsey, 1998). To search for induced cover sets of constraint (2.1) based on Definition 2.2, one can apply MILP techniques as well as heuristics. Adding (2.39) to PNDP-cont-joint will potentially improve computational performance by tightening the formulation. However, for large scale problems when $|K|$ is large, the number of possible induced cover sets can be exponential. Similar to (2.39), one can derive cover inequalities for both PNDP-cont-c and PNDP-cont-n, by identifying the induced cover sets associated with each joint chance constraint in (2.11) and (2.12), respectively.

### 2.4.2 Polynomial Time Algorithms for PNDP-cont-nc/c/n

In this section, we explore polynomial-time algorithms for special cases of PNDPs with the presence of single-row chance constraints.

#### 2.4.2.1 A Polynomial-Time Algorithm for PNDP-cont-nc

As an alternative to solving PNDP-cont-nc as an MILP, we transform the constraints of PNDP-cont-nc into deterministic constraints without binary variables $z_{iw}$, resulting in a reformulation that is equivalent to solving multiple commodity-wise minimum-cost flow subproblems with revised arc costs. Note that each chance constraint in (2.10) only has a single row. According to Definition 2.1, we analogously define partial orders $\preceq_{iw}$ for every combination of $i$ and $w$ as follows.

**Definition 2.3.** Given any feasible $y$, and any $i \in D_w$ and $w \in W$, a partial order $\preceq_{iw}$ defined on scenarios $K$ satisfies

$$a \preceq_{iw} b \;\Leftrightarrow\; d^a_{iw} + \sum_{j:(i,j)\in A} y_{ijw} - \sum_{j:(j,i)\in A} y_{jiw} \leq d^b_{iw} + \sum_{j:(i,j)\in A} y_{ijw} - \sum_{j:(j,i)\in A} y_{jiw},$$

□

Similarly, define the set

$$L^k_{iw} := \{k' \in K \;:\; k \preceq_{iw} k'\},\ \forall k \in K,\ \forall w \in W,\ i \in D_w,$$

as the collection of scenarios in which the demand of commodity $w$ at node $i$ cannot be satisfied if there is demand lost in scenario $k \in K$. Let $\mathcal{P}(L^k_{iw}) := \sum_{k' \in L^k_{iw}} p^{k'}$, indicating the *minimal* violation probability of the $(i,w)^{\text{th}}$ chance constraint, given that it has been violated in scenario $k$. The goal, for all $w \in W$ and $i \in D_w$, is to identify threshold values

of the violation probability, such that it yields the same optimal solution by assigning $\epsilon_{iw}$ any values from the same interval between neighboring thresholds.

For two consecutive scenarios $k'$ and $k$ in a partial order $\preceq_{iw}$, suppose that $d_{iw}^{k'} < d_{iw}^{k}$. Given some flow solution $y$, suppose that

$$-\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} < d_{iw}^{k} \tag{2.40}$$

$$\text{and} \quad -\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} \geq d_{iw}^{k'}. \tag{2.41}$$

Then, the probability of $-\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} < d_{iw}$ must be $\mathcal{P}(L_{iw}^{k})$. The definitions are demonstrated on a small example as follows.

**Example 2.1.** We focus on one dimension of the demand vector $d$, and assume that $d_{iw}$ is a random scalar from the realizations $\{d_{iw}^{k}\}_{k\in K} = \{1, 4, 6, 7, 10, 11\}$, corresponding to the numbered scenarios $K = \{1, 2, 3, 4, 5, 6\}$ and probabilities $0.1,\ 0.1,\ 0.2,\ 0.3,\ 0.2$, and $0.1$, respectively. According to Definition 2.3, the current order is already a partial order $\preceq_{iw}$, and we obtain

$$L_{iw}^{1} = \{1, 2, 3, 4, 5, 6\},\ L_{iw}^{2} = \{2, 3, 4, 5, 6\},\ L_{iw}^{3} = \{3, 4, 5, 6\},$$
$$L_{iw}^{4} = \{4, 5, 6\},\ L_{iw}^{5} = \{5, 6\},\ L_{iw}^{6} = \{6\}.$$

If $-\sum_{j:(i,j)\in A} y_{ijw} + \sum_{j:(j,i)\in A} y_{jiw} < d_{iw}^{1} = 1$, then the chance constraint is violated in all scenarios, and $\mathcal{P}(L_{iw}^{1}) = 1$. Similarly, we have $\mathcal{P}(L_{iw}^{2}) = 0.9$, $\mathcal{P}(L_{iw}^{3}) = 0.8$, $\mathcal{P}(L_{iw}^{4}) = 0.6$, $\mathcal{P}(L_{iw}^{5}) = 0.3$, $\mathcal{P}(L_{iw}^{6}) = 0.1$. □

For all combinations of $i$ and $w$, we sort all scenarios, yielding an ordered set $\mathcal{O}(i, w) = \{k_1, \ldots, k_{|K|}\}$ such that $d_{iw}^{k_1} \leq d_{iw}^{k_2} \leq \cdots \leq d_{iw}^{k_{|K|}}$. Note that the order of scenarios $k_1, \ldots, k_{|K|}$ in each $\mathcal{O}(i, w)$ may be different for different $i$ and $w$ by considering independent chance constraints for individual $(i, w)$-pairs. For notational brevity, we denote $K = \{1, \ldots, |K|\}$ as the reassembled set of scenarios in $\mathcal{O}(i, w)$, which may contain different specific scenario sequences for different pairs of $i$ and $w$. Therefore, for all $w \in W$ and $i \in D_w$, a partial order $\preceq_{iw}$ is given by

$$L_{iw}^{1} \supseteq L_{iw}^{2} \supseteq \cdots \supseteq L_{iw}^{|K|}, \text{ and thus } 1 = \mathcal{P}(L_{iw}^{1}) \geq \mathcal{P}(L_{iw}^{2}) \geq \cdots \geq \mathcal{P}(L_{iw}^{|K|}) \geq 0.$$

Moreover, we denote $\mathcal{P}(L_{iw}^{|K|+1}) = 0$, combine identical values of $d_{iw}^{k}$, $\forall k \in K$, and associate each new scenario with a cumulative probability. This pre-processing procedure may decrease the number of scenarios $|K|$. We continue to use $K = \{1, \ldots, |K|\}$, but now

assume that each scenario $k \in K$ has a unique $d_{iw}^k,\ \forall w,\ i$, and $p^k > 0$. Thus,

$$L_{iw}^1 \supset L_{iw}^2 \supset \cdots \supset L_{iw}^{|K|}, \text{ and } 1 = \mathcal{P}(L_{iw}^1) > \mathcal{P}(L_{iw}^2) > \cdots > \mathcal{P}(L_{iw}^{|K|}) > \mathcal{P}(L_{iw}^{|K|+1}) = 0.$$

We now establish the equivalence of a single-row chance constraint and a deterministic constraint.

**Theorem 2.1.** If $\epsilon_{iw} \in [0, 1)$, the $(i, w)^{\text{th}}$ chance constraint (2.10) is equivalent to the deterministic constraint

$$\sum_{j:(j,i)\in A} y_{jiw} - \sum_{j:(i,j)\in A} y_{ijw} \geq d_{iw}^{\widehat{k}}, \tag{2.42}$$

where scenario $\widehat{k} \in \{1, \ldots, |K|\}$ is such that $\mathcal{P}(L_{iw}^{\widehat{k}}) > \epsilon_{iw} \geq \mathcal{P}(L_{iw}^{\widehat{k}+1})$. If $\epsilon_{iw} = 1$, the chance constraint (2.10) is relaxed. □

*Proof.* Given any $\epsilon_{iw} \in [0, 1)$, we have $\mathcal{P}(L_{iw}^{\widehat{k}}) > \epsilon_{iw} \geq \mathcal{P}(L_{iw}^{\widehat{k}+1})$ for some $\widehat{k} \in \{1, \ldots, |K|\}$ due to the pre-processing procedure, which results in a strict ordering of scenarios and $\mathcal{P}(L_{iw}^k) \in [0, 1)$ for all $k \in \{1, \ldots, |K|\}$.

Suppose chance constraint (2.10) holds true, so the maximum chance constraint violation probability is $\epsilon_{iw}$. If we assume that constraint (2.42)is violated, then the minimum chance constraint violation probability is $\mathcal{P}(L_{iw}^{\widehat{k}}) > \epsilon_{iw}$ by the definition of $\mathcal{P}(L_{iw}^k)$. This is a contradiction, so constraint (2.42) must hold true.

Now suppose constraint (2.42) holds true. Since scenarios are strictly ordered, we have $\sum_{j:(j,i)\in A} y_{jiw} - \sum_{j:(i,j)\in A} y_{ijw} \geq d_{iw}^k$ for all $k \in \{\widehat{k}, \widehat{k}-1, \ldots, 1\}$. The strict ordering also implies that $K \backslash L_{iw}^{\widehat{k}+1} = \{\widehat{k}, \widehat{k}-1, \ldots, 1\}$. Hence, the chance constraint is satisfied with probability

$$\sum_{k\in\{\widehat{k},\widehat{k}-1,\ldots,1\}} p^k = \sum_{k\in K\backslash L_{iw}^{\widehat{k}+1}} p^k = 1 - \sum_{k\in L_{iw}^{\widehat{k}+1}} p^k = 1 - \mathcal{P}(L_{iw}^{\widehat{k}+1}) \geq 1 - \epsilon_{iw}.$$

Therefore, chance constraint (2.10) is equivalent to the deterministic constraint (2.42). When $\epsilon_{iw} = 1$, the chance constraint is satisfied by any feasible $y$ due to the non-negativity of any probability value. This completes the proof. □ □

We can compare $\epsilon_{iw}$ with adjacent values of $\mathcal{P}(L_{iw}^k)$ and $\mathcal{P}(L_{iw}^{k+1})$, for all $k = 1, \ldots, |K|$, and allocate $\epsilon_{iw}$ to an appropriate interval. By applying Theorem 2.1, we transform the chance constraint into a deterministic constraint. As all chance constraints for every combination of $w$ and $i$ are independently stated, we repeat the procedures for each $w \in W$ and

$i \in D_w$, and solve PNDP-cont-nc as a deterministic formulation. Algorithm 2.1 describes this procedure in greater detail.

**Algorithm 2.1** Polynomial-time algorithm for optimizing PNDP-cont-nc

1: **for** $w \in W$ **do**
2:  **for** $i \in D_w$ **do**
3:  Sort $d^k_{iw}$ in an ascending order, combine identical values, and re-arrange the scenario number.
4:  Set $\{1, \ldots, |K|\}_{iw}$ as the new partial order of the $(i, w)^{\text{th}}$ constraint (2.10)
5:  **for** $k \in \{1, \ldots, |K|\}_{iw}$ **do**
6:  Define

$$L^k_{iw} := \{k, k+1, \ldots, |K|\} \text{ and } \mathcal{P}(L^k_{iw}) := \sum_{k'=k}^{|K|} p^{k'}.$$

7:  **end for**
8:  Identify $\widehat{k} \in \{1, \ldots, |K|\}$ such that $\mathcal{P}(L^{\widehat{k}}_{iw}) > \epsilon_{iw} \geq \mathcal{P}(L^{\widehat{k}+1}_{iw})$.
9:  Replace $(i, w)^{\text{th}}$ chance constraint (2.10) with deterministic constraint (2.42).
10:  **end for**
11: **end for**
12: Solve PNDP-cont-nc as

$$\min \left\{ \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{w\in W} \sum_{(i,j)\in A} a_{ijw}y_{ijw} : \text{(2.3)–(2.5),(2.42)} \right\}.$$

13: **return** Optimal solution $(x^*, y^*)$.

#### 2.4.2.2 Complexity Analysis of Algorithm

In Algorithm 2.1, the transformation steps before Step 12 are polynomial in $|K|$. Thus, the algorithmic complexity is mainly determined by

$$\min_{x,y} \left\{ \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{w\in W} \sum_{(i,j)\in A} a_{ijw}y_{ijw} : \text{(2.3)–(2.5), (2.42)} \right\},$$

which is equivalent of solving $|W|$ minimum cost flow problems for all commodities, by using $(c_{ij} + a_{ijw})$ as a revised arc cost for every arc $(i, j)$ with respect to commodity $w$, for all $w \in W$. The arguments are given as follows. By assumption, let $c_{ij} \geq 0,\ \forall (i, j) \in A$ and $a_{ijw} \geq 0,\ \forall (i, j) \in A,\ w \in W$. Multiplying constraints (2.3) by $c_{ij}$ gives

$$\sum_{w\in W} c_{ij}y_{ijw} \leq c_{ij}x_{ij},\ \forall (i, j) \in A. \tag{2.43}$$

We add up the above terms over all arcs $(i,j) \in A$, and also add $\sum_{w\in W}\sum_{(i,j)\in A} a_{ijw}y_{ijw}$ to both sides, yielding

$$\sum_{w\in W}\sum_{(i,j)\in A}(c_{ij}+a_{ijw})y_{ijw} \leq \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{w\in W}\sum_{(i,j)\in A} a_{ijw}y_{ijw}. \tag{2.44}$$

Due to the minimizing nature of the objective, the original problem is then equivalent to

$$\min_{y}\left\{\sum_{w\in W}\sum_{(i,j)\in A}(c_{ij}+a_{ijw})y_{ijw} : (2.4),(2.5),\ (2.42)\right\},$$

which only involves variables $y$. PNDP-cont-nc is then equivalent of solving $|W|$ minimum cost flow problems with revised arc costs, each of which involves commodity-based constraints (2.4), (2.5), and (2.42). The overall computational complexity is linear in the complexity of algorithms used for sorting and for solving minimum-cost flow problems.

#### 2.4.2.3 Algorithm demonstration

We demonstrate our approaches on an example of PNDP-cont-nc which contains three single-row chance constraints, and is formulated on a network depicted in Figure 2.1. Three commodities are shipped from nodes 0, 1, and 2, respectively, all to demand node 4, with supply capacities given as $s_{01} = s_{12} = s_{23} = 10$. Demand $(d_{41}, d_{42}, d_{43})$ is jointly realized from a set

$$\{d^k\}_{k\in K} = \{(3,1,10),(4,3,9),(5,5,8),(6,7,7),(7,8,6),(8,6,5),(9,4,4),(10,2,3)\}$$

with an equal probability of $1/8$ for each realization. The costs are $c_{01} = c_{02} = c_{24} = 1$, $c_{34} = c_{13} = c_{32} = 2$, and for commodities $w = 1, 2, 3$, $a_{ijw} = 0.5,\ 0.2,\ 0.3$ respectively on all arcs $(i,j) \in A$. The risk levels for unsatisfied demand for commodities 1, 2, and 3 are mandated as $\epsilon_1 = 0.2$, $\epsilon_2 = 0.4$, and $\epsilon_3 = 0.3$, respectively.

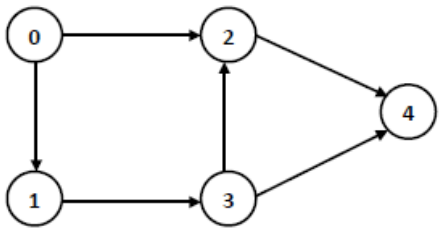



Figure 2.1: Network topology for PNDP-cont-nc example

Different from previously presented Algorithm 2.1, here we directly compute all values of $L_w^k$ based on the definition, without the pre-sorting steps. For commodities 1, 2, and 3,

$$\begin{aligned}
&L_1^1 = \{1,2,\ldots,8\},\ L_1^2 = \{2,\ldots,8\},\ldots,\ L_1^7 = \{7,8\},\ L_1^8 = \{8\},\\
&L_2^1 = \{1,2,\ldots,8\},\ L_2^2 = \{2,\ldots,7\},\ L_2^3 = \{3,\ldots,6\},\ L_2^4 = \{4,5\},\\
&\qquad\qquad L_2^5 = \{5\},\ L_2^6 = \{6,5,4\},\ L_2^7 = \{7,\ldots,3\},\ L_2^8 = \{8,\ldots,2\},\\
&L_3^1 = \{1\},\ L_3^2 = \{1,2\},\ldots,\ L_3^8 = \{1,\ldots,8\}.
\end{aligned}$$

where $L_w^k$ are abbreviations of $L_{iw}^k$ since $i = 4$ is the only demand node, for all $w$. The associated probability thresholds are $\mathcal{P}(L_w^k) = 0.125|L_w^k|$ for all sets $L_w^k$. We examine all scenarios in ascending order of $\mathcal{P}(L_w^k)$, (i.e. in increasing cardinality of $L_w^k$, for all $w = 1, 2, 3$). Subsequently, we allocate $\epsilon_1$, $\epsilon_2$, and $\epsilon_3$ in the following intervals

$$\begin{aligned}
&\mathcal{P}(L_1^7) = 0.25 > \epsilon_1 = 0.2 \geq \mathcal{P}(L_1^8) = 0.125 \ \Rightarrow\ \widehat{k}_1 = 7 \text{ for commodity 1,}\\
&\mathcal{P}(L_2^3) = 0.5 > \epsilon_2 = 0.4 \geq \mathcal{P}(L_2^6) = 0.375 \ \Rightarrow\ \widehat{k}_2 = 3 \text{ for commodity 2,}\\
&\mathcal{P}(L_3^3) = 0.375 > \epsilon_3 = 0.3 \geq \mathcal{P}(L_3^2) = 0.25 \ \Rightarrow\ \widehat{k}_3 = 3 \text{ for commodity 3,}
\end{aligned}$$

which allows the transformation of the three chance constraints into the following deterministic inequalities.

$$\begin{aligned}
y_{241} + y_{341} &\geq d_{41}^7 = 9\\
y_{242} + y_{342} &\geq d_{42}^3 = 5\\
y_{243} + y_{343} &\geq d_{43}^3 = 8.
\end{aligned}$$

Alternatively, one can formulate the problem by defining binary variables $z_{4w}^k$ taking value 1 if the $w^{\text{th}}$ chance constraint violates scenario $k$ at the demand node 4. According to Definition 2.2, $\{L_1^7\}$, $\{L_2^3\}$, and $\{L_3^3\}$ are minimal cover sets for each commodity-based chance constraint, respectively. Generating the corresponding cover inequalities will lead to the same solution:

$$z_{41}^7 \leq 0 \Rightarrow d_{41} \geq d_{41}^7 = 9,\ z_{42}^3 \leq 0 \Rightarrow d_{42} \geq d_{42}^3 = 5,\ \text{and } z_{43}^3 \leq 0 \Rightarrow d_{43} \geq d_{43}^3 = 8,$$

The original problem is then equivalent to solving three shortest path problems, which aim to transport deterministic demands (i.e. 9, 5, 8 units of commodities $1, 2$, and 3, respectively) from a single supply node to demand node 4, with $(c_{ij} + a_{ijw})$ being the flow cost for all $(i, j) \in A$ for commodity $w$. Given that paths 0–2–4, 1–3–4, and 2–4 are the shortest paths, an optimal solution is $x_{02}^* = 9$, $x_{24}^* = 9 + 8 = 17$, $x_{13}^* = x_{34}^* = 5$, yielding the

minimum cost of $61.4$.

#### 2.4.2.4 Polynomial time algorithms for special cases of PNDP-cont-c/n

The following discussion of special cases and the development of algorithms can be applied to both PNDP-cont-c and PNDP-cont-c/n. Using PNDP-cont-c as an example, when $|D_w| = 1$ for all $w \in W$, we then have a single-row chance constraint for every commodity $w$. The special case corresponds to real-world applications in which only one demand node exists for each commodity, who can be satisfied by one or multiple suppliers. Recall that with single-row chance constraints, we can reformulate PNDP-cont-c as a deterministic problem via the enumeration of violation risk thresholds, for all commodities $w \in W$.

We apply similar approaches as Algorithm $2.1$ by considering one demand node $i_w$ for each commodity $w$, and revise the for-loop from Step 3 through Step 6 as follows. With respect to each $w \in W$, we identify scenario $\widehat{k}$, such that $\mathcal{P}(L_w^{\widehat{k}}) > \epsilon_w \geq \mathcal{P}(L_w^{\widehat{k}+1})$, and replace the chance constraint $(2.11)$ with

$$\sum_{j:(j,i)\in A} y_{ji_ww} - \sum_{j:(i,j)\in A} y_{i_wjw} \geq d_{i_ww}^{\widehat{k}} \quad \forall w \in W. \tag{2.45}$$

PNDP-cont-c is then equivalent to solving

$$\min_{x,y} \left\{ \sum_{(i,j)\in A} c_{ij}x_{ij} + \sum_{w\in W} \sum_{(i,j)\in A} a_{ijw}y_{ijw} : (2.3)\text{–}(2.5),(2.45) \right\},$$

which is further equivalent to

$$\min_{y} \left\{ \sum_{w\in W} \sum_{(i,j)\in A} (c_{ij} + a_{ijw})y_{ijw} : (2.4),\ (2.5),(2.45) \right\}. \tag{2.46}$$

Moreover, the algorithm employs the shortest-path algorithm rather than solve $W$ minimum cost flow problems, because of the single-demand-node assumption for each commodity. The following elaborates on the special cases of "one supply, one demand" and "multiple supplies, one demand".

**Case I:** $|S_w| = |D_w| = 1,\ \forall w$**:** The algorithm seeks a shortest paths for each supply and demand node pair, and solves $W$ shortest-path problems, by setting arc cost as $(c_{ij} + a_{ijw})$ for all $(i, j) \in A$ and $w \in W$. Dijkstra's algorithm finds each shortest path in $O(|N|^2)$ time by noting that all revised costs are nonnegative. For every commodity $w$, $d_{i_ww}^{\widehat{k}}$ units of flow

are transported on arcs of the corresponding path, yielding the optimal objective value of Formulation (2.46) as

$$\sum_{w \in W} d^{\widehat{k}}_{i_w w} \left( \sum_{(i,j) \in \mathbb{S}_w} (c_{ij} + a_{ijw}) \right),$$

where $\mathbb{S}_w$ is a shortest path identified for commodity $w$.

**Case II: $|S_w| > 1$ and $|D_w| = 1,\ \forall w$:** When having multiple suppliers, we first compute the shortest paths from each supplier in $S_w$ to the demand node $i_w$, and start from the "cheapest" supplier to flow as much as its capacity allows. We greedily repeat the "capacity-saturating" procedure for each ordered supply node, until $d^{\widehat{k}}_{i_w w}$ units of required demand are all satisfied. The details are given in Algorithm 2.2.

**Algorithm 2.2** Greedy Algorithm to Solve Problem (2.46) in Case II

**for** $w \in W$ **do**
  **for** $j \in S_w$ **do**
    Use Dijkstra's algorithm to find a shortest path from node $j$ to the singleton demand node $i_w$, denoted as $\mathbb{S}(j, i_w)$.
  **end for**
  Order all supply nodes in $S_w$ such that the shortest distances are ordered $\ell_{\mathbb{S}(1,i_w)} \leq \ell_{\mathbb{S}(2,i_w)} \leq \cdots \leq \ell_{\mathbb{S}(|S_w|,i_w)}$.
  Denote $e_w$ as the current unsatisfied demand and set $e_w = d^{\widehat{k}}_{i_w w}$.
  Denote $m$ as the supplier number under examination and set $m = 1$.
  **while** $e_w > 0$ **do**
    **if** $s_m \geq e_w$ **then**
      Flow $e_w$ on the shortest path $\mathbb{S}(m, i_w)$.
      $e_w \leftarrow 0$
    **else**
      Flow $s_m$ on the shortest path $\mathbb{S}(m, i_w)$.
      $e_w \leftarrow e_w - s_m$
      $m \leftarrow m + 1$
    **end if**
  **end while**
**end for**
**return** Optimal $y^*_w$ as flows on shortest paths, optimal $x^*_{ij} = \sum_{w \in W} y^*_{ijw},\ \forall (i,j) \in A$.

### 2.4.3 Relationship Between the PNDP-cont Models

Among all general PNDP-cont models we have discussed so far, only PNDP-cont-nc can be quickly solved by Algorithm 2.1. For PNDP-cont-joint, PNDP-cont-c, and PNDP-cont-n,

we need to solve a deterministic MILP model (involving binary $z$). The cover inequalities can be used to potentially improve computational efficacy. Next we describe how to use optimal solutions of specially designed PNDP-cont-nc for computing upper bounds of the objectives of the other models. These bounds can be incorporated into branch-and-bound nodes, together with lower bounds obtained by solving linear programming relaxations of the original MILP models.

Consider PNDP-cont-joint, which contains a joint chance constraint with a risk tolerance $\epsilon$. Assume that demand distributions are independent among all $w \in W$ and $i \in D_w$. Let $E_{iw}$ represent the event of no demand loss at node $i \in D_w$ for commodity $w \in W$, so $\sum_{j:(j,i)\in A} y_{jiw} - \sum_{j:(i,j)\in A} y_{ijw} \geq d_{iw}$. We have

$$\mathbb{P}(E_{iw},\ \forall w \in W,\ i \in D_w) = \prod_{w\in W,\ i\in D_w} \mathbb{P}(E_{iw}). \tag{2.47}$$

Given risk tolerances $\epsilon_{iw}$ for all combinations of $i$ and $w$, let $(x^*, y^*)$ be an optimal solution to PNDP-cont-nc. Note that

$$\prod_{w\in W,\ i\in D_w} \mathbb{P}(E_{iw}(x^*, y^*)) \geq \prod_{w\in W,\ i\in D_w} (1-\epsilon_{iw}) \geq 1 - \sum_{w\in W,\ i\in D_w} \epsilon_{iw}, \tag{2.48}$$

where the last inequality is due to $\epsilon_{iw} \in [0, 1)$ for all $i$ and $w$. Based on (2.47) and (2.48), $(x^*, y^*)$ is a feasible solution to PNDP-cont-joint by letting $\epsilon = \sum_{w\in W,\ i\in D_w} \epsilon_{iw}$. As a result, for PNDP-cont-joint with $\epsilon$, one can design $\epsilon_{iw}$ in PNDP-cont-nc with $\sum_{w\in W,\ i\in D_w} \epsilon_{iw} = \epsilon$, to ensure the aforementioned relationship holds. We then solve the designed PNDP-cont-nc in polynomial time to obtain a feasible solution to PNDP-cont-joint, which also yields an upper bound for the optimal objective. Tighter bounds can be generated by varying $\epsilon_{iw},\ \forall i$ and $w$, while making sure $\sum_{w\in W,\ i\in D_w} \epsilon_{iw} \geq \epsilon$.

Similarly, each joint chance constraint in PNDP-cont-c and PNDP-cont-n can be approximated by a series of single-row chance constraints in PNDP-cont-nc, while respectively ensuring

$$\epsilon_w = \sum_{i\in D_w} \epsilon_{iw},\ \text{and}\ \epsilon_i = \sum_{w\in W} \epsilon_{iw}. \tag{2.49}$$

An optimal solution to PNDP-cont-nc is then feasible to PNDP-cont-c and PNDP-cont-n, and provides a valid upper bound for the corresponding optimal objective values.

For computing the benchmark SNDPs described in this chapter, we employ a decomposition algorithm, which follows standard Benders procedures. We demonstrate the details of decomposition and cutting-plane generation in Appendix A. The approach is alternative to solving the MILPs via off-the-shelf solvers, and can be also generalized for solving the

deterministic MILP reformulations of the PNDP-bin-nc.

## 2.5 Computational Results

We test our models and algorithms on two sets of numerical instances based on randomly generated networks and the Sioux-Falls network, respectively. Section 2.5.1 focuses on the comparison between joint and single-row chance constraints, as well as comparison of different algorithms for solving PNDP-cont, by testing moderate-size network instances. Section 2.5.2 computes representative models of PNDP and SNDP on instances of a real-world network. We aim to derive managerial insights of implementing various continuous/discrete PNDP and SNDP formulations under different demand/supply situations.

### 2.5.1 Randomly Generated Networks

#### 2.5.1.1 Experimental Setup

Here we generate random network instances that have sizes of $|N| = 20$ and 30, with a density (defined as $\frac{|A|}{|N|\times|N|}$) being approximately 25%. Each instance has three commodities. We test $|K| = 50, 100, 200$ scenarios, and let $|S_w| = 4$ and $|D_w| = 2$ for all commodities $w = 1, 2, 3$. We test the four PNDP-cont models with fixed flow variables. The aim is to compare QoS results yielded by different models and demonstrate the efficacy of implementing Algorithm 2.1 compared with directly solving the MILP models.

For all arcs $(i, j)$, we assign $c_{ij}$ an integer obtained from rounding up a random number generated from a uniform distribution over the interval $(0, 8]$. For all arcs $(i, j) \in A$, we randomly generate $a_{ijw}$ from uniform distributions over the intervals $[0.1, 0.3]$, $[0.1, 0.4]$, and $[0.2, 0.4]$ for $w$ =1, 2, and 3, respectively. For every scenario $k \in K$, we first generate a random number from $1, \ldots, |K|$. The probability $p^k$ is then computed by dividing the random number by the sum of all random numbers generated for each scenario, such that $\sum_{k\in K} p^k = 1$ is enforced. We randomly select nodes in $N$ to be supply nodes in $S_w$ or demand nodes in $D_w$ for all $w \in W$. Finally, for $w = 1, 2, 3$ we generate the amounts of supply/demand from uniform distributions depicted in Table 2.1. The designated distributions also guarantee that $\sum_{i\in S_w} s_{iw} \geq \sum_{j\in D_w} d_{jw}$, $\forall w \in W$. Thus, given any possible data realizations from Table 2.1, we always have feasible solutions.

All models and algorithms use CPLEX 12.3 via ILOG Concert Technology with C++, and computations are performed on a HP Workstation z400 Windows 7 machine with Intel(R) Xeon(R) CPU E31230 3.20 GHz, and 8GB memory. The CPU time is reported in seconds.

Table 2.1: Distributions for generating supply/demand at corresponding locations

| Commodity | Supply 1, 2 | Supply 3, 4 | Demand 1, 2 |
|---|---|---|---|
| $w = 1$ | U(20,30) | U(40,50) | U(50,60) |
| $w = 2$ | U(10,15) | U(20,25) | U(25,30) |
| $w = 3$ | U(5,7) | U(10,12) | U(12,15) |

#### 2.5.1.2 Results Summary

We test models of PNDP-cont-joint/nc to demonstrate the relationship between joint and single-row chance constraints. For the single-row chance constraints for $w = 1, 2, 3$, we assume two cases having homogeneous (denoted by "-ho") and heterogeneous (denoted by "-he") risk tolerances among different chance constraints. Corresponding to each case of $\epsilon$, the homogeneous case uses $\epsilon_{11} = \epsilon_{12} = \epsilon_{21} = \epsilon_{22} = \epsilon_{31} = \epsilon_{32} = \epsilon/6$ when $|D_w| = 2$. When $|D_w| = 1,\ \forall w = 1, 2, 3$, the heterogeneous case uses $\epsilon_{11} = \epsilon_{12} = 1/12, \epsilon_{21} = \epsilon_{22} = 1/6$, and $\epsilon_{31} = \epsilon_{32} = 1/4$. The later setting follows an intuition that we aim to guarantee higher QoS levels for satisfying demands with higher variations (as indicated in Table 1). For each combination, we test fifteen instances and compute the averages to report.

Table 2.2 and Table 2.3 respectively report the CPU time and optimal objective values of the three PNDP MILP models tested. We use $\epsilon = 0.03$, $0.06$, $0.15$, and $0.3$, indicated in Column $\epsilon$, which guarantee the probability of no demand loss for any commodity at any node being no less than $97\%$, $94\%$, $85\%$, and $70\%$, respectively.

Table 2.2: CPU time of the PNDP chance-constrained models (in seconds)

| | $\epsilon$ | PNDP-cont-joint | | | PNDP-cont-nc-ho | | | PNDP-cont-nc-he | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $\|K\| = 50$ | $\|K\| = 100$ | $\|K\| = 200$ | $\|K\| = 50$ | $\|K\| = 100$ | $\|K\| = 200$ | $\|K\| = 50$ | $\|K\| = 100$ | $\|K\| = 200$ |
| $\|N\| = 20$ | 0.03 | 0.049 | 0.189 | 0.767 | 0.022 | 0.079 | 0.168 | 0.024 | 0.045 | 0.164 |
| | 0.06 | 0.170 | 0.395 | 3.207 | 0.022 | 0.127 | 0.472 | 0.019 | 0.069 | 0.472 |
| | 0.15 | 0.507 | 2.218 | 41.311 | 0.061 | 0.455 | 3.381 | 0.082 | 0.371 | 2.951 |
| | 0.3 | 1.179 | 7.046 | 1589.000 | 0.258 | 0.483 | 5.567 | 0.209 | 0.539 | 5.139 |
| $\|N\| = 30$ | 0.03 | 0.048 | 0.250 | 0.873 | 0.028 | 0.123 | 0.266 | 0.029 | 0.076 | 0.212 |
| | 0.06 | 0.256 | 0.509 | 3.875 | 0.028 | 0.161 | 0.672 | 0.028 | 0.127 | 0.428 |
| | 0.15 | 0.595 | 2.646 | 91.265 | 0.045 | 0.567 | 4.947 | 0.144 | 0.446 | 4.096 |
| | 0.3 | 1.738 | 8.929 | 739.115 | 0.452 | 0.640 | 7.728 | 0.334 | 0.831 | 7.929 |

For PNDP-cont-nc, we solve all instances with Algorithm 2.1 and the MILP approach, and depict the CPU time comparison in Figure 2.2. The CPU time of the MILP approach on the heterogeneous 200-scenario instances is used as the benchmark, being the pair of approach and instances requiring the most CPU time. The CPU time of the other cases is then compared as a percentage of this benchmark.

Table 2.3: Optimal objective values of the PNDP chance-constrained models

| | $\epsilon$ | PNDP-cont-joint | | | PNDP-cont-nc-ho | | | PNDP-cont-nc-he | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | $\|K\| = 50$ | $\|K\| = 100$ | $\|K\| = 200$ | $\|K\| = 50$ | $\|K\| = 100$ | $\|K\| = 200$ | $\|K\| = 50$ | $\|K\| = 100$ | $\|K\| = 200$ |
| $\|N\| = 20$ | 0.03 | 781.93 | 766.87 | 868.07 | 781.93 | 766.87 | 868.07 | 781.93 | 766.87 | 868.07 |
| | 0.06 | 763.20 | 748.20 | 850.89 | 781.93 | 757.65 | 861.64 | 763.20 | 748.20 | 850.89 |
| | 0.15 | 674.73 | 659.81 | 761.93 | 724.65 | 691.79 | 784.59 | 674.73 | 659.81 | 761.93 |
| | 0.3 | 663.94 | 648.84 | 750.92 | 717.99 | 684.55 | 779.52 | 663.94 | 648.84 | 750.92 |
| $\|N\| = 30$ | 0.03 | 682.07 | 692.80 | 650.47 | 682.07 | 693.13 | 650.47 | 682.07 | 692.80 | 650.47 |
| | 0.06 | 664.53 | 679.40 | 636.60 | 682.07 | 690.27 | 643.71 | 664.53 | 679.40 | 636.60 |
| | 0.15 | 574.60 | 586.74 | 545.13 | 664.32 | 619.01 | 571.46 | 574.60 | 586.74 | 545.13 |
| | 0.3 | 564.17 | 579.68 | 536.05 | 663.91 | 617.63 | 565.69 | 564.17 | 579.68 | 536.05 |

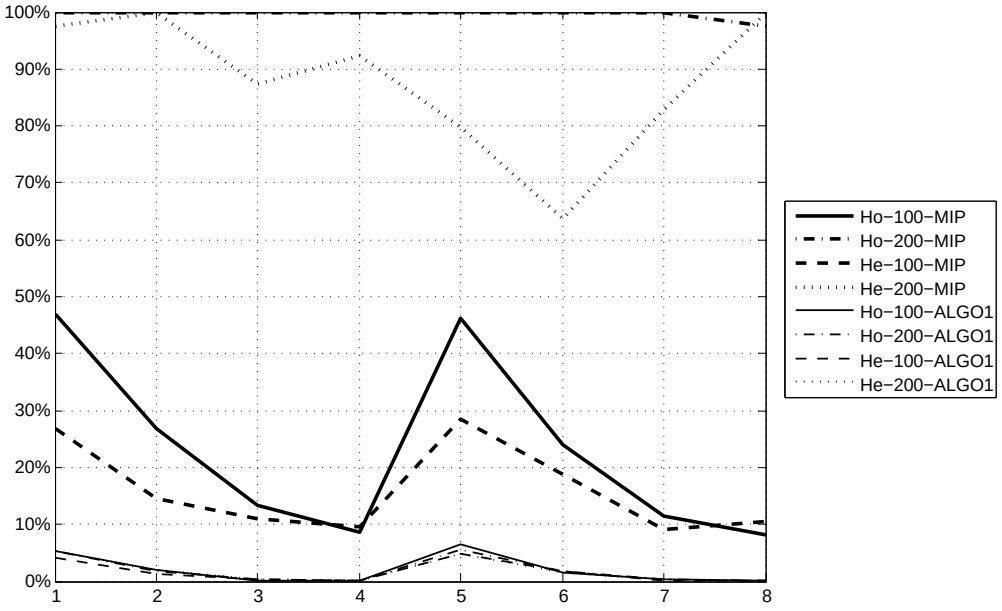



Figure 2.2: Percentage comparison of CPU time taken by Algorithm 2.1 and the MILP approach

We list our observations based on the computational results as follows.

- The optimal objective values decrease as we increase $\epsilon$ (i.e. by allowing more unmet demands and lower QoS levels). Also, PNDP-cont-nc is less sensitive to changes in $\epsilon$ compared to PNDP-cont-joint. This is because the risk is divided and distributed into several chance constraints, and thus the same change in $\epsilon$ will result in relatively smaller changes in $\epsilon_i$ for each constraint $i$.

- As described in Section 2.4.3, optimal solutions to PNDP-cont-nc will serve as feasible solutions to PNDP-cont-joint when $\epsilon \leq \sum_{w \in W, i \in D_w} \epsilon_{iw}$. Such (upper) bounds in general become tighter when $\epsilon$ is small. (In particular, when $\epsilon = 0.03$, optimal solutions to PNDP-cont-nc are also optimal to PNDP-cont-joint.) The bounds get much worse after we lower the QoS level from $94\%$ to $85\%$ (i.e. increasing $\epsilon$ from

$0.06$ to $0.15$). The heterogeneous risk setting in general yields better bounds than the homogeneous case, indicating the importance and necessity of differentiating risk tolerances (or QoS levels) for different customers and commodities. In general, all bounds become tighter in both tables when $|K|$ increases.

- For all MILP models tested, the CPU time dramatically increases as we increase $\epsilon$ (i.e. when allowing higher probabilities of violating chance constraints). We have much longer computational time spent on solving the MILPs compared with Algorithm 2.1, which also dramatically increases as we increase $|K|$.

- Solving the MILP model of PNDP-cont-nc is significantly faster than solving the MILP model of PNDP-cont-joint. In particular, the CPU time taken by Algorithm 2.1 is almost the same for all instances, regardless of changes to (i) the number of scenarios $|K|$, (ii) the sum of risk tolerances $\epsilon$, and (iii) homogeneous or heterogeneous risk settings. This is consistent with the observation that the complexity of Algorithm 2.1 is not determined by the number of scenarios but by the complexity of solving $|W|$ minimum-cost-flow problems.

## 2.5.2 Sioux-Falls Network

### 2.5.2.1 Experimental Setup

In this experiment, we use the Sioux Falls road network (LeBlanc et al., 1975), as shown in Figure 2.3, which is widely used in transportation literature. This network consists of 24 nodes and 76 links. We continue to use three commodities and $|K| = 100$ scenarios in instances created based on this network. The aim in this experiment is to glean managerial insights through the use and comparison of PNDP and SNDP models on an instance that is closer to a real-world instance.

We simulate a high inflow instance, where commodities flow into the network exclusively from the outer nodes of the network, with higher mean demands for nodes that are more centralized in the network. We select nodes 1, 2, 12, 13, 18, and 20 as the supply nodes (bold nodes in Figure 2.3), as these are the most likely entry points into Sioux Falls, and select the inner nodes 4, 5, 8–11, 14–17, 19, 22, and 23 as the demand nodes (shaded nodes), as these are the more populated areas in Sioux Falls.

For all arcs $(i, j)$, the travel distances $c_{ij}$ between nodes are indicated on the arcs in Figure 2.3. For simplicity, the arcs are symmetric, i.e. $(j, i)$ always exists and is always equal in length to $(i, j)$, for all $(i, j) \in A$. For each arc $(i, j)$ and commodity $w \in W$, we

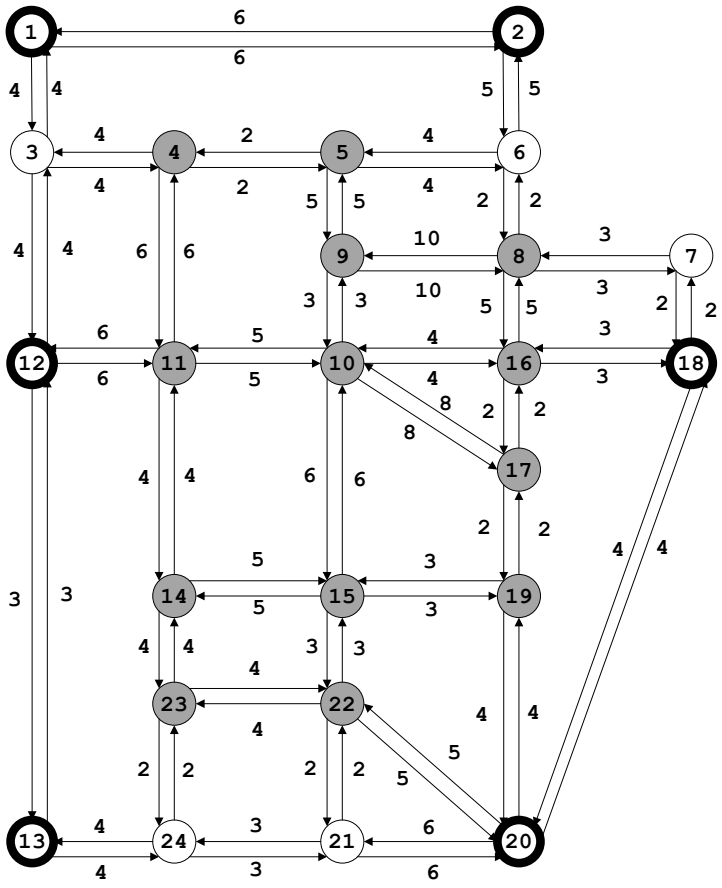



Figure 2.3: Sioux Falls road network

assign $a_{ijw}$ as 0.2, 0.25, and 0.3 for $w = 1, 2, 3$ respectively. For each scenario $k \in K$, we generate the probability $p^k$ by following the same method in Section 2.5.1.

Denote the mean demands for commodity $w$ at node $j$ by $\bar{d}_{jw}$. We set the mean demands at node 10 (the *center*) to be $\bar{d}_{10,1} = 1000, \bar{d}_{10,2} = 3000$, and $\bar{d}_{10,3} = 7000$, and set the mean demands at the other demand nodes $j$ to be $\bar{d}_{jw} = (1 - \lambda^n)\bar{d}_{10,w}$, $\forall w = 1, 2, 3$, where $n$ is the minimum number of links of a path from node 10 to node $j$ and $\lambda$ is the average rate of decay of demand from the center. Finally, we sample the realizations $d^k_{jw}$ from the distribution $U(0, 2\bar{d}_{jw})$, $w = 1, 2, 3$.

For each of the two instances, the supply for each commodity at each supply node is then generated randomly, while ensuring that $\sum_{i \in S_w} s_{iw} \geq \sum_{j \in D_w} d^k_{jw}$, $\forall w \in W$, $k \in K$, so that we will always have feasible solutions in every scenario.

#### 2.5.2.2 Results Summary

We analyze the sensitivity of optimal objective values to the parameters $\epsilon$ and $G$ for different values of $\lambda$. The models of PNDP-joint, PNDP-nc, and SNDP-wp are used in this comparison. Here, we used the homogenous versions of PNDP-nc (i.e. $\epsilon_{iw} = \epsilon/39$ for each $i \in D_w$, $w \in W$) and SNDP-wp (i.e. $G_{iw} = G$ for some fixed constant $G$ for each $i \in D_w$, $w \in W$).

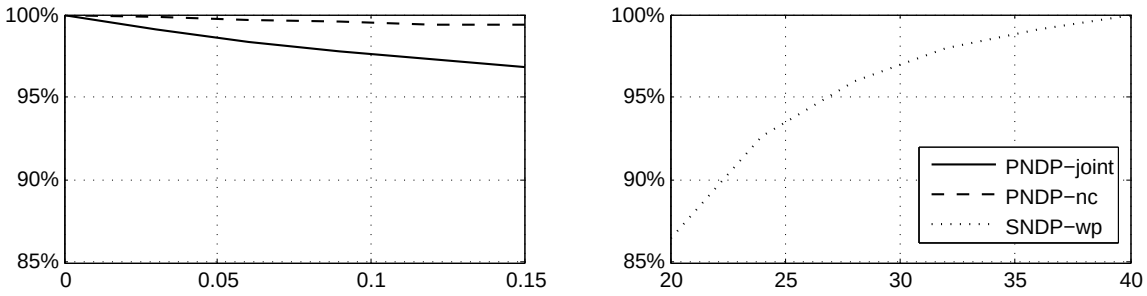
For the PNDP-joint and PNDP-nc cases, we used $\epsilon$ from 0 to $0.15$, in intervals of $0.03$. For the SNDP-wp case, we used $G$ from 20 to 40, in intervals of 4. For each of these values of $\epsilon$ or $G$, the optimal values of each model was found and taken as a percentage of the optimal value of the most restricted instance of each model, i.e. $\epsilon = 0$ for both PNDP models, and $G = 40$ for the SNDP-wp model. Figure $2.4$ illustrates the comparison of the three models for different levels of demand decay.

We list our observations based on the computational results as follows.

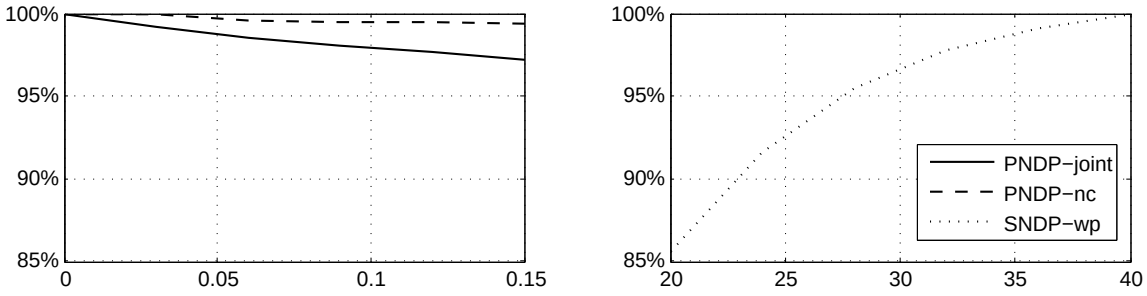
- The optimal values of PNDP-joint and PNDP-nc have a fairly linear relationship with $\epsilon$. This is in contrast with the curved graph of the SNDP-wp model; this is to be expected. As $G$ increases, the dominant term in the bi-objective function changes from the real cost of the solution to the virtual penalty cost incurred by unmet demand.

- The above point reveals the rather ambiguous nature of the penalty value $G$ - without first experimenting with several values of $G$, one cannot determine a suitable value of $G$ for which the objective function weighs the actual and virtual costs in a manner that is reasonable to the decision maker. PNDP models in general mitigate the ambiguity of solution reliability, and provide a decision maker with confidence levels on the QoS that (s)he can place in the solution, which an SNDP model cannot usually achieve.

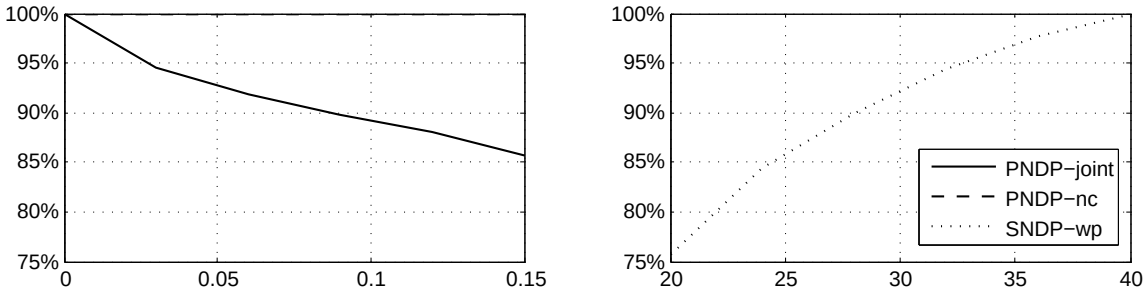
- The optimal objectives of PNDP-cont-nc serve relatively good upper bounds for the optimal objectives of PNDP-cont-joint, and become tighter when (i) $|K|$ increases, and/or (ii) in the heterogeneous risk setting. This might be of interest to some decision makers, who try to satisfy certain QoS levels by prioritizing their customer demands and requiring higher QoS levels for demands with higher variations. Instead of computing a time-consuming MILP of PNDP-cont-joint with a joint chance constraint, one can solve PNDP-cont-nc as a variant with multiple single-row chance constraints. This approximation in general might provide very tight bounds and high quality solutions based on all numerical results.




(a) $\lambda = 0$: even distribution of demand among demand nodes.




(b) $\lambda = 0.2$: moderate decay of demand from node 10.




(c) $\lambda = 1$: all demand at node 10 only.

Figure 2.4: Comparisons of the objective values of SNDP-wp, PNDP-joint, and PNDP-nc for varying values of $\epsilon$ and $G$

## 2.6 Concluding Remarks

In this chapter, we analyzed model variants and solution approaches of the probabilistic network capacity design problem with multicommodity flows. We first examined PNDP-cont-joint with a joint chance constraint that guarantees certain probability of no-demand losses at all nodes for all commodities. The problem was reformulated as an MILP by defining binary variables associated with each scenario. We then formulated three model variants that distribute risks into multiple chance constraints, namely, PNDP-cont-nc, PNDP-cont-c, and PNDP-cont-n. In addition to MILP reformulations, we discussed polynomial algorithms for solving PNDP-cont-nc by identifying risk thresholds of every single-row chance constraint. The modified approach then transformed the problem, and solved several shortest-path problems to attain optimality. Similar approaches were developed for solving special cases of PNDP-cont-c and PNDP-cont-n. We formulated benchmark stochastic programming models by either enforcing to meet all demand or penalizing unmet demand via a linear cost function, and tested different models and approaches on randomly generated network instances and an instance given by the Sioux-Falls network. Our results show that differentiating QoS levels for different commodities and/or customers (by using models with multiple chance constraints) can result in cost savings in network capacity design and transportation, as well as can yield better solution bounds (with much shorter computational times needed) for models having joint chance constraints (via unified QoS levels).

# CHAPTER 3

# Distributionally Robust Network Design

## 3.1 Introductory Remarks

In Chapter 2 we assumed that the distribution of demand was fully captured with known discrete realizations. This is a reasonable assumption to make in most cases when the problem has many historical data points that can be used to accurately construct the demand distribution. However, for NDPs applied to newly implemented networks, or for NDPs in which obtaining data is expensive (e.g. via manual survey/census) historical data may be scarce. In such cases, the data may be insufficient to apply a chance-constrained or even a stochastic optimization model, as the small number of observed realizations is less likely to accurately capture the true demand distribution and is more sensitive to the occurrence of rare outlying realizations. Similarly, traditional robust optimization (see Bertsimas and Sim, 2003, 2004) is often sensitive to outliers – since it is computed solely with the single worst-case realization, it may lead to overly conservative solutions.

When the demand distribution is not fully known, robust optimization is often used to solve NDPs. Robust optimization utilizes bounds on the uncertain parameters to construct an uncertainty set, and optimize the worst-case objective value for any parameter realization in the uncertainty set. Since only the bounds of the uncertainty are required for robust optimization, it is well-suited to conservatively solve problems without full information on the distribution of uncertainty. We refer the reader to Ben-Tal and Nemirovski (1998) and Bertsimas and Sim (2004) for the methodology of robust optimization in generic convex optimization problems and linear programs respectively. Ukkusuri et al. (2007) robustly designed a transportation network under discrete demand uncertainty. Similarly, Mudchanatongsuk et al. (2008) proposed a robust transportation NDP under demand and transportation cost uncertainty, representing the uncertainty in both cases with box-shaped, polyhedral and ellipsoidal uncertainty sets. In supply chain network design, Pishvaee et al. (2011) applied robust optimization to minimize the cost of the supply chain network under

box-shaped uncertainty sets for demand, returns, and transportation costs.

More recently, data-driven distributionally robust optimization has been used to tackle optimization problems under distributional uncertainty (e.g. Calafiore and El Ghaoui, 2006; Goh and Sim, 2010). In contrast with robust optimization, which plans against the worst-case realization, distributionally robust optimization plans against the worst-case distribution. This worst-case distribution is selected from an ambiguity set of feasible candidate distributions, which is typically constructed based on the uncertainty structure. We use a moment-based ambiguity set (Delage and Ye, 2010) in this study, which comprises distributions which have moments equal to those of the small set of observations. Since a moment-based ambiguity set is constructed using only the moments of the observations, a benefit of using such an ambiguity set is that solution time of the problem is unaffected by the number of observations, yet all observations are implicitly used when computing the moments, as opposed to robust optimization, which uses a single worst-case realization.

To the best of our knowledge, the use of distributionally robust optimization to solve NDPs is a novel approach. The majority of the literature on NDPs focuses on problems with fully known distributional information or sufficient historical observations to approximate the distribution; when distributional information is sparse, robust optimization is used instead. Solving NDPs via distributionally robust optimization is less conservative than using robust optimization, yet is robust enough to be less sensitive to the distribution of the observations, when compared to using a risk-neutral stochastic optimization approach.

We approximate DR-NDP under a marginal moment-based ambiguity set with an optimization problem that contains an embedded MILP in its constraints. While the expressions in the optimization problem are linear, it cannot be solved directly due to the embedded problem. We develop a cutting-plane algorithm that iteratively generates cuts with the embedded optimization problem, to obtain approximate solutions to DR-NDP. Through our results, we will show that the solutions obtained via this algorithm are less sensitive to the distribution of observations than those obtained via an expectation-based stochastic optimization model.

This chapter is organized as follows. Section 3.2 describes a generic NDP and the formulation of the marginal moment-based ambiguity set for DR-NDP. Section 3.3 reformulates DR-NDP under the marginal moment-based ambiguity set as an approximate problem, which is utilized in Section 3.4 to develop a cutting-plane algorithm to obtain approximate solutions to DR-NDP. Section 3.5 details the solution performance of DR-NDP relative to a benchmark expectation-based stochastic NDP formulation, and Section 3.6 concludes the chapter and describes future research directions.

## 3.2 Problem formulation

The setup of DR-NDP is similar to that of PNDP, albeit with only a single commodity. We are given a network $G(N, A)$ with directed arcs $(i, j) \in A$ of undetermined capacity, where $i, j \in N$, with the aim of minimizing the total cost of arc capacity allocation, commodity flow and penalties due to unmet demand. The network has disjoint sets of supply nodes $S \subset N$ and demand nodes $D \subset N$. Each supply node $i \in S$ has deterministic supply $s_i > 0$ while each demand node $i \in D$ has uncertain demand $d_i \geq 0$. The demands are assumed to be jointly distributed and, while the true distribution of the demand vector $d$ is not known, we are given a set of $|K|$ previously observed realizations of $d$, denoted by $\{d^k\}_{k \in K}$. We assume $d$ has a boxed-shaped support $\Xi$, given by $\Xi := \{d \in \mathbb{R}_+^{|D|} : \underline{d}_i \leq d_i \leq \overline{d}_i\}$, where $\underline{d}, \overline{d} \in \mathbb{R}_+^{|D|}$ denote the vector of simple lower and upper bounds of $d$, respectively. The use of a box-shaped support is analogous to the use of a box-shaped uncertainty set (Soyster, 1973) in robust optimization.

### 3.2.1 Formulation of the NDP Under Demand Uncertainty

The NDP aims to determine the optimal capacities on the arcs in the network that minimizes overall cost. The cost is determined in the two stages of the NDP. The first stage determines the capacity $x_{ij} \in \mathbb{R}_+$ on each arc $(i, j) \in A$ in the network. Each unit of capacity allocated to arc $(i, j)$ incurs a cost of $c_{ij} > 0$ for all $(i, j) \in A$. We refer to the total capacity planning cost in the first stage as the *capacity cost*, which is incurred before demand is realized. Since the capacity on any arc never exceeds the maximum total demand, we impose an explicit feasibility constraint $x \in X$ on the vector of capacity variables $x$, where

$$X := \left\{ x \in \mathbb{R}_+^{|A|} : \ 0 \leq x_{ij} \leq \sum_{i \in D} \overline{d}_i, \ \forall (i, j) \in A \right\}.$$

In specific NDP contexts, $X$ can include other linear constraints. For example, budget constraints may be present in supply chain NDP contexts, limiting the total capacity or capacity cost of the network. In transportation NDPs, roads may be required to always be two-way, necessitating a constraint $x_{ij} = x_{ji}$ for all arcs $(i, j), (j, i) \in A$. As these constraints are specific to the context of the problem, they are not the focus of this study and we exclude them from our discussion.

In the second stage, we optimize the flows on the network to minimize the *second-stage cost* of the first-stage decisions. Given a first-stage capacity vector $x$ and a demand vector $d$, the second-stage cost $g(x, d)$ is given by a capacitated minimum cost flow that penalizes

any unmet demand due to insufficient arc capacity. In other words, we formulate $g(x, d)$ by

$$
\begin{aligned}
g(x,d) := \quad & \min_{y,t} \quad \sum_{(i,j)\in A} a_{ij}y_{ij} + \sum_{i\in D} G_i t_i && && (3.1)\\
& \text{s.t.} \quad y_{ij} \le x_{ij} && \forall (i,j)\in A && (3.2)\\
& \qquad \sum_{(i,j)\in A} y_{ij} - \sum_{(j,i)\in A} y_{ji} \le s_i && \forall i \in S && (3.3)\\
& \qquad \sum_{(i,j)\in A} y_{ij} - \sum_{(j,i)\in A} y_{ji} = 0 && \forall i \in N\backslash(S\cup D) && (3.4)\\
& \qquad \sum_{(i,j)\in A} y_{ij} - \sum_{(j,i)\in A} y_{ji} - t_i \le -d_i && \forall i \in D && (3.5)\\
& \qquad y, t \ge 0, && && (3.6)
\end{aligned}
$$

where $y \in \mathbb{R}_+^{|A|}$ denotes the vector of flows $y_{ij}$ on each arc $(i, j) \in A$ and $t \in \mathbb{R}_+^{|D|}$ denotes the vector of unmet demand $t_i$ at each demand node $i \in D$. Each unit of flow on arc $(i, j)$ incurs a *flow cost* of $a_{ij} > 0$, and each unit of unmet demand at demand node $i \in D$ incurs a *penalty cost* of $G_i > 0$ in the objective function. Constraint (3.2) restricts the capacities on the arcs, while constraints (3.3)–(3.5) balance the flows on the network. Constraint (3.5) also determines the amount of unmet demand at each demand nodes.

Under ideal circumstances, if the distribution $F$ of the demand vector $d$ is known, then a risk-neutral formulation of NDP is given by

$$
\min_{x\in X} \left\{ \sum_{(i,j)\in A} c_{ij}x_{ij} + \mathbb{E}_F\left[g(x,d)\right] \right\}, \tag{3.7}
$$

where $\mathbb{E}_F$ is the expectation taken under the distribution $F$. However, $F$ is rarely known in practical applications of the NDP – distribution information on $d$ is often limited to the previously observed realizations of $d$. If data is scarce, the observed realizations will not be sufficient to determine $F$ with reasonable accuracy. Here, we use distributionally robust optimization to tackle such NDPs with scarce data, using the marginal moments of the scarce data to construct ambiguity sets for DR-NDP.

### 3.2.2 Formulation of DR-NDP with a Marginal Moment-based Ambiguity Set

Traditional robust optimization seeks the worst-case realization of $d$. Such an approach may be too robust if the observed set contains an extremely bad but very rare realization. Distributionally robust optimization instead seeks the worst-case distribution $F$ of $d$ for which the expected second-stage cost $\mathbb{E}_F[g(x,d)]$ is maximized. We first establish an *ambiguity set*, denoted by $\mathcal{M}$, of candidate distributions from which the worst-case $F$ is found. The DR-NDP is then formulated as follows:

$$\min_{x \in X} \left\{ \sum_{(i,j) \in A} c_{ij} x_{ij} + \max_{F \in \mathcal{M}} \left\{ \mathbb{E}_F \big[ g(x,d) \big] \right\} \right\}. \tag{3.8}$$

The solutions to DR-NDP depend on the ambiguity set $\mathcal{M}$, since the solutions have to be robust with respect to distributions in this set. Similarly, the reformulation methodology of DR-NDP depends on the type of ambiguity set used. We use an ambiguity set based on the marginal first and second moments of the observed realizations of the random parameter $d$. We construct the ambiguity set with the marginal mean and variance vectors $\mu \in \mathbb{R}_+^{|D|}$ and $\sigma^2 \in \mathbb{R}_+^{|D|}$ of the observed realizations, given by

$$\begin{aligned} \mu_i &:= \frac{1}{|K|} \sum_{k \in K} d_i^k && \forall i \in D \\ \sigma_i^2 &:= \frac{1}{|K|} \sum_{k \in K} \left( d_i^k - \mu_i \right)^2 && \forall i \in D. \end{aligned}$$

This ambiguity set considers only distributions whose marginal means and variances match $\mu_i$ and $\sigma_i^2$ respectively, for each $i \in D$. Explicitly, this *marginal moment-based ambiguity set* is defined by

$$\begin{aligned} \mathcal{M}_0(\Xi, \mu, \sigma) := \{ F : & \\ & \int_\Xi dF(d) = 1 && (3.9) \\ & \int_\Xi d_i \, dF(d) = \mu_i && \forall i \in D \quad (3.10) \\ & \int_\Xi d_i^2 \, dF(d) = \mu_i^2 + \sigma_i^2 && \forall i \in D \}. \quad (3.11) \end{aligned}$$

Constraint (3.9) ensures $\mathcal{M}_0$ only contains valid distributions over $\Xi$, while constraint (3.10) restricts these distributions to have marginal means and variances equal to those

of the observed realizations. Note that the solution time of DR-NDP using this set is unaffected by the size of the set of observations – since the ambiguity set is constructed from only $\mu$ and $\sigma^2$, only pre-processing time is affected by the size of the set.

## 3.3 Reformulation of DR-NDP as an Approximate Problem

We develop methodologies for reformulating and eventually solving the DR-NDP model (3.8) using a marginal moment-based ambiguity set. We begin by dualizing the second-stage maximization problem

$$\max_{F \in \mathcal{M}} \left\{ \mathbb{E}_F \big[ g(x, d) \big] \right\} \tag{3.12}$$

to obtain a minimization problem that can be optimized together with the first-stage arc capacity design problem, which also has a minimization objective. This section details this dualization and a subsequent approximation of the objective function to estimate the value of DR-NDP under a marginal moment-based ambiguity set.

### 3.3.1 Dualizing the Second-stage Problem

If $\mathcal{M} = \mathcal{M}_0(\Xi, \mu, \sigma)$, then the second-stage problem written explicitly is

$$\max_{F} \left\{ \int_{\Xi} g(x, d)\ dF(d) : (3.9)\text{–}(3.11) \right\}, \tag{3.13}$$

a linear program maximizing over all distributions $F$ in $\mathcal{M}_0(\Xi, \mu, \sigma)$, with the expectation of $g(x, d)$ over this distribution as the objective function.

We denote the dual variables corresponding to the constraints (3.9), (3.10) and (3.11) by $z \in \mathbb{R}$, $r \in \mathbb{R}^{|D|}$ and $v \in \mathbb{R}^{|D|}$, respectively. Taking the dual of the second-stage problem then yields

$$\min_{z,r,v} \quad z + \sum_{i \in D} \mu_i r_i + \sum_{i \in D} \left( \mu_i^2 + \sigma_i^2 \right) v_i \tag{3.14}$$

$$\text{s.t.} \quad z + \sum_{i \in D} d_i r_i + \sum_{i \in D} d_i^2 v_i \geq g(x, d) \qquad \forall d \in \Xi. \tag{3.15}$$

Next, we replace (3.12) with (3.14)–(3.15) in DR-NDP (3.8), and merge the minimization objective of the dualized second-stage problem with the minimization objective of the

first-stage arc capacity design problem to obtain the reformulation of DR-NDP (3.8) under the moment-based ambiguity set $\mathcal{M}_0(\Xi, \mu, \sigma)$:

$$\min_{x,z,r,v} \left\{ z + \sum_{i \in D} \mu_i r_i + \sum_{i \in D} \left(\mu_i^2 + \sigma_i^2\right) v_i + \sum_{(i,j) \in A} c_{ij} x_{ij} : x \in X, (3.15) \right\}. \tag{3.16}$$

However, this formulation still remains intractable as constraint (3.15) is a semi-infinite constraint (i.e. a constraint that must be fulfilled over the infinite set $\Xi$). We reformulate this constraint as a tractable one in the next section.

### 3.3.2 Approximating the Semi-infinite Constraint

Since constraint (3.15) must be satisfied for all $d \in \Xi$, it must also be satisfied for the worst-case $d$. We move all terms containing $d$ to the right-hand side (RHS) of constraint (3.15) and reformulate it as

$$z \geq \max_{d \in \Xi} \left\{ g(x, d) - \sum_{i \in D} d_i r_i - \sum_{i \in D} d_i^2 v_i \right\}. \tag{3.17}$$

The RHS of constraint (3.17) now has a minimization problem $g(x, d)$ within the maximization over $d \in \Xi$. We obtain the dual maximization problem of $g(x, d)$ to merge with the maximization over $d \in \Xi$.

**Lemma 3.1.** For fixed $r$, $v$ and $x$ feasible to problem (3.16), the value of

$$\max_{d \in \Xi} \left\{ g(x, d) - \sum_{i \in D} d_i r_i - \sum_{i \in D} d_i^2 v_i \right\} \tag{3.18}$$

is equal to that of

$$\max_{d \in \Xi, (\theta, \pi) \in \mathcal{P}} \left\{ \sum_{i \in D} d_i \pi_i - \sum_{i \in S} s_i \pi_i - \sum_{(i,j) \in A} x_{ij} \theta_{ij} - \sum_{i \in D} d_i r_i - \sum_{i \in D} d_i^2 v_i \right\} \tag{3.19}$$

where $\mathcal{P}$ is the feasible region given by

$$\begin{aligned} \mathcal{P} := \{ & (\theta, \pi) : \\ & \pi_j - \pi_i - \theta_{ij} \leq a_{ij} && \forall (i,j) \in A && (3.20) \\ & \theta_{ij} \geq 0 && \forall (i,j) \in A && (3.21) \end{aligned}$$

$$\pi_i \geq 0 \qquad \forall i \in S \tag{3.22}$$

$$0 \leq \pi_i \leq G_i \qquad \forall i \in D\}. \tag{3.23}$$

□

*Proof.* We begin by taking the dual of $g(x, d)$. By denoting the dual variables corresponding to constraints (3.2), (3.3), (3.4), and (3.5) of the capacitated minimum cost flow problem $g(x, d)$ by $\theta \in \mathbb{R}_+^{|A|}$, $\pi_i$ for $i \in S$, $\pi_i$ for $i \in N \backslash (S \cup D)$, and $\pi_i$ for $i \in D$, respectively, we can write $g(x, d)$ in its dual form:

$$g(x,d) = \max_{\theta,\pi} \left\{ \sum_{i \in D} d_i \pi_i - \sum_{i \in S} s_i \pi_i - \sum_{(i,j) \in A} x_{ij} \theta_{ij} : (3.20)\text{–}(3.23) \right\}. \tag{3.24}$$

Substituting the dual (3.24) into (3.18) yields

$$\max_{d \in \Xi} \left\{ \max_{(\theta,\pi) \in \mathcal{P}} \left\{ \sum_{i \in D} d_i \pi_i - \sum_{i \in S} s_i \pi_i - \sum_{(i,j) \in A} x_{ij} \theta_{ij} \right\} - \sum_{i \in D} d_i r_i - \sum_{i \in D} d_i^2 v_i \right\}. \tag{3.25}$$

We merge the maximization over $(\theta, \pi) \in \mathcal{P}$ with the maximization over $d \in \Xi$ to complete the reformulation. □ □

Problem (3.19) still cannot be solved directly as it contains bilinear terms $d_i \pi_i$ in its objective, with both $d_i$ and $\pi_i$ being continuous for all $i \in D$. Since $\pi_i$ is bounded below and above (due to duality), we aim to rewrite $d_i$ in terms of binary variables to reformulate the bilinear terms. Consequently, the following lemma further reformulates problem (3.19) by leveraging the fact that $d$ has a box-shaped support.

**Lemma 3.2.** For fixed $r$, $v$ and $x$ feasible to problem (3.16), the value of problem (3.19) is equal to that of

$$\max_{(\theta,\pi) \in \mathcal{P}} \left\{ \sum_{i \in D} \left( \max_{\underline{d}_i \leq d_i \leq \overline{d}_i} \left\{ d_i \pi_i - d_i r_i - d_i^2 v_i \right\} \right) - \sum_{i \in S} s_i \pi_i - \sum_{(i,j) \in A} x_{ij} \theta_{ij} \right\}. \tag{3.26}$$

□

*Proof.* Due to the simple feasible region of $d$, we separate the maximizations over $(\theta, \pi) \in \mathcal{P}$ and $d \in \Xi$ in the objective. The order of maximization is swapped from that in (3.25), so that the terms in $d$ can be grouped then separated by the index $i \in D$. Note that the

swapping of maximization order does not change the problem as the final solution in both cases maximizes over both feasible regions $\mathcal{P}$ and $\Xi$.

$$\max_{(\theta,\pi)\in\mathcal{P}} \left\{ \max_{d\in\Xi} \left\{ \sum_{i\in D} d_i\pi_i - \sum_{i\in D} d_i r_i - \sum_{i\in D} d_i^2 v_i \right\} - \sum_{i\in S} s_i\pi_i - \sum_{(i,j)\in A} x_{ij}\theta_{ij} \right\}. \quad (3.27)$$

We observe that the maximization over $d \in \Xi$ is separable by the indices $i \in D$ since the box-shaped support $\Xi$ is defined by independent simple lower and upper bounds on each dimension. We separate this maximization problem into the $|D|$ maximization problems given in (3.26), each over a simple interval, to complete the reformulation. □ □

**Theorem 3.2.** For fixed $r$, $v$ and $x$ feasible to problem (3.16), the value of problem (3.18) is equal to that of problem (3.26). □

*Proof.* This is a direct consequence of Lemmas 3.1 and 3.2. □ □

Through Lemma 3.2, we first consider each maximization problem

$$\max_{\underline{d}_i \le d_i \le \overline{d}_i} \left\{ d_i\pi_i - d_i r_i - d_i^2 v_i \right\} \quad (3.28)$$

for all $i \in D$ separately. Although obtaining the analytic solution of (3.28) in terms of $\pi_i$, $r_i$ and $v_i$ is trivial, the analytic solution is not useful in generating cuts in our subsequent cutting-plane algorithm. Instead, we approximate the objective function of (3.28) as a piecewise-linear function with $N_i$ intervals, and maximize over the piecewise-linear function. Since the maximum over the piecewise-linear function lies on exactly one of the $N_i+1$ interval endpoints, the problem simplifies to finding the maximum of $N_i+1$ different values.

We define intervals of equal length such that

$$\widehat{d}_{in} = \underline{d}_i + \frac{n}{N_i}\left(\overline{d}_i - \underline{d}_i\right), \quad \forall n = 0, 1, \ldots, N_i,$$

is the $n^{\text{th}}$ endpoint for each $i \in D$. Given these endpoints, we approximate problem (3.28) as the largest value obtained when substituting $d_i$ with one of the endpoints in $\{\widehat{d}_{in}\}_{n=0,1,\ldots,N_i}$, or, in other words,

$$\max_{n\in\{0,1,\ldots,N_i\}} \left\{ \widehat{d}_{in}\pi_i - \widehat{d}_{in} r_i - \widehat{d}_{in}^2 v_i \right\} \quad (3.29)$$

We define binary variables $\rho_{in} \in \{0, 1\}$, for each $n = 0, 1, \ldots, N_i$, such that $\rho_{in} = 1$

if the $n^{\text{th}}$ endpoint $\widehat{d}_{in}$ yields the maximum for problem (3.29), and $\rho_{in} = 0$ otherwise, for each $i \in D$. Each set $\{\rho_{in}\}_{n=0,1,\ldots,N_i}$, for all $i \in D$, is a Specially Ordered Set of Type 1 (SOS1), containing binary variables that sum to one, such that

$$\sum_{n=0}^{N_i} \rho_{in} = 1 \tag{3.30}$$

$$\rho_{in} \in \{0,1\},\ \forall n = 0,1,\ldots,N_i \tag{3.31}$$

for all $i \in D$, to ensure that exactly one endpoint is selected to maximize problem (3.29). Subsequently problem (3.29), can be written equivalently as

$$\max_{\rho_i} \left\{ \sum_{n=0}^{N_i} \rho_{in} \left( \widehat{d}_{in}\pi_i - \widehat{d}_{in} r_i - \widehat{d}_{in}^2 v_i \right) : (3.30)\text{–}(3.31) \right\} \tag{3.32}$$

for all $i \in D$, where $\rho_i \in \mathbb{R}^{N_i}$ denotes the vector of $\rho_{in}$ for $n = 0,1,\ldots,N_i$. Then, the objective takes the value $\widehat{d}_{in}\pi_i - \widehat{d}_{in} r_i - \widehat{d}_{in}^2 v_i$ if $\rho_{in} = 1$ (and the remaining SOS1 variables in the same set are zero), or, equivalently, when $\widehat{d}_{in}$ maximizes problem (3.29).

Since $\rho_{in}$ is binary and (3.23) gives us the bounds $0 \leq \pi_i \leq G_i$, we can subsequently reformulate the bilinear terms $\rho_{in}\pi_i$ with McCormick inequalities (McCormick, 1976). The following McCormick inequalities introduce the auxiliary variable $\lambda_{in}$, for which the equivalence relation $\lambda_{in} \equiv \rho_{in}\pi_i$ holds for all $n = 0,1,\ldots,N_i$, $i \in D$.

$$\lambda_{in} - \pi_i - G_i\rho_{in} \geq -G_i \qquad \forall n = 0,1,\ldots,N_i,\ i \in D \tag{3.33}$$

$$\lambda_{in} - \pi_i \leq 0 \qquad \forall n = 0,1,\ldots,N_i,\ i \in D \tag{3.34}$$

$$\lambda_{in} - G_i\rho_{in} \leq 0 \qquad \forall n = 0,1,\ldots,N_i,\ i \in D \tag{3.35}$$

$$\lambda_{in} \geq 0 \qquad \forall n = 0,1,\ldots,N_i,\ i \in D. \tag{3.36}$$

When $\rho_{in} = 1$, constraints (3.33) and (3.34) bound $\lambda_{in}$ below and above by $\pi_i$, and when $\rho_{in} = 0$, constraints (3.35) and (3.36) bound $\lambda_{in}$ below and above by $0$. This results in the desired equivalence relation of $\lambda_{in} \equiv \rho_{in}\pi_i$.

Using the approximation of (3.28) with (3.32) and the above linear reformulation of bilinear terms via McCormick inequalities, we approximate the value of problem (3.18), and consequently problem (3.26), with the problem

$$\begin{aligned} h(x,r,v) := \quad & \max_{\theta,\pi,\rho,\lambda} \quad f(x,r,v,\theta,\pi,\rho,\lambda) \\ & \text{s.t.} \quad (3.20)\text{–}(3.23),(3.33)\text{–}(3.36) \end{aligned} \tag{3.37}$$

$$\sum_{n=0}^{N_i} \rho_{in} = 1 \qquad \forall i \in D \tag{3.38}$$

$$\rho_{in} \in \{0,1\} \qquad \forall n = 0,1,\ldots,N_i,\ i \in D, \tag{3.39}$$

where

$$f(x,r,v,\theta,\pi,\rho,\lambda) := \sum_{i\in D}\sum_{n=0}^{N_i} \left(\widehat{d}_{in}\lambda_{in} - \left(\widehat{d}_{in}r_i\rho_{in} + \widehat{d}_{in}^2 v_i\right)\rho_{in}\right) - \sum_{i\in S} s_i\pi_i - \sum_{(i,j)\in A} x_{ij}\theta_{ij}. \tag{3.40}$$

Therefore, we approximate the value of DR-NDP (3.16) under the marginal moment-based distributional set with the problem

$$\textbf{APPROX}: \quad \min_{x,z,r,v} \quad z + \sum_{i\in D}\mu_i r_i + \sum_{i\in D}\left(\mu_i^2+\sigma_i^2\right)v_i + \sum_{(i,j)\in A} c_{ij}x_{ij} \tag{3.41}$$

$$\text{s.t.} \quad z \geq h(x,r,v) \tag{3.42}$$

$$x \in X. \tag{3.43}$$

**APPROX**, while having only linear expressions and both binary and continuous variables, is not a mixed-integer linear problem due to constraint (3.42) having a maximization problem on the RHS. Using the embedded maximization problem $h(x,r,v)$ as a cut-generating subproblem, we develop a cutting-plane algorithm to solve DR-NDP. We describe this algorithm in further detail in Section 3.4.

Note that by approximating (3.26) with $h(x,r,v)$, we have replaced $|D|$ continuous variables with $\sum_{i\in D} N_i$ binary variables and $\sum_{i\in D} N_i$ continuous variables, and have also introduced $4\sum_{i\in D} N_i + |D|$ new constraints to the formulation. Although the binary variables comprise $|D|$ sets of SOS1 variables, the introduction of a large number of binary variables will negatively impact computational efficiency. While having a larger $N_i$ for each $i \in D$ gives better approximations, the accuracy of the approximation will come at a cost of greater computational time, so some care should be taken to ensure that the appropriate number of intervals is used in the piecewise approximation.

## 3.4 Cutting-plane Algorithm

We describe a cutting-plane algorithm that iteratively passes solutions to the maximization problem $h(x,r,v)$ and generates a cut for $z$, $r$, $v$ and $x$ should the current solution not be feasible to $h(x,r,v)$.To perform this cutting-plane algorithm, constraint (3.42) is relaxed,

with its maximization RHS solved as a cut-generating subproblem. However, **APPROX** without (3.42) is unbounded since $z$, $r_i$ and $v_i$ are not bounded below. We use a trivial lower bound of 0 on the objective function to ensure a bounded problem in the first iteration.

With this lower bound, the master problem at its $M^{\text{th}}$ iteration is given by

$$\textbf{MASTER}: \min_{x,z,r,v} \quad z + \sum_{i \in D} \mu_i r_i + \sum_{i \in D} \left(\mu_i^2 + \sigma_i^2\right) v_i + \sum_{(i,j) \in A} c_{ij} x_{ij} \tag{3.44}$$

$$\text{s.t.} \quad z + \sum_{i \in D} \mu_i r_i + \sum_{i \in D} \left(\mu_i^2 + \sigma_i^2\right) v_i + \sum_{(i,j) \in A} c_{ij} x_{ij} \geq 0 \tag{3.45}$$

$$z \geq f(x, r, v, \theta^m, \pi^m, \rho^m, \lambda^m) \qquad \forall m = 1, \ldots, M \tag{3.46}$$

$$x \in X, \tag{3.47}$$

where constraint (3.45) enforces a trivial lower bound of 0 and constraint (3.46) is the set of cuts generated by the algorithm. The number of generated cuts is denoted by $M$, and the solution of the subproblem in the $m^{\text{th}}$ iteration is denoted by $(\theta^m, \pi^m, \rho^m, \lambda^m)$.

These cuts are iterative generated by passing the current solution generated by **MASTER** with its current set of cuts to the subproblem, which checks if the solution is feasible with respect to constraint (3.42). This is checked by solving the subproblem given by $h(x, r, v)$. Should the current solution be infeasible with respect to constraint (3.42), the algorithm generates a cut to remove the current solution. The cutting-plane algorithm for DR-NDP is described in Algorithm 3.3.

**Algorithm 3.3** Cutting-plane algorithm for solving DR-NDP

**for** $i \in D$ **do**
  Select a number of piecewise intervals $N_i$.
**end for**
Select a non-negative tolerance value $\epsilon$.
Initialize iteration counter $M = 0$.
**repeat**
  $M \leftarrow M + 1$
  Solve **MASTER** and denote optimal solution by $(x^M, z^M, r^M, v^M)$.
  Solve $h(x^M, r^M, v^M)$ via branch-and-bound with each node solved via the simplex algorithm, and denote optimal solution by $(\theta^M, \pi^M, \rho^M, \lambda^M)$.
  **if** $z^M < (1 - \epsilon) f(x^M, r^M, v^M, \theta^M, \pi^M, \rho^M, \lambda^M)$ **then**
    Add $w \geq f(x, r, v, \theta^M, \pi^M, \rho^M, \lambda^M)$ to **MASTER** as the $M^{\text{th}}$ cut.
  **end if**
**until** No cuts were added to **MASTER**
**return** Optimal solution for **APPROX**: $(x^M, z^M, r^M, v^M)$.

It is important to note that the cutting-plane algorithm only terminates if the subproblem

$h(x, r, v)$ is solved via branch-and-bound with each node solved via the simplex algorithm.

**Theorem 3.3.** Algorithm 3.3 converges to a solution optimal to **APPROX** in a finite number of iterations. □

*Proof.* We first note that $h(x, r, v)$ has a feasible region independent of $r$, $v$ and $x$. Furthermore, the feasible region at each node of the branch-and-bound algorithm when solving for $h(x, r, v)$ is polyhedral. Since each node is solved via the simplex algorithm, the optimal solution obtained for each node is necessarily an extreme point solution. Consequently, each generated cut necessarily corresponds to some extreme point of the feasible region of some branch-and-bound node.

**MASTER** generates solutions that satisfy all previously generated cuts, and a new cut is generated only if the new cut is violated by the current solution, so each cut corresponds to a unique extreme point. Furthermore, there are a finite number of binary variables $\rho_{in}$ to branch on and the polyhedral feasible region of each branch-and-bound node must have a finite number of extreme points. Hence, Algorithm 3.3 will terminate within a finite number of cuts, by generating at most one cut for each possible extreme point. □ □

## 3.5 Computational Results

We describe computational results from using distributionally robust optimization to solve NDPs. In particular, we focus on illustrating the relative insensitivity of the solutions, obtained with DR-NDP, to changes in input observed realizations. We describe a benchmark sample average approximation (SAA, see Kleywegt et al., 2002; Shapiro and Homem-de-Mello, 2000)-based model, given by Problem (3.48), which we compare against DR-NDP. The setup of the grid networks and the models that we used in our computations are discussed in Section 3.5.2, together with a description of the NOBEL-US network as representative a real-world network. The comparison of the sensitivity of these solutions against those obtained with the SAA-based model will the main focus of our analysis, which is covered in Section 3.5.3. The full set of results for our computations can be found in Appendix B. We implement all the algorithms in Java using Gurobi 6.0.3. The results below are obtained by using a Dell Alienware X51, with an Intel Core i7-3770 dual-core CPU @ 3.4GHz each and 8GB RAM.

### 3.5.1 Benchmark SAA-based Model

We use a benchmark model based on SAA, which generates realizations of the uncertain parameter, typically through Monte Carlo sampling, and solves the optimization problem

risk-neutrally with the true distribution approximated by the generated realizations under a discrete uniform distribution. Our benchmark model generates additional demand realizations using the marginal moment information of the initial small set of observed realizations, together with an assumption on the family of distributions that the marginal distributions of demand belong to. The parameters of the assumed marginal distributions are then computed to have first and second moments equal to those of the observed realizations.

For example, suppose we are given an initial set of observations with marginal means and variances equal to $\mu_i$ and $\sigma_i^2$ for all $i \in D$. If we assume that marginal distributions of the observations are uniform, then we generate an additional demand vector realization $d^\omega$ by sampling the demands $d_i^\omega$ for each demand node $i \in D$ from the uniform distribution with lower bound $\mu_i - \sqrt{3}\sigma_i$ and upper bound $\mu_i + \sqrt{3}\sigma_i$ (so that the marginal uniform distributions have equal means and variances with the observations). Alternatively, if we assume that the marginal distributions of the sample are gamma, then we sample each $d_i^\omega$ instead from the gamma distribution with shape $\mu_i^2/\sigma_i^2$ and scale $\sigma_i^2/\mu_i$ for each $i \in D$. We denote the $|\Omega|$ generated demand vectors by $\{d^\omega\}_{\omega\in\Omega}$.

Following SAA, we assume that the realizations are sampled with equal probability $\frac{1}{|\Omega|}$, and solve the problem

$$\min_{x\in X}\left\{\sum_{(i,j)\in A} c_{ij}x_{ij} + \frac{1}{|\Omega|}\sum_{\omega\in\Omega} g\left(x, d^\omega\right)\right\}. \tag{3.48}$$

Problem (3.48) converges to

$$\min_{x\in X}\left\{\sum_{(i,j)\in A} c_{ij}x_{ij} + \mathbb{E}_{F_0}\Big[g\left(x, d^\omega\right)\Big]\right\},$$

as $|\Omega| \to \infty$, where $\mathbb{E}_{F_0}$ is the expectation taken over the assumed distribution. Hence, this benchmark aims to give risk-neutral solutions under the assumption that the demand distribution has marginal mean and variance vectors matching $\mu$ and $\sigma^2$ respectively, and belongs to the distribution family assumed in the approach. In our computations, we will only assume gamma and uniform marginal distributions when generating additional realizations.

### 3.5.2 Experimental Setup

We use square grid networks for the majority of our computational results, with sizes ranging from 3x3 up to 7x7. As shown in the 3x3 network in Figure 3.1, arcs are present both

ways between adjacent nodes. However, it should be noted that while the arcs are *present*, whether they have non-zero capacity still depends on the solution given by the model.

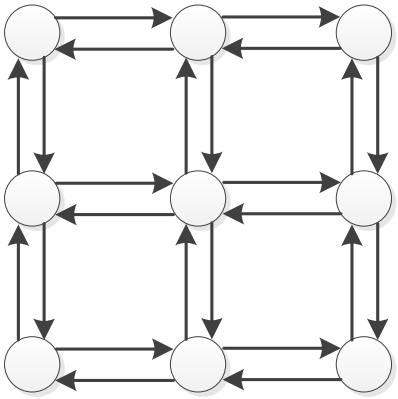


Figure 3.1: Example 3x3 grid network

To avoid bias caused by network topology, we also use the NOBEL-US network (Orlowski et al., 2010), shown in Figure 3.2, as a representative real-world network. As seen in the figure, it is based on major road connections in the United States of America. For the purposes of our experiments, connections in the NOBEL-US network are two-way (just as with the grid networks), giving a total of 14 nodes and 42 arcs.

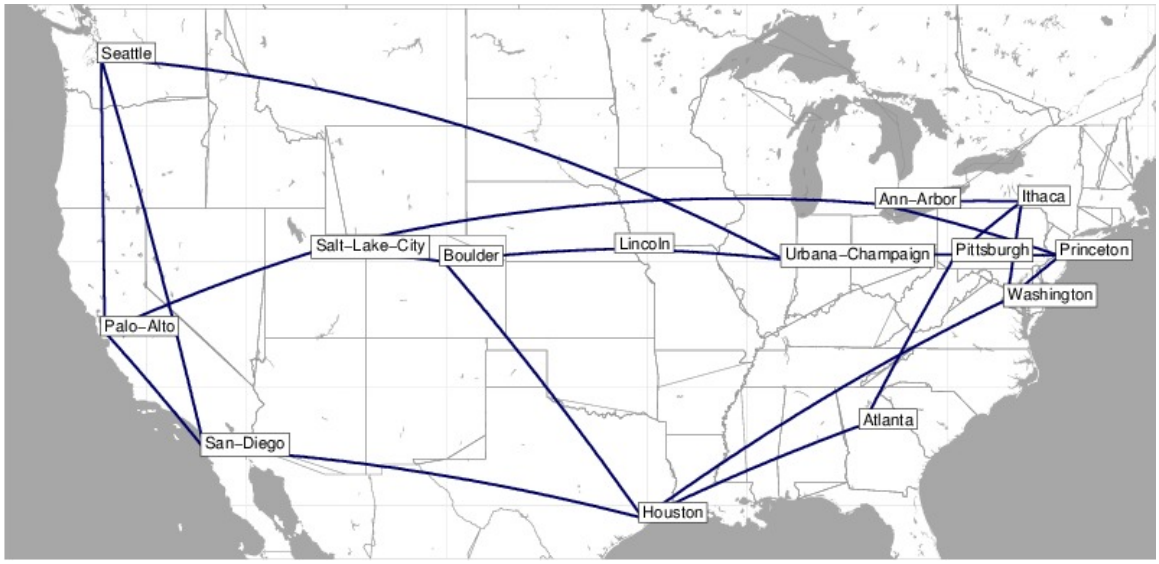



Figure 3.2: NOBEL-US network

We allocate three supply and three demand nodes to each network, regardless of network size, and locate these six nodes randomly in the network such that the supply and de-

mand nodes do not overlap. Each supply node is given a deterministic supply of 20 units. The demand nodes have uncertain demand, with independent distributions. We generate 10,000 demand realization vectors as the "true" distribution of demand. For convenience, we will refer to these 10,000 demand realizations as the reference set. The three components of each realization vector are independently sampled from a gamma distribution with shape 4 and scale 5 (resulting in a "true" mean of 20 and variance of 100 for each demand node). We cap the demand realizations at 50 – if a generated demand realization exceeds 50, we discard the value and generate a new realization. The same reference set is used for models applied to the same network.

We sample the arc capacity costs and arc flow costs independently from $10U$, and sample the unmet demand penalty at each demand node independently from $2.5|N|^2U$, where $|N|$ is the number of nodes in the network and $U$ is a uniform random variable from 0 to 5. It should be noted that the demand penalty increases with the number of nodes in the network to compensate for the increased cost of supplying to demand nodes, as the number of supply nodes and demand nodes remains unchanged for different sized networks.

To set up DR-NDP, we set $\underline{d}_i = 0$ and $\overline{d}_i = 50$ for all $i \in D$, the values of which correspond to the lower bound the gamma distribution and the hard upper cap that we imposed when generating the reference set. We chose to use $N_i = 25$ intervals for each $i \in D$, giving us the cut points of $0, 2, \ldots, 50$ for the piecewise approximation of the quadratic objective in (3.28). In the algorithm, we used a tolerance value of $\epsilon = 0.02$ for all our experimental runs. For the SAA-based model, we assumed either a uniform or gamma distribution (see Section 3.5.1 for distribution parameter values), and generated 50 and 1,000 realizations in separate models, giving four different models for the benchmark. In our results, these four models are labeled "uniform50", "uniform1000", "gamma50" and "gamma1000", with the prefix indicating the assumed distribution family and the suffix indicating the number of generated realizations.

We draw 20 realizations from the reference set as our initial small set of observations, and run DR-NDP and the four benchmark models on the 20 observations. For each reference set, we draw three sets of 20 observations. The first *normal* observed demand set is drawn such that each marginal mean of the observations in the set deviates no more than $7.5\%$ from that of the reference set. The second *very low* observed demand set comprises the 20 realizations in the reference set with the lowest total demand. The third *low* observed demand set is drawn such that difference between the total marginal means of the set and that of the very low demand set is at most $70\%$ of the difference between the total true marginal means and that of the very low demand set. We did not sample high demand cases, as it would be likely that all models would perform well by planning for more de-

mand than what is actually realized. For ease of reading, the results for the three set are ordered by "normal", "low", then "very low" demand in our results.

Finally, to evaluate the performance of the $x$ solutions generated by each of the five models, we solve the capacitated minimum cost flow problem (3.1)–(3.6), with capacities as given by $x$, for each of the 10,000 demand vectors in the reference set. The results are then averaged to obtain their expected values (e.g. expected flow cost, expected penalty cost).

### 3.5.3 Analysis of Results

We give an analysis of the computational results when using grid networks. In our results, we look at three main components of the results: capacity cost, flow cost, and penalty cost. Readers are reminded that the flow cost and penalty cost as computed by the models and by using the capacity solution $x$ with the reference set (see Section 3.5.2) are different, and we refer to the latter case by *expected* flow cost and penalty cost.

#### 3.5.3.1 General Observations on Solution Performance

Table 3.1 gives the expected cost components and expected total cost for each observed demand level and model used on the 3x3 grid network, which we use as a representative network for our observations on solution performance.

Table 3.1: Cost components in objective of 3x3 grid network

| Demand | Model | Capacity cost | $\mathbb{E}$[Flow cost] | $\mathbb{E}$[Penalty cost] | $\mathbb{E}$[Total cost] |
|---|---|---|---|---|---|
| normal | drndp | 4,730.17 | 2,474.93 | 3,361.01 | 10,566.12 |
| normal | gamma50 | 5,040.10 | 2,505.70 | 2,795.68 | 10,341.48 |
| normal | gamma1000 | 4,775.20 | 2,570.70 | 2,945.14 | 10,291.03 |
| normal | uniform50 | 4,998.70 | 2,593.56 | 2,806.04 | 10,398.31 |
| normal | uniform1000 | 4,999.90 | 2,578.93 | 2,752.63 | 10,331.47 |
| low | drndp | 4,030.68 | 2,226.96 | 4,993.53 | 11,251.18 |
| low | gamma50 | 4,560.20 | 2,306.21 | 4,723.42 | 11,589.83 |
| low | gamma1000 | 4,181.10 | 2,320.49 | 4,547.73 | 11,049.32 |
| low | uniform50 | 4,311.30 | 2,381.18 | 4,321.74 | 11,014.22 |
| low | uniform1000 | 4,443.10 | 2,406.43 | 4,001.44 | 10,850.97 |
| very low | drndp | 1,645.20 | 1,277.19 | 14,511.62 | 17,434.01 |
| very low | gamma50 | 1,497.90 | 1,154.45 | 16,455.75 | 19,108.10 |
| very low | gamma1000 | 1,597.00 | 1,180.39 | 15,626.80 | 18,404.20 |
| very low | uniform50 | 1,527.10 | 1,117.44 | 16,583.06 | 19,227.60 |
| very low | uniform1000 | 1,580.20 | 1,149.79 | 15,963.94 | 18,693.93 |

In Table 3.1, we observe that capacity cost is closely related to the total mean demand of the observations. Regardless of the model used, the lower the total mean demand of the observations, the less capacity will be planned for the network, and the lower the capacity cost will be. Capacity cost and expected penalty are also closely related quantities – when capacity cost is high, expected penalty is low. This is evident in the much higher expected penalty cost corresponding to the lower capacity cost for the very low demand set. This is an expected relationship, since a lower capacity cost implies less capacity on the network to flow commodities, resulting in less demand being satisfied and more unmet demand. This is also supported by the low flow cost for the very low demand set, implying less commodity flowing on average when compared to the solutions when using observations with normal and low total mean demand.

Note that while the performance of the solutions when using 50 generated realizations is different from that when using 1,000 generated realizations, they are fairly similar, but with arbitrary relationship with one another (either could be larger than the other). Hence, in our subsequent tables, we will only show the results for "gamma1000" and "uniform1000", which are presumably a better representation of the results for the SAA-based model due to a more accurate representation of the assumed distribution. However, it should be noted that solving 1,000 generated realizations will be slower than solving with 50 generated realizations (see Section 3.5.3.5).

#### 3.5.3.2 Sensitivity to Observed Demand Level

Table 3.2 gives the expected cost components and expected total cost for each observed demand level and model used on the 5x5 and 7x7 grid networks, which we will use in conjunction with Table 3.1 to illustrate the main benefit of using DR-NDP: insensitivity of solution performance to the initial set of observed realizations.

Unsurprisingly, assuming a gamma distribution results in the lowest expected total cost for the normal demand set. This is because the assumed distribution in this case most closely resembles the true distribution, giving the best quality solutions. This remains true even when the total mean demand of the observations is low.

However, this may not be the case when the observations have very low demand. In some networks, DR-NDP has lower expected total cost than the SAA-based model under a gamma assumption. This supports our claim that DR-NDP produces conservative solutions that safeguard against cases where the set of observations is not representative of the true distribution.

Furthermore, DR-NDP is relatively insensitive to the total mean demand of the initial observations. This is most evident when comparing the expected total cost of the solutions.

Table 3.2: Cost components in objective of 5x5 and 7x7 grid network

| Network | Demand | Model | Capacity cost | $\mathbb{E}$[Flow cost] | $\mathbb{E}$[Penalty cost] | $\mathbb{E}$[Total cost] |
|---|---|---|---|---|---|---|
| Grid 5x5 | normal | drndp | 6,193.99 | 4,236.08 | 21,005.01 | 31,435.07 |
| Grid 5x5 | normal | gamma1000 | 6,971.70 | 4,353.00 | 16,247.94 | 27,572.64 |
| Grid 5x5 | normal | uniform1000 | 6,255.70 | 4,292.96 | 17,464.34 | 28,013.00 |
| Grid 5x5 | low | drndp | 5,527.98 | 4,171.82 | 21,070.91 | 30,770.71 |
| Grid 5x5 | low | gamma1000 | 5,904.40 | 4,259.00 | 17,990.02 | 28,153.42 |
| Grid 5x5 | low | uniform1000 | 5,173.40 | 4,176.68 | 21,642.24 | 30,992.32 |
| Grid 5x5 | very low | drndp | 4,694.49 | 4,061.02 | 26,147.14 | 34,902.65 |
| Grid 5x5 | very low | gamma1000 | 3,835.00 | 3,973.93 | 34,504.50 | 42,313.44 |
| Grid 5x5 | very low | uniform1000 | 2,935.80 | 3,513.28 | 62,653.05 | 69,102.13 |
| Grid 7x7 | normal | drndp | 4,915.98 | 2,235.85 | 85,559.51 | 92,711.34 |
| Grid 7x7 | normal | gamma1000 | 7,644.40 | 2,452.86 | 57,062.42 | 67,159.69 |
| Grid 7x7 | normal | uniform1000 | 5,589.90 | 2,383.54 | 61,701.22 | 69,674.66 |
| Grid 7x7 | low | drndp | 4,713.25 | 2,160.79 | 85,383.54 | 92,257.57 |
| Grid 7x7 | low | gamma1000 | 5,434.40 | 2,401.82 | 61,170.01 | 69,006.22 |
| Grid 7x7 | low | uniform1000 | 4,312.20 | 2,233.40 | 80,840.05 | 87,385.65 |
| Grid 7x7 | very low | drndp | 4,716.65 | 2,269.03 | 79,744.76 | 86,730.43 |
| Grid 7x7 | very low | gamma1000 | 5,328.50 | 2,509.00 | 72,035.93 | 79,873.43 |
| Grid 7x7 | very low | uniform1000 | 3,650.50 | 2,276.27 | 127,896.90 | 133,823.67 |

Figure 3.3 gives the expected total cost of the solution for each model as a percentage of that of the same model when using a set of observations with normal demand.

For the 7x7 grid network, DR-NDP gives an expected total cost of $92,711$1.34 with a normal level of demand and gives an expected total cost of $86,730.43$ with a very low level of demand, a change of only $6\%$. In contrast, "gamma1000" has a change of $19\%$ in the same comparison. For the 5x5 grid network, DR-NDP has a change of $11\%$, while "gamma1000" has a change of $53\%$; for the 3x3 grid network, DR-NDP has a change of $65\%$, while "gamma1000" has a change of $79\%$. The decrease in the percentages as network size increases further suggests that the sensitivity of the expected total cost is affected by the size of the network.

It should be noted that while the biggest component in the total cost is the penalty cost, that the expected total cost for the solutions of DR-NDP is relatively insensitive to the total mean demand of the observations is not solely because the expected penalty cost is relatively insensitive. Even though expected flow cost is generally the same for all observed demand levels and models for the same network, capacity cost does also appear less sensitive to the total demand, and contributes to the overall stability of the expected total cost (to a lesser degree than the expected penalty cost).

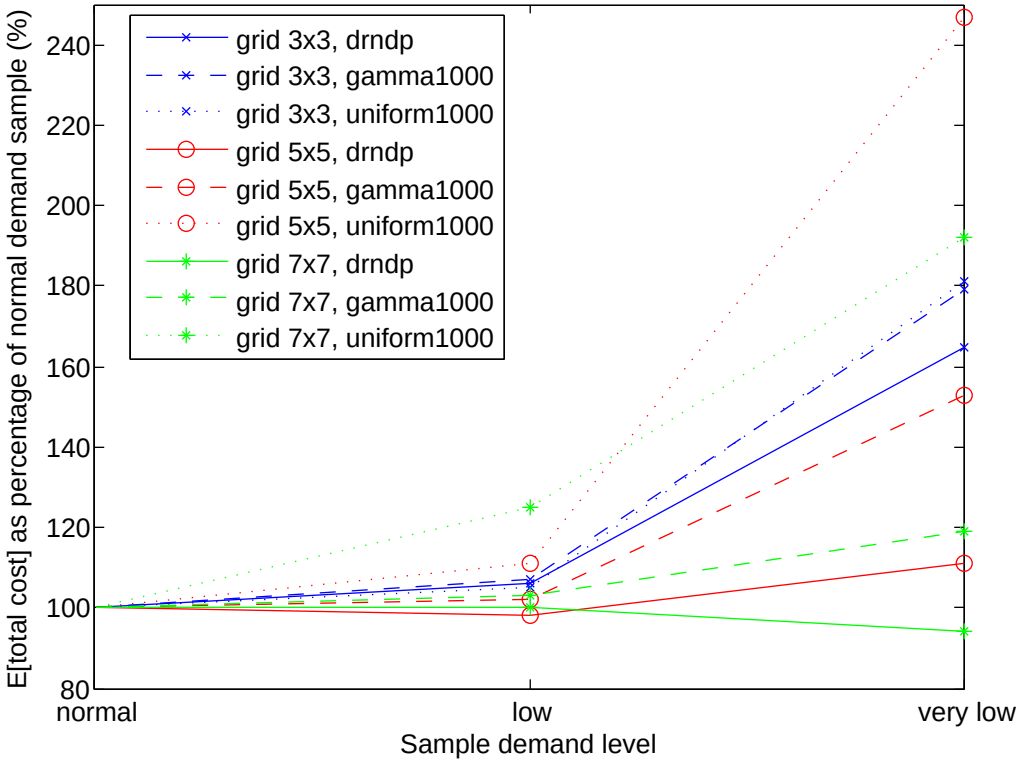



Figure 3.3: Relative expected total costs for different observed total mean demand levels

#### 3.5.3.3 Comparison of Real-world Network and Grid Network

We compare a real-world network, the NOBEL-US network described in Section 3.5.2, against a grid network of similar size to determine if a less structured network topology gives the same results. We use a 4x4 grid network, with 16 nodes and 48 arcs, for comparison with NOBEL-US, which has 14 nodes and 42 arcs. The expected cost components and expected total cost for each of the observed demand levels and models used these two networks are given in Table 3.3.

When comparing the two networks, there is a striking difference between the penalty costs, resulting in a fairly large difference in the expected total costs. Despite the 4x4 grid network being slightly larger than NOBEL-US, the expected total costs for NOBEL-US with a normal and low observed level of demand is approximately 30% more than those for the 4x4 grid network. Importantly, the expected total costs for NOBEL-US for all models are less sensitive to the observed demand level, with DR-NDP being the least sensitive of the models. This suggests that DR-NDP may be more suited to solve networks with less structured topologies.

Table 3.3: Cost components in objective of NOBEL-US and 4x4 grid network

| Network | Demand | Model | Capacity cost | $\mathbb{E}$[Flow cost] | $\mathbb{E}$[Penalty cost] | $\mathbb{E}$[Total cost] |
|---|---|---|---|---|---|---|
| NOBEL-US | normal | drndp | 3,484.15 | 2,047.28 | 24,641.67 | 30,173.10 |
| NOBEL-US | normal | gamma1000 | 5,172.70 | 2,402.14 | 16,175.56 | 23,750.40 |
| NOBEL-US | normal | uniform1000 | 5,300.80 | 2,386.56 | 16,150.45 | 23,837.81 |
| NOBEL-US | low | drndp | 3,380.12 | 2,054.57 | 25,142.06 | 30,576.75 |
| NOBEL-US | low | gamma1000 | 4,606.20 | 2,337.80 | 17,629.49 | 24,573.50 |
| NOBEL-US | low | uniform1000 | 4,365.20 | 2,241.08 | 19,161.49 | 25,767.77 |
| NOBEL-US | very low | drndp | 3,094.48 | 1,973.34 | 28,889.95 | 33,957.78 |
| NOBEL-US | very low | gamma1000 | 2,698.80 | 1,766.28 | 41,131.60 | 45,596.68 |
| NOBEL-US | very low | uniform1000 | 2,247.50 | 1,556.56 | 52,296.97 | 56,101.04 |
| Grid 4x4 | normal | drndp | 3,325.40 | 3,022.78 | 13,043.65 | 19,391.84 |
| Grid 4x4 | normal | gamma1000 | 3,987.20 | 3,052.59 | 9,678.73 | 16,718.53 |
| Grid 4x4 | normal | uniform1000 | 3,954.40 | 3,025.63 | 10,335.57 | 17,315.59 |
| Grid 4x4 | low | drndp | 3,190.53 | 2,985.49 | 13,412.88 | 19,588.89 |
| Grid 4x4 | low | gamma1000 | 3,603.40 | 2,982.62 | 11,286.14 | 17,872.15 |
| Grid 4x4 | low | uniform1000 | 3,357.70 | 2,906.60 | 13,860.95 | 20,125.25 |
| Grid 4x4 | very low | drndp | 1,876.82 | 2,174.60 | 30,767.12 | 34,818.54 |
| Grid 4x4 | very low | gamma1000 | 1,591.60 | 1,567.57 | 39,838.19 | 42,997.37 |
| Grid 4x4 | very low | uniform1000 | 1,406.70 | 1,513.28 | 44,215.69 | 47,135.66 |

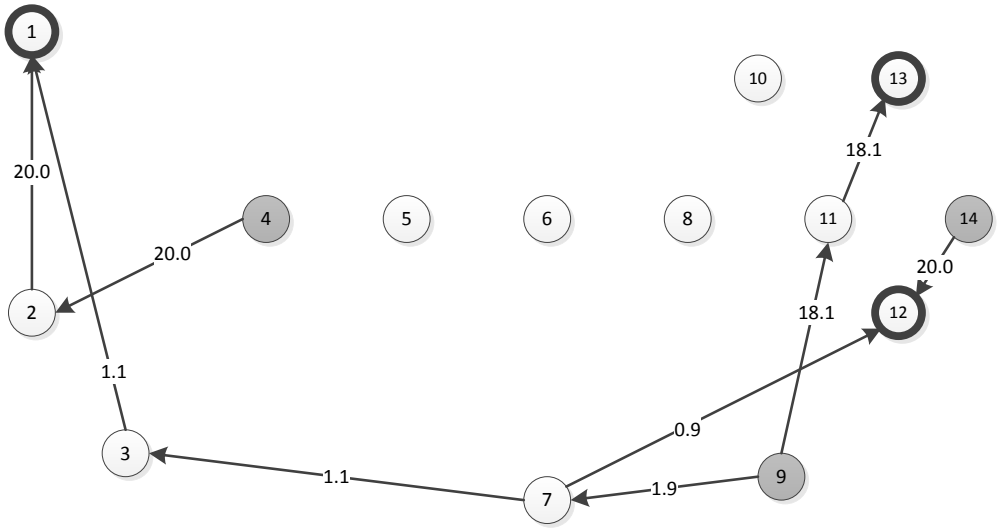
#### 3.5.3.4 Arc Capacity Allocations

Here, we compare the first-stage capacity solution vectors $x$ for DR-NDP and "gamma1000", the latter of which we use as a representative model for the SAA-based model. We use the NOBEL-US network, with a very low level of observed demand. For convenience, we use numeric labels for the nodes in the network as follows:

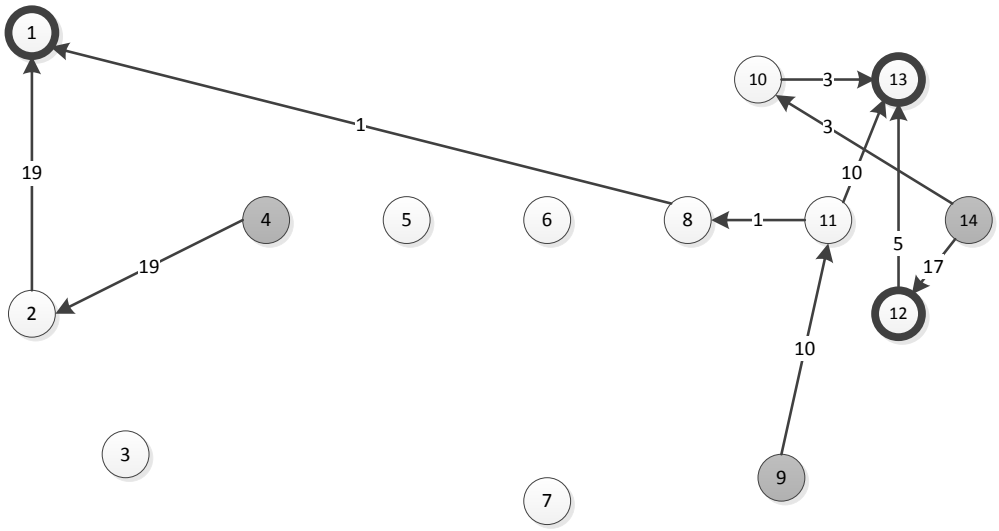
1. Seattle
2. Palo Alto
3. San Diego
4. Salt Lake City
5. Boulder
6. Lincoln
7. Houston
8. Urbana-Champaign
9. Atlanta
10. Ann Arbor
11. Pittsburgh
12. Washington
13. Ithaca
14. Princeton

Figure 3.4 illustrates the solutions given by DR-NDP and "gamma1000", with the former at the top of the figure and the latter at the bottom. The supply nodes are 4, 9 and 14 (shaded in gray), while the demand nodes are 1, 12 and 13 (circled in bold), with mean observed demand $9.0$, $8.5$ and $8.3$ respectively, lower than the "true" mean demand of $20$ at each node (see Section 3.5.2). The allocated arc capacities are given by the numbers on the arcs. Arcs with zero capacity are not shown.

When comparing the two solutions, it is apparent that "gamma1000" will perform worse in terms of demand satisfaction as only a maximum of $19 + 10 + 17 + 3 = 49$




(a) DR-NDP

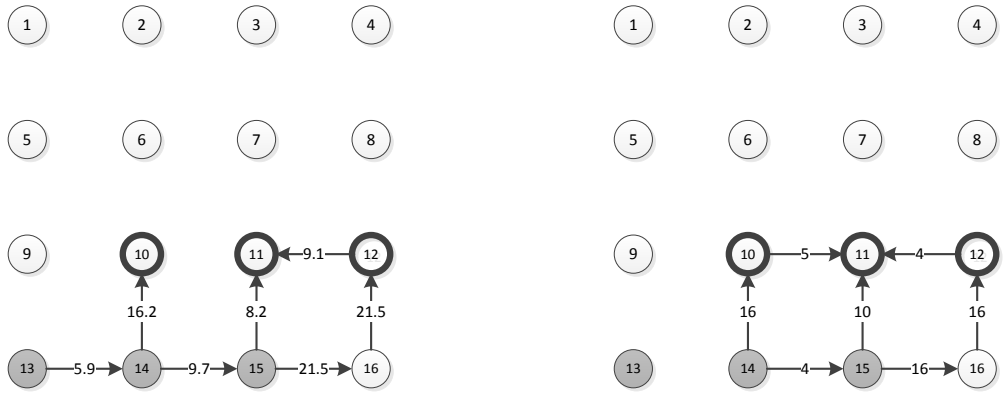



(b) “gamma1000”

Figure 3.4: Capacity solutions obtained for NOBEL-US for very low total mean observed demand

units of commodity can be supplied with its solution. This is less than the true total mean demand of $60$, as the solution was planned for a much lower total mean demand of $9.0 + 8.5 + 8.3 = 25.8$.

In contrast, DR-NDP produces a more conservative solution, with greater capacity provided on arcs leading out of the supply nodes and into the demand nodes. All $20.0 + 1.9 + 18.1 + 20.0 = 60$ units of commodity can supplied if required, although the maximum demand that can be satisfied at each demand node is not necessarily equal to its true mean. In particular, nodes 1, 12 and 13 can receive up to $21.1$, $20.9$ and $18.1$ units of commodity respectively. Also of note in the solutions is that the capacities given by DR-NDP can have fractional parts, as opposed to the integer capacities that are always given by using the SAA-based models.




(a) DR-NDP

(b) "gamma1000"

Figure 3.5: Capacity solutions obtained for 4x4 grid for very low total mean observed demand

We perform the same comparison on the 4x4 grid network, with the arc allocations shown in Figure 3.5. For this network, nodes 13, 14 and 15 are the supply nodes, while nodes 10, 11 and 12 are the demand nodes with respective mean demands of $7.9$, $9.4$ and $8.75$. The maximum amount of commodity that can be supplied for DR-NDP is $16.2 + 8.2 + 21.5 = 45.9$ and the maximum demand that can be satisfied is also $45.9$. Similar to NOBEL-US, these values for DR-NDP are higher than those for "gamma1000", which supplies a maximum of $40$ units of commodity (note that it is not $16 + 10 + 16 = 42$ as node 13 is unable to supply any commodity) and satisfies a maximum of $42$ units of demand, resulting in a better performance for DR-NDP than "gamma1000" in the case of very low demand.

The solutions for DR-NDP and "gamma1000" are similar, both in the placement of arcs (with positive capacity) and in the capacity allocated to the arcs. In contrast, the solutions for the two models for NOBEL-US are almost entirely different in placement, and with very different allocated capacities. This is because the structured topology of a grid network allows less variation in flow paths from supply to demand nodes, resulting in similar solutions regardless of the model used. This is also consistent with our earlier claim that solutions are generally more sensitive to demand levels in less structured networks, and using DR-NDP may reduce this sensitivity.

#### 3.5.3.5 Solution Time

We compare the solution times for varying demand levels and network sizes. Table 3.4 describes the growth of solution times for each of the five models as the size of the network increases from a 3x3 grid to a 7x7 grid and for the NOBEL-US network. The reader should be aware that there is a minor overhead for generating realizations and reading them in for the SAA-based model, approximately 8 milliseconds for 50 realizations and 90 milliseconds for 1,000 realizations, which we did not include in our results.

Table 3.4 shows an increase in solution times for all five models as network size increases. Generating 50 realizations offers a vast improvement in solution time over generating 1,000 realizations, typically taking approximately 1%-2% of the time. The DR-NDP takes significantly longer than any of the benchmark models, for all grid network sizes from 3x3 to 7x7. However, the growth of the solution time for DR-NDP appears to be slower compared to that of the benchmarks. This indicates that the solution time difference could be smaller for much larger networks. Topological structure also does not seem to be a large factor in solution times, with the 4x4 grid network having slightly longer solve times than NOBEL-US – this difference can be attributed to the 4x4 grid network being slightly larger.

## 3.6 Concluding Remarks

We proposed a novel approach to solving NDPs under demand uncertainty. Utilizing distributionally robust techniques, DR-NDP aims to conservatively solve NDPs with scarce distributional data, yet ensure that solutions are not overly conservative (as in the case of robust optimization). We approximated DR-NDP as an optimization problem with an embedded problem in its constraints, and developed a cutting-plane algorithm that leverages the structure of the approximate reformulation to iteratively generate cuts.

Our results showed that the solutions generated by DR-NDP were conservative, and,

Table 3.4: Solution times (in milliseconds)

| | | Demand | | |
|---|---|---|---|---|
| Network | Model | normal | low | very low |
| Grid 3x3 | drndp | 62,280 | 42,800 | 7,424 |
| Grid 3x3 | gamma50 | 15 | 15 | 10 |
| Grid 3x3 | gamma1000 | 2,181 | 1,895 | 1,325 |
| Grid 3x3 | uniform50 | 20 | 10 | 10 |
| Grid 3x3 | uniform1000 | 1,665 | 1,431 | 1,395 |
| NOBEL-US | drndp | 110,928 | 103,113 | 71,919 |
| NOBEL-US | gamma50 | 25 | 20 | 10 |
| NOBEL-US | gamma1000 | 4,566 | 2,605 | 1,240 |
| NOBEL-US | uniform50 | 36 | 25 | 15 |
| NOBEL-US | uniform1000 | 4,571 | 3,870 | 860 |
| Grid 4x4 | drndp | 144,597 | 115,689 | 43,233 |
| Grid 4x4 | gamma50 | 27 | 25 | 10 |
| Grid 4x4 | gamma1000 | 6,236 | 4,421 | 845 |
| Grid 4x4 | uniform50 | 40 | 35 | 10 |
| Grid 4x4 | uniform1000 | 5,796 | 5,731 | 1,200 |
| Grid 5x5 | drndp | 185,190 | 294,715 | 185,899 |
| Grid 5x5 | gamma50 | 80 | 65 | 35 |
| Grid 5x5 | gamma1000 | 11,642 | 9,747 | 4,720 |
| Grid 5x5 | uniform50 | 120 | 75 | 30 |
| Grid 5x5 | uniform1000 | 15,167 | 13,602 | 3,956 |
| Grid 6x6 | drndp | 531,350 | 463,441 | 462,732 |
| Grid 6x6 | gamma50 | 240 | 125 | 95 |
| Grid 6x6 | gamma1000 | 42,977 | 20,859 | 11,607 |
| Grid 6x6 | uniform50 | 255 | 170 | 65 |
| Grid 6x6 | uniform1000 | 36,025 | 40,061 | 14,237 |
| Grid 7x7 | drndp | 453,094 | 507,749 | 467,700 |
| Grid 7x7 | gamma50 | 585 | 330 | 290 |
| Grid 7x7 | gamma1000 | 45,573 | 26,434 | 18,098 |
| Grid 7x7 | uniform50 | 701 | 360 | 275 |
| Grid 7x7 | uniform1000 | 114,202 | 36,295 | 96,905 |

crucially, were insensitive to the observed level of demand. However, the conservativeness of the solutions comes at a high computational cost. Even though the growth of solution time with network size appears to be slow, this suggests that DR-NDP is more suited for niche uses, in which data is scarce, but yet the network planner can ill afford to have too much unmet demand due to "bad" data.

Given the niche uses of DR-NDP, it will be instructive to apply DR-NDP to real-world problems. A possible application that could fit this specific use is humanitarian relief supply network design. The data scarcity due to the unknown reach of the disaster, and the need for high demand satisfaction to aid as many people as possible, make such a problem a possible candidate for future application. In addition to discovering if DR-NDP is suitable for such specific types of problems, studying DR-NDP with real-world problems will give much insight into the performance of DR-NDP with networks with less structured topologies.

# CHAPTER 4

# Optimizing Profitability and QoS of Carsharing Systems

## 4.1 Introductory Remarks

This chapter applies a network design framework to carsharing systems. This framework utilizes a spatial-temporal network to approximate the movement of vehicles in a carsharing fleet. The first stage of the model determines the allocation of shared car fleet to pre-designated zones, and the number of contracted parking lots to rent or free-float permits to obtain in these zones. The second stage solves a stochastic minimum cost flow problem on the spatial-temporal network, which integrates one-way and round-trip rentals, as well as ad-hoc vehicle relocation into the same model, to optimize total profits less any penalties from unsatisfied demand. Additionally, we consider a risk-averse variant of the second-stage model that penalizes the CVaR of unsatisfied demand.

### 4.1.1 Problem Description

We consider allocating a carsharing fleet in a region serviced by a carsharing company to satisfy uncertain travel demand. The region is discretized into smaller zones, with parking costs different from zone to zone. To regulate carsharing companies, city governments issue parking lot contracts and free-float parking permits for shared cars. Parking lot contracts grant companies exclusivity to lots, and are typically used by reservation-based carsharing systems, for which the carsharing company takes into account parking capacity when accepting rental demand. On the other hand, we consider an alternative as purchasing free-float parking permits, which are often used by free-floating carsharing systems, whose vehicles can be parked at any available city parking lot. Note that contracted parking lots can also be applicable for free-floating based customers, while the free-float parking permits may not be realistic for reservation-based systems, since it is hard to appoint customers

who made the reservations to parked shared cars in an online fashion. For the problem of interest, we are tasked to allocate a homogeneous fleet of cars to the zones at the start of a planning horizon, and to determine the number of parking lots to purchase in each zone (for reservation-based and/or free-floating systems) or the number of free-float permits to purchase overall (for free-floating systems).

The above are the "here-and-now" decisions in the first stage, and afterwards customer demand for one-way and round rentals is realized in the second stage. Cars are unavailable while being used by customers, and should the demand for vehicles in a zone exceed the number of cars available for use at that zone, any excess demand is immediately lost (we assume that any excess demand is not carried over to the next period). During the planning horizon, we may relocate the vehicles as a recourse action, but similar to rented cars, customers will be unable to use cars that are being relocated. We aim to find the optimal first-stage decisions to maximize the expected profit less the costs associated with the QoS in the second-stage.

### 4.1.2 Methodology Overview

We formulate a two-stage stochastic optimization problem. Given the first-stage decisions as the number of parking lots or free-float permits to purchase and the initial assignment of the cars, we model the movement of cars from zone to zone in the second stage as flows on a spatial-temporal network, with each node in the network representing the state of a zone at a point in time during the planning horizon.

We point out three advantages of using such a model. Firstly, vehicle movement can be easily represented. Since each zone is replicated by the number of periods in the planning horizon, it is straightforward to represent vehicle movement as flows that are conserved between the spatial-temporal nodes, and to keep track of the overall status of the vehicles (e.g. whether they are in use or, if available, where they are located). Secondly, the second-stage problem for each demand scenario is a minimum cost flow problem, which is a well-solved problem (Ahuja et al., 1993). Finally, the actual topology of the road network connecting the zones is of secondary importance to this model, as only the travel times between each pair of zones is required to construct the network. This means the construction of the spatial-temporal network can be driven entirely by past customer usage data, through analysis of typical travel times between zones – the positioning of the zones relative to one another would not matter to the spatial-temporal network. Furthermore, the travel time between a pair of nodes is more accurate with a higher historical demand of travel between them. Hence, a less accurate travel time would unlikely be used in the optimization simply

because there will be lower demand of travel between the particular pair of nodes.

To manage QoS, we penalize unserved customers via two different models. Firstly, we formulate a risk-neutral model that penalizes the expected number of unserved customers. Secondly, we formulate a risk-averse model that penalizes the conditional value-at-risk of the number of unserved customers. Regardless of the model used, the objective aims to achieve the optimal balance between high profit and high QoS (by keeping the number of unserved customers low).

However, a spatial-temporal network constructed for the above carsharing problem consists of a large number of nodes and arcs, as a result of the large number of locations of a real-world problem, the large number of periods in the planning horizon due to the need for finer time period granularity, or both. Consequently, there will be a large number of variables in the second-stage problem. We use the Benders decomposition approach (cf. Benders, 1962; Van Slyke and Wets, 1969) to generate cuts as needed to a relaxed master problem. Moreover, since the first-stage decision variables are integers, the Benders cuts are generally weak because of the weaker master problem where the integer constraints are relaxed. We further generalize a branch-and-cut procedure with mixed-integer rounding (MIR) developed in Bodur and Luedtke (2014). In this procedure, pairs of Benders cuts are used to derive stronger cuts via MIR.

### 4.1.3 Literature Review

The use of carsharing as a means of transportation has increased tremendously. Shaheen et al. (1998) and Shaheen and Cohen (2007) provide an excellent review of the history and recent rapid growth of the carsharing industry. Katzev (2003) explores the early adoption processes of several carsharing systems, and also evaluates the effects of carsharing on commuter mobility behavior and the environment. Benefits to the latter is an oft-quoted reason for the increased popularity of carsharing. Zhou (2015) considers a network design problem for urban transportation systems with one-way carsharing rentals, and formulates integer programming models for designing carsharing systems with an aim of minimizing the total traveling time and reducing congestion and emissions.

Given fixed carsharing systems and transport networks, how to relocate and redistribute cars during operation is a major consideration for satisfying customer demand. Optimizing the relocations can be a complex process as the current fleet distribution and current demand spread must both be taken into consideration. Weikl and Bogenberger (2013) summarize and categorize several strategies used by carsharing companies to relocate their fleet, with examples from real-life carsharing systems. In addition to vehicle redistribution, Pfrommer

et al. (2014) and Febbraro et al. (2012) also suggest real-time price incentives as a means to shape demand to reduce the need for excessive vehicle relocations.

Representing vehicle movement in a spatial-temporal network has previously been used by Kek et al. (2009) to determine a set of nearly optimal manpower and operating parameters to satisfy given relocation needs, by de Almeida Correia and Antunes (2012) to optimize the placement of zones in a network, and by Fan (2014) to optimize the allocation of vehicles to zones in one-way carsharing systems. However, there has been no attempt to integrate one-way with round-trip rentals in the same model, or to consider both reservation-based and free-floating carsharing systems in a generalized model. Furthermore, the difficulty of handling a large spatial-temporal network was not addressed in the above literature, for which we propose an effective branch-and-cut algorithm with strengthened valid inequalities, which shows promising computational performance when being implemented in parallel computing.

### 4.1.4 Main Contributions

Many papers on the redistribution of a carsharing fleet focus on the operational considerations of the system, aiming to find relocation solutions that satisfy the rental needs in time. In this chapter, we focus on a more strategic problem of purchasing parking lots and free-float permits together with car fleet allocation in zones. We are interested in the profitability and the QoS under demand uncertainty, and also explore the use of a risk-averse measure CVaR to quantify QoS, which is not found in related literature.

We apply our models to the carsharing fleet allocation problem with real-world data. Our numerical results show that the proportion of one-way trips can significant impact on the profitability and the QoS. As this proportion increases, the net profit decreases drastically. Its impact on the QoS, on the other hand, shows a non-monotone pattern. Presence of a small proportion of one-way trips can actually help lowering the number of unfulfilled demand. However, as this proportion further increases, the number of unfulfilled demand will increase. The detriments of one-way rentals are mainly from the loss of rental hours and the need for vehicle relocation. Carsharing companies have been trying to remedy revenue loss by charging higher rates for one-way trips. Similarly, we study the effect of lower vehicle relocation cost, which is approximately equivalent to increasing the pricing of both one-way and round-trip rentals (barring significant idling or maintenance costs). We find that lower relocation cost has different impacts on profitability and QoS. While it only results in slightly higher profits, it can significantly improve the QoS by eliminating most of unfulfilled demand. These results provide valuable insights and recommendations

to city governments who often rely on and sponsor carsharing programs as a means to solve their public transportation problems.

### 4.1.5 Structure of Chapter

The remainder of this chapter is organized as follows. Section 4.2 describes the construction of the spatial-temporal network and provides the two-stage optimization models, while Section 4.3 presents an efficient method for solving these models using a branch-and-cut algorithm with MIR-enhanced Benders cuts. Section 4.4 provides the insights drawn from applying the models on real-world data and evaluates the computational efficiency of the proposed algorithm. Section 4.5 briefly describes some extensions to proposed carsharing model, and Section 4.6 presents conclusions and future research directions.

## 4.2 Problem Formulation

Given a budget of $S$ cars to place in a set of zones, $I$, we maximize profit and QoS over $T$ time periods. We denote the number of parking lots to purchase in zone $i$ by $w_i$ and the initial number of cars to place in zone $i$ by $x_i$. Each parking lot purchased in zone $i \in I$ incurs a one-time cost of $c_i^{\text{lot}}$ and each free-float permit purchased incurs a one-time cost of $c^{\text{ffp}}$. Any additional costs of allocating a car to zone $i$ is captured by $c_i^{\text{loc}}$. We denote the demand for one-way rentals from zone $i$ to zone $j$ starting at period $t$ by $d_{ijt}^{\text{one}}$, and denote the demand for round-trip rentals from zone $i$ starting at period $t$ and ending at period $s$ by $d_{its}^{\text{two}}$. We also denote the time taken to travel from zone $i$ to zone $j$ by $\ell_{ij}$.

As carsharing companies typically use a time-based payment scheme, we assume that cost and revenue parameters for rentals are independent of zones but dependent on the amount of time a car is used. The revenue generated comes solely from customers using vehicles, while costs incurred by the company include the cost of relocating vehicles (to potentially satisfy demand elsewhere), the cost of maintenance due to wear and tear from car usage, and the cost of vehicles idling (such as opportunity costs and depreciation costs). We denote the revenue per period per car from one-way rentals by $r^{\text{one}} \geq 0$ and that from round-trip rentals by $r^{\text{two}} \geq 0$, while the relocation cost per period per car is denoted by $c^{\text{rel}} \geq 0$. When a car is in use, whether during one-way rentals, round-trip rentals, or relocation, it incurs a maintenance cost of $c^{\text{mnt}} \geq 0$ per period; when it is not in use, it incurs an idle cost of $c^{\text{idle}} \geq 0$ per period.

### 4.2.1 Construction of the Spatial-Temporal Network

To model the movement of vehicles from zone to zone over the planning horizon, we construct a spatial-temporal network $G(N, A)$, with each node $n_{it} \in N$ representing a zone $i \in I$ at period $t \in \{0, 1, 2, \ldots, T\}$. The arcs in this network are directed and represent a spatial-temporal movement of vehicles from one zone to another from an earlier period to a later one. There are four types of arcs in the network:

- One-way arcs $(n_{it}, n_{j,t+\ell_{ij}}) \in A^{\text{one}}$ for each $d^{\text{one}}_{ijt} > 0$, with capacity $d^{\text{one}}_{ijt}$ and cost $-(r^{\text{one}} - c^{\text{mnt}})\ell_{ij}$ per unit flow. Flows on these arcs represent vehicles being rented one-way from zone $i$ to zone $j$ starting from period $t$.

- Round-trip arcs $(n_{it}, n_{is}) \in A^{\text{two}}$ for each $d^{\text{two}}_{its} > 0$, with capacity $d^{\text{two}}_{its}$ and cost $-(r^{\text{two}} - c^{\text{mnt}})(s - t)$ per unit flow. Flows on these arcs represent vehicles being rented round-trip from zone $i$ starting from period $t$ and ending in period $s$.

- Relocation arcs $(n_{it}, n_{j,t+\ell_{ij}}) \in A^{\text{rel}}$ for all pairs of zones $i$ and $j$ and periods $0 \leq t \leq T - \ell_{ij}$, with infinite capacity and cost $(c^{\text{rel}} + c^{\text{mnt}})\ell_{ij}$ per unit flow. Flows on these arcs represent vehicles being relocated from zone $i$ to zone $j$ starting from period $t$.

- Idle arcs $(n_{it}, n_{i,t+1}) \in A^{\text{idle}}$ for each zone $i$ and period $0 \leq t \leq T - 1$, with capacity $w_i$ and cost $c^{\text{idle}}$ per unit flow. Flows on these arcs represent vehicles that are idling at zone $i$ from period $t$ to $t + 1$.

The set $A$ is the union of the four types of arcs described above, i.e. $A = A^{\text{one}} \cup A^{\text{two}} \cup A^{\text{rel}} \cup A^{\text{idle}}$. For convenience, we use arc-based notation subsequently. We denote the unit cost of flow and the capacity of arc $a$ by $f_a$ and $u_a$, respectively, while $\delta^+(n_{it})$ and $\delta^-(n_{it})$ denote the sets of arcs for which $n_{it}$ is their supply or demand node, respectively. The unit flow costs and capacities of each type of arcs are summarized in Table 4.1. In particular, it should be noted that the capacities of idle arcs depend on the vector $w \in \mathbb{Z}^{|I|}_+$ of parking lot purchases $w_i$ with $i \in I$, which we will define formally in Section 4.2.2.

Table 4.1: Unit flow costs and capacities for each arc type

| Type of arc | Cost per unit flow $f_a$ | Capacity $u_a$ |
|---|---|---|
| One-way arc $(n_{it}, n_{j,t+\ell_{ij}})$ | $-(r^{\text{one}} - c^{\text{mnt}})\ell_{ij}$ | $d^{\text{one}}_{ijt}$ |
| Round-trip arc $(n_{it}, n_{is})$ | $-(r^{\text{two}} - c^{\text{mnt}})(s - t)$ | $d^{\text{two}}_{its}$ |
| Relocation arc $(n_{it}, n_{j,t+\ell_{ij}})$ | $(c^{\text{rel}} + c^{\text{mnt}})\ell_{ij}$ | $+\infty$ |
| Idle arc $(n_{it}, n_{i,t+1})$ | $c^{\text{idle}}$ | $w_i$ |

We illustrate the construction of a spatial-temporal network with the following example. Consider two zones labeled A and B that require two periods of time to travel between them. Figure 4.1 shows the corresponding spatial-temporal network over periods $t = 0, \ldots, 3$. Included in the spatial-temporal network is a one-way arc corresponding to a demand of four cars to travel from A to B starting at period 0 (note that the ending period is automatically two periods after), and a round-trip arc corresponding to a demand of two cars picked up at and returned to zone B, starting in period 1 and ending in period 3. The numbers on the two arcs denote the respective capacities.

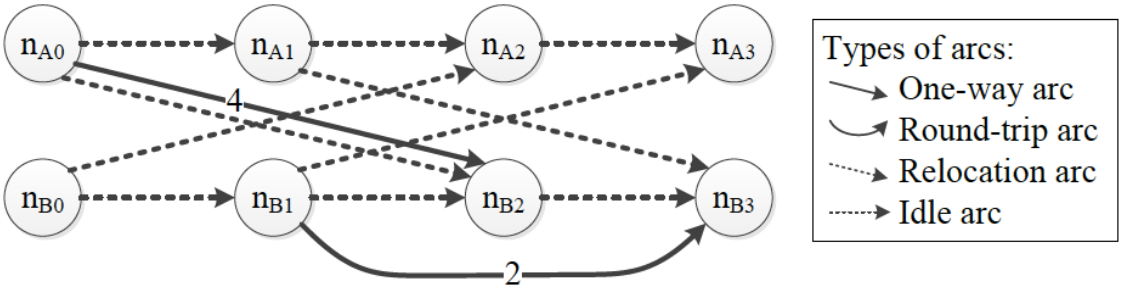



Figure 4.1: Spatial-temporal network example for a two-zone, three-period instance

### 4.2.2 A Two-Stage Stochastic Integer Programming Formulation

We employ two-stage stochastic optimization to model the above problem. In the following formulation, $w \in \mathbb{Z}_+^{|I|}$ denotes the vector of $w_i$'s, $x \in \mathbb{Z}_+^{|I|}$ denotes the vector of $x_i$'s and $u \in \mathbb{Z}_+^{|A|}$ denotes the vector of $u_a$'s.

$$\min_{w,x} \quad \sum_{i \in I} \left( c_i^{\text{lot}} w_i + \left( c^{\text{ffp}} + c_i^{\text{loc}} \right) x_i \right) + Q(w, x) \tag{4.1}$$

$$\text{s.t.} \quad (w, x) \in X = \{ w \in \mathbb{Z}_+^{|I|}, x \in \mathbb{Z}_+^{|I|} : \tag{4.2}$$

$$x_i \leq w_i \qquad \forall i \in I$$

$$\sum_{i \in I} x_i \leq S \}.$$

The set $X$ requires that the number of cars $x_i$ initially deployed to zone $i$ does not exceed the number of parking lots $w_i$ and that the total number of cars does not exceed the given budget of $S$ cars. We minimize the cost of purchasing parking lots, the cost of purchasing free-float permits and any additional cost of allocating cars to their initial zones. The function $Q(w, x)$ returns the optimal cost of the second-stage problem given decisions $w$ and $x$ from

the first stage.

The capacities $u_a$ of the one-way and round-trip arcs in Table 4.1 are random due to the random demand. Given a joint distribution of one-way and round-trip rentals, we employ Monte Carlo sampling to generate a finite number of scenarios. We index the scenarios by $k \in K$, and denote the vector of capacities of the arcs in scenario $k$ by $u^k = [u_a^k,\ a \in A^{\text{one}} \cup A^{\text{two}}]^{\mathsf{T}}$, and the probability of occurrence of scenario $k$ by $p^k$. The second stage optimizes flows in the spatial-temporal network given that the supply level at each node $n_{i0}$ is $x_i$ and the capacity on each arc $a$ is $u_a$ partially determined by $w_i,\ \forall i \in I$. We define the feasible region of shared car movement as

$$
\begin{aligned}
Y(w,x,u) :=& \{y \in \mathbb{R}_+^{|A|} : \\
& \sum_{a\in\delta^+(n_{it})} y_a - \sum_{a\in\delta^-(n_{it})} y_a = \begin{cases} x_i & \text{if } t=0 \\ 0 & \text{if } t=1,\dots,T-1 \\ -x_i & \text{if } t=T \end{cases} \qquad \forall i \in I \qquad (4.3)\\
& y_a \le u_a \qquad \forall a \in A^{\text{one}} \cup A^{\text{two}} \qquad (4.4)\\
& y_a \le w_i \qquad \forall i \in I, a=(n_{it}, n_{i,t+1}) \in A^{\text{idle}}\}, \qquad (4.5)
\end{aligned}
$$

where (4.3) is the flow balance constraint, and (4.4)–(4.5) are the capacity constraints. The flow balance constraints for the spatial-temporal nodes in the last period require that the final allocation of cars be the same as the initial allocation, for the purpose of operating the carsharing system every $T$ periods with the same initial deployment of cars.

In this chapter, we use two different stochastic optimization models for the second-stage problem. Both models minimize the expected cost of flow in the spatial-temporal network plus a measure of the random penalty incurred from unserved customers, so the second-stage value function $Q(w,x)$ in both models is of the form

$$
\begin{aligned}
Q(w,x) = \min_{y^1,\dots,y^{|K|}} \quad & \sum_{k\in K} p^k \sum_{a\in A} f_a y_a^k + g(y^1,\dots,y^{|K|}) \qquad (4.6)\\
\text{s.t.} \quad & y^k \in Y(w,x,u^k) \qquad \forall k \in K. \qquad (4.7)
\end{aligned}
$$

The first term $\sum_{k\in K} p^k \sum_{a\in A} f_a y_a^k$ in the objective denotes the expected flow cost while the second term $g(y^1,\dots,y^{|K|})$ denotes the penalty incurred in all the scenarios. We employ risk-neutral and risk-averse approaches for maintaining a desired QoS level – we describe the related function $g$ in further detail in Sections 4.2.3 and 4.2.4, respectively.

### 4.2.3 A Risk-Neutral Model: Minimizing the Expected Penalty

We first propose a risk-neutral model that minimizes the expected penalty of unserved customers, equal to unused capacities on the one-way and round-trip arcs. We impose penalty $G_a \geq 0$ for each unit of unused capacities on arcs $a \in A^{\text{one}} \cup A^{\text{two}}$, and minimize $\mathbb{E}_{\mathbb{P}}\left[\sum_{a \in A^{\text{one}} \cup A^{\text{two}}} G_a H_a(w,x)\right]$, where $H_a(w,x)$ denotes the unused capacity on arc $a$ given decisions $w$ and $x$, and the expectation $\mathbb{E}_{\mathbb{P}}$ is taken with respect to the probability distribution $\mathbb{P}$ of the uncertain demand. As a result, we specify the second-stage problem $Q(w,x)$ in the risk-neutral model as

$$\begin{aligned} Q_e(w,x) = \sum_{k \in K} p^k \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} G_a u_a^k + \\ \min_y \left\{ \sum_{k \in K} p^k \left( \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} (f_a - G_a) y_a^k + \sum_{a \in A^{\text{rel}} \cup A^{\text{idle}}} f_a y_a^k \right) : \right. \\ \left. y^k \in Y(w,x,u^k),\ k \in K \right\} \end{aligned}$$

with $Q_e(w,x)$ denoting the second-stage value function with expected penalty.

### 4.2.4 A Risk-Averse Model: Minimizing the Penalized CVaR

Conditional value-at-risk (CVaR) is a risk measure employed to cope with loss distributions. It is also known as mean excess loss or mean shortfall for continuously distributed random variables. Its value is dependent on the value-at-risk (VaR) of the same random variable. Given $0 < \epsilon < 1$, we define the $(1-\epsilon)$-VaR (the VaR at confidence level $1-\epsilon$ of the number of unserved customers $\sum_{a \in A^{\text{one}} \cup A^{\text{two}}} H_a(w,x)$) as

$$\text{VaR}_{1-\epsilon}\left( \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} H_a(w,x) \right) = \min \left\{ v : \mathbb{P}\left( \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} H_a(w,x) \leq v \right) \geq 1 - \epsilon \right\}.$$

In other words, when ranking all scenarios $k \in K$ by the number of unserved customers, the $(1-\epsilon)$-VaR is the best value of the $100\epsilon\%$ worst scenarios. It is important to note that in our model, higher values of unserved customers are worse (typically, lower values are worse when computing VaR and CVaR). Then the $(1-\epsilon)$-CVaR is the expected number of unserved customers given that the number exceeds the $(1-\epsilon)$-VaR, or equivalently the average value of the $100\epsilon\%$ worst scenarios.

We propose a risk-averse model, and impose a penalty $G_0$ on the $(1-\epsilon)$-CVaR of unserved customers, with $\epsilon$ being a given risk parameter. Such a model is appropriate

when a company accepts that not providing service to a small number of customers is inevitable but wants to ensure a relatively high QoS on average in the worst-case scenarios. Employing the well-known expectation-based reformulation of CVaR in Rockafellar and Uryasev (2000), we reformulate

$$G_0 \text{CVaR}_{1-\epsilon}\left(\sum_{a\in A^{\text{one}}\cup A^{\text{two}}} H_a(w,x)\right) = G_0 \min_{v\geq 0}\left\{v + \frac{1}{\epsilon}\mathbb{E}_{\mathbb{P}}\left[\left(\sum_{a\in A^{\text{one}}\cup A^{\text{two}}} H_a(w,x) - v\right)^+\right]\right\}, \tag{4.8}$$

where the non-negative variable $v$ denotes $\text{VaR}_{1-\epsilon}\left(\sum_{a\in A^{\text{one}}\cup A^{\text{two}}} H_a(w,x)\right)$. We incorporate (4.8) to replace $g(y^1,\ldots,y^{|K|})$ in the second-stage problem $Q(w,x)$, yielding

$$Q_c(w,x) = \min_{y,z,v\geq 0} \quad \sum_{k\in K} p^k \sum_{a\in A} f_a y_a^k + G_0\left(v + \frac{1}{\epsilon}\sum_{k\in K} p^k z^k\right) \tag{4.9}$$

$$\text{s.t.} \quad y^k \in Y(w,x,u^k),\ z^k \geq 0 \qquad \forall k\in K \tag{4.10}$$

$$z^k \geq \sum_{a\in A^{\text{one}}\cup A^{\text{two}}} (u_a^k - y_a^k) - v \qquad \forall k\in K, \tag{4.11}$$

with $Q_c(w,x)$ denoting the risk-averse second-stage value function based on the CVaR measure.

## 4.3 Solution Approaches

Solving the carsharing model directly can be problematic in both models as there are $O(|I|)$ integer variables $w_i$ and $x_i$ and $O(|A||K|)$ continuous variables $y_a^k$, with $|A|$ potentially being very large should there be a large number of zones or periods (due to a fine granularity of definition). To solve the large mixed-integer linear program more efficiently, we use a branch-and-cut algorithm with cuts enhanced by mixed-integer rounding (MIR) first proposed by Bodur and Luedtke (2014).

In this algorithm, the basic procedure is a branch-and-cut algorithm that branches on the integer variables and solves individual nodes via Benders decomposition; the Benders cuts at each node are added to the master problems of subsequent nodes. However, the Benders cuts may be weak due to the relaxed integer constraints in the first-stage. Consequently, the branch-and-cut algorithm may branch many times before termination. This motivates us to employ the proposed procedure, which applies MIR to pairs of previously generated Benders cuts to obtain stronger valid cuts.

We present our solution algorithm in two parts. Section 4.3.1 decomposes the prob-

lem to solve with a branch-and-cut algorithm and Section 4.3.2 gives improvements to the algorithm using MIR.

### 4.3.1 Benders Decomposition

Benders decomposition (Benders, 1962; Van Slyke and Wets, 1969) is a well-known algorithm often used for decomposing stochastic optimization problems with discrete scenarios into smaller problems corresponding to each scenario. To optimize formulation (4.2) for $Q(w,x) = Q_e(w,x)$ in the risk-neutral model, we formulate a master problem, consisting of the variables not indexed by $k$, and $|K|$ subproblems, consisting of the remaining variables separated by $k$. Each subproblem corresponds to the spatial-temporal network resulting from each scenario in $K$. The Benders approach iteratively generates cuts from each subproblem and adds them to the master problem. In the risk-neutral model, the variables in the master problem are the first-stage variables $w$, $x$ and auxiliary variables $q^k \in \mathbb{R}$ for $k \in K$, denoting the values of the subproblems at optimality. The master problem is thus given by

$$\textbf{MP-Stoch}: \min_{(w,x)\in\widetilde{X},q^1,\ldots,q^{|K|}} \left\{ \sum_{i\in I} \left( c_i^{\text{lot}} w_i + \left( c^{\text{ffp}} + c_i^{\text{loc}} \right) x_i \right) + \sum_{k\in K} p^k q^k : L^k_{\text{Stoch}}(q^k, w, x) \geq 0 \ \forall k \in K \right\},$$

where $\widetilde{X}$ represents $X$ with its integer constraints relaxed and $L^k_{\text{Stoch}}(q^k, w, x) \geq 0$ is the set of cuts generated by **SP-Stoch**$_k(w,x)$, the $k^{\text{th}}$ subproblem of the decomposed risk-neutral model. We formulate the subproblems as the duals of the primal $Q_e(w,x)$, separated by $k \in K$:

$$\begin{aligned}
\textbf{SP-Stoch}_k(w,x): \max_{\pi,\lambda} \quad & \sum_{i\in I} x_i(\pi_{i0} - \pi_{iT}) + \sum_{a\in A^{\text{one}}\cup A^{\text{two}}} u_a^k \lambda_a + \sum_{i\in I} w_i \left( \sum_{a=(n_{it},n_{i,t+1})\in A^{\text{idle}}} \lambda_a \right) \\
\text{s.t.} \quad & \pi_{it} - \pi_{j,t+\ell_{ij}} + \lambda_a \leq f_a - G_a \qquad \forall a = (n_{it}, n_{j,t+\ell_{ij}}) \in A^{\text{one}} \\
& \pi_{it} - \pi_{is} + \lambda_a \leq f_a - G_a \qquad \forall a = (n_{it}, n_{is}) \in A^{\text{two}} \\
& \pi_{it} - \pi_{j,t+\ell_{ij}} \leq f_a \qquad \forall a = (n_{it}, n_{j,t+\ell_{ij}}) \in A^{\text{rel}} \\
& \pi_{it} - \pi_{i,t+1} + \lambda_a \leq f_a \qquad \forall a = (n_{it}, n_{i,t+1}) \in A^{\text{idle}} \\
& \lambda_a \leq 0 \qquad \forall a \in A^{\text{one}} \cup A^{\text{two}} \cup A^{\text{idle}},
\end{aligned}$$

where $\pi_{it}$ and $\lambda_a$ are the dual variables associated with the flow balance constraints (4.3) and the capacity constraints (4.4)–(4.5), respectively. Note that the subproblems for each

scenario differ not only by their objective function, but by their constraints as well, as the arcs in the spatial-temporal network depend on the volume of demand in the scenario.

Similarly, the risk-averse model with $Q(w,x) = Q_c(w,x)$ in model 4.2 can be decomposed into a master problem and $|K|$ subproblems. The master problem consists of $w$, $x$ and $q^k,\ k = 1,\ldots,|K|$, as in the risk-neutral model, and a scalar variable $v$ as the $(1-\epsilon)$-VaR of unserved customers:

$$\textbf{MP-CVaR}: \quad \min_{(w,x)\in\widetilde{X},v\geq 0,q^1,\ldots,q^{|K|}} \left\{ \sum_{k\in K} p^k q^k + G_0 v : L^k_{\text{CVaR}}(q^k,w,x,v) \geq 0 \quad \forall k \in K \right\}$$

where and $L^k_{\text{CVaR}}(q^k,w,x,v) \geq 0$ is the set of cuts generated by **SP-CVaR**$_k(w,x,v)$, the $k^{\text{th}}$ subproblem of the decomposed CVaR model, defined below.

We formulate each subproblem $k$ as the dual of $Q_c(w,x)$, separated by $k \in K$:

$$\begin{aligned}
\textbf{SP-CVaR}_k(w,x,v): \max_{\eta,\kappa,\theta} \quad & \sum_{i\in I} x_i(\eta_{i0} - \eta_{iT}) + \sum_{a\in A^{\text{one}}\cup A^{\text{two}}} (u^k_a \kappa_a + (u^k_a - v)\theta) \\
& + \sum_{i\in I} w_i \left( \sum_{a=(n_{it},n_{i,t+1})\in A^{\text{idle}}} \kappa_a \right) \\
\text{s.t.} \quad & \eta_{it} - \eta_{j,t+\ell_{ij}} + \kappa_a + \theta \leq f_a \qquad \forall a = (n_{it}, n_{j,t+\ell_{ij}}) \in A^{\text{one}} \\
& \eta_{it} - \eta_{is} + \kappa_a + \theta \leq f_a \qquad \forall a = (n_{it}, n_{is}) \in A^{\text{two}} \\
& \eta_{it} - \eta_{j,t+\ell_{ij}} \leq f_a \qquad \forall a = (n_{it}, n_{j,t+\ell_{ij}}) \in A^{\text{rel}} \\
& \eta_{it} - \eta_{i,t+1} + \kappa_a \leq f_a \qquad \forall a = (n_{it}, n_{i,t+1}) \in A^{\text{idle}} \\
& \kappa_a \leq 0 \qquad \forall a \in A^{\text{one}} \cup A^{\text{two}} \cup A^{\text{idle}} \\
& 0 \leq \theta \leq \frac{G_0}{\epsilon},
\end{aligned}$$

where $\eta_{it}$, $\kappa_a$ and $\theta$ are the dual variables associated with the flow balance constraints (4.3), the capacity constraints (4.4)–(4.5), and the CVaR constraint (4.11), respectively.

At each iteration of the Benders decomposition algorithm, we optimize a relaxed master problem MP-Stoch or MP-CVaR, to obtain optimal solutions $(\hat{x},\hat{w},\hat{q}^k)$ or $(\hat{x},\hat{w},\hat{q}^k,\hat{v})$. We pass the solutions to the respective subproblems, and optimize SP-Stoch$_k(\hat{w},\hat{x})$ or SP-CVaR$_k(\hat{w},\hat{x},\hat{v})$. If the optimal objective value of the subproblem corresponding to scenario $k$ is greater than the optimal value of $q^k$ given by the master problem, an optimality cut is generated to the master problem to remove this solution. For an optimal dual solution $(\widehat{\pi},\widehat{\lambda})$ to SP-Stoch$_k(\hat{w},\hat{x})$ for scenario $k$ in the risk-neutral model, the Benders optimality cut is

of the form

$$q^k - \sum_{i \in I} \left( \sum_{a=(n_{it}, n_{i,t+1}) \in A^{\text{idle}}} \widehat{\lambda}_a \right) w_i - \sum_{i \in I} (\widehat{\pi}_{i0} - \widehat{\pi}_{iT}) x_i - \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} \widehat{\lambda}_a u_a^k \geq 0. \quad (4.12)$$

For an optimal dual solution $(\widehat{\eta}, \widehat{\kappa}, \widehat{\theta})$ to SP-CVaR$_k(\hat{w}, \hat{x}, \hat{v})$ for scenario $k$ in the risk-averse model, the Benders cut is of the form

$$q^k - \sum_{i \in I} \left( \sum_{a=(n_{it}, n_{i,t+1}) \in A^{\text{idle}}} \widehat{\kappa}_a \right) w_i - \sum_{i \in I} (\widehat{\eta}_{i0} - \widehat{\eta}_{iT}) x_i + \widehat{\theta} v - \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} (\widehat{\kappa}_a + \widehat{\theta}) u_a^k \geq 0. \quad (4.13)$$

Feasibility cuts will not be required in either model, because the subproblems are always feasible when $x$ and $w$ are feasible for the relaxed master problem. Indeed, for any feasible $x$ and $w$, having the vehicles idle until the last period (i.e. $y_a = x_i$ for all arcs $a = (n_{it}, n_{i,t+1}) \in A^{\text{idle}}$ and $y_a = 0$ for all other arcs $a \in A^{\text{one}} \cup A^{\text{two}} \cup A^{\text{rel}}$) is always a feasible solution to the primal problem.

### 4.3.2 MIR Procedure

Mixed-integer rounding (MIR) is a procedure used to remove non-integer extreme point solutions from the linear relaxation of a mixed-integer program. We present a generic form of the MIR inequality with a non-negative real variable and multiple non-negative integer variables.

**Proposition 4.1.** (Wolsey, 1998) Let $U := \{(x, y) \in \mathbb{R}_+ \times \mathbb{Z}_+^m : x + \sum_{i=1}^m \alpha_i y_i - \delta \geq 0\}$ and let $\Delta > 0$. If $\textbf{frac}\,(\Delta\delta) > 0$, then the cut

$$x + \sum_{i=1}^{m} \frac{\min\left\{\lceil \Delta\alpha_i \rceil \textbf{frac}\,(\Delta\delta), \textbf{frac}\,(\Delta\alpha_i) + \lfloor \Delta\alpha_i \rfloor \textbf{frac}\,(\Delta\delta)\right\}}{\Delta} y_i - \frac{\lceil \Delta\delta \rceil \textbf{frac}\,(\Delta\delta)}{\Delta} \geq 0$$

is valid for $U$. □

The function $\textbf{frac}\,(b)$ is defined as $\textbf{frac}\,(b) := b - \lfloor b \rfloor$, the fractional part of a scalar $b$. Note that in Proposition 4.1, the variables $x$ and $y$ are generically defined and are unrelated to the $x$ and $y$ given in our formulations. Bodur and Luedtke (2014) extend Proposition 4.1 for a set defined by two inequalities. Following their idea, consequently, one can generate a different valid cut from two valid Benders cuts. We describe this result below in Theorem

4.4 with nomenclature relevant to our risk-neutral model and its corresponding Benders cuts.

**Theorem 4.4.** Let

$$q^k - \sum_{i\in I}\left(\sum_{a=(n_{it},n_{i,t+1})\in A^{\text{idle}}} \widehat{\lambda}_a^j\right) w_i - \sum_{i\in I}\widehat{\pi}_{i0}^j x_i - \sum_{a\in A^{\text{one}}\cup A^{\text{two}}} \widehat{\lambda}_a^j u_a^k \geq 0,\ j=1,2$$

be any pair of Benders cuts valid for the set of cuts $L^k_{\text{Stoch}}(q^k,w,x)\geq 0$ and let $\Delta>0$. Define

$$\begin{aligned}
\alpha_i &:= -\sum_{a=(n_{it},n_{i,t+1})\in A^{\text{idle}}}\left(\widehat{\lambda}_a^2-\widehat{\lambda}_a^1\right),\\
\beta_i &:= -\left(\widehat{\pi}_{i0}^2-\widehat{\pi}_{i0}^1\right),\\
\delta &:= \sum_{a\in A^{\text{one}}\cup A^{\text{two}}}\left(\widehat{\lambda}_a^2 u_a^k-\widehat{\lambda}_a^1 u_a^k\right).
\end{aligned}$$

If $\textbf{frac}\,(\Delta\delta)>0$, then the cut

$$\begin{aligned}
&q^k+\sum_{i\in I}\left(\frac{\min\left\{\lceil\Delta\alpha_i\rceil\,\textbf{frac}\,(\Delta\delta),\textbf{frac}\,(\Delta\alpha_i)+\lfloor\Delta\alpha_i\rfloor\,\textbf{frac}\,(\Delta\delta)\right\}}{\Delta}-\sum_{a=(n_{it},n_{i,t+1})\in A^{\text{idle}}}\widehat{\lambda}_a^1\right)w_i\\
&\quad+\sum_{i\in I}\left(\frac{\min\left\{\lceil\Delta\beta_i\rceil\,\textbf{frac}\,(\Delta\delta),\textbf{frac}\,(\Delta\beta_i)+\lfloor\Delta\beta_i\rfloor\,\textbf{frac}\,(\Delta\delta)\right\}}{\Delta}-\widehat{\pi}_{i0}^1\right)x_i\\
&\quad-\left(\frac{\lceil\Delta\delta\rceil\,\textbf{frac}\,(\Delta\delta)}{\Delta}+\sum_{a\in A^{\text{one}}\cup A^{\text{two}}}\widehat{\lambda}_a^1 u_a^k\right)\geq 0
\end{aligned}\tag{4.14}$$

is valid for $L^k_{\text{Stoch}}(q^k,w,x)\geq 0$. □

*Proof.* Let $(q^k,w,x)$ be an arbitrary solution satisfying the set of cuts $L^k_{\text{Stoch}}(q^k,w,x)\geq 0$ and define

$$q' := q^k - \sum_{i\in I}\left(\sum_{a=(n_{it},n_{i,t+1})\in A^{\text{idle}}}\widehat{\lambda}_a^1\right)w_i - \sum_{i\in I}\widehat{\pi}_{i0}^1 x_i - \sum_{a\in A^{\text{one}}\cup A^{\text{two}}}\widehat{\lambda}_a^1 u_a^k.$$

Then $q'\geq 0$ and

$$q'\geq q' - \left[q^k - \sum_{i\in I}\left(\sum_{a=(n_{it},n_{i,t+1})\in A^{\text{idle}}}\widehat{\lambda}_a^2\right)w_i - \sum_{i\in I}\widehat{\pi}_{i0}^2 x_i - \sum_{a\in A^{\text{one}}\cup A^{\text{two}}}\widehat{\lambda}_a^2 u_a^k\right]$$

$$= \delta - \sum_{i \in I} \alpha_i w_i - \sum_{i \in I} \beta_i x_i.$$

That is,

$$(q', w, x) \in U := \left\{ (q', w, x) \in \mathbb{R}_+ \times \mathbb{Z}_+^{|I|} \times \mathbb{Z}_+^{|I|} : q' + \sum_{i \in I} \alpha_i w_i + \sum_{i \in I} \beta_i x_i - \delta \geq 0 \right\}.$$

By applying Proposition 4.1, we obtain the cut

$$\begin{aligned} & q' + \sum_{i \in I} \frac{\min\left\{\lceil \Delta\alpha_i \rceil \, \mathbf{frac}\left(\Delta\delta\right), \mathbf{frac}\left(\Delta\alpha_i\right) + \lfloor \Delta\alpha_i \rfloor \, \mathbf{frac}\left(\Delta\delta\right)\right\}}{\Delta} w_i \\ & + \sum_{i \in I} \frac{\min\left\{\lceil \Delta\beta_i \rceil \, \mathbf{frac}\left(\Delta\delta\right), \mathbf{frac}\left(\Delta\beta_i\right) + \lfloor \Delta\beta_i \rfloor \, \mathbf{frac}\left(\Delta\delta\right)\right\}}{\Delta} x_i \\ & - \frac{\lceil \Delta\delta \rceil \, \mathbf{frac}\left(\Delta\delta\right)}{\Delta} \geq 0 \end{aligned} \tag{4.15}$$

valid for $U$. Finally, we substitute the expression $q'$ to obtain (4.14), valid for $L^k_{\text{Stoch}}(q^k, w, x) \geq 0$. □ □

We can directly apply Theorem 4.4 to pairs of Benders cuts (4.12) generated by **SP-Stoch**$_k(w, x)$ to generate new valid cuts. We call the process of generating a cut obtained via MIR on a pair of Benders cuts the *MIR procedure*. Similar procedures are employed in Bodur and Luedtke (2014) for optimizing a two-stage call center staffing problem under uncertain number of incoming calls.

On the other hand, cuts generated by **SP-CVaR**$_k(w, x, v)$ do not follow the form given in Theorem 4.4. We develop the following theorem to generate MIR cuts from pairs of Benders cuts (4.13) for the risk-averse CVaR-based model.

**Theorem 4.5.** Let

$$q^k - \sum_{i \in I} \left( \sum_{a=(n_{it}, n_{i,t+1}) \in A^{\text{idle}}} \widehat{\kappa}^j_a \right) w_i - \sum_{i \in I} \widehat{\eta}^j_{i0} x_i + \widehat{\theta}^j v - \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} (\widehat{\kappa}^j_a + \widehat{\theta}^j) u^k_a \geq 0, \; j = 1, 2 \tag{4.16}$$

be any pair of Benders cuts valid for the set of cuts $L^k_{\text{CVaR}}(q^k, w, x, v) \geq 0$ and let $\Delta > 0$. Define

$$\begin{aligned} \alpha_i &:= - \sum_{a=(n_{it}, n_{i,t+1}) \in A^{\text{idle}}} \left( \widehat{\kappa}^2_a - \widehat{\kappa}^1_a \right), \\ \beta_i &:= - \left( \widehat{\eta}^2_{i0} - \widehat{\eta}^1_{i0} \right), \end{aligned}$$

$$\delta := \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} \left( \left( \widehat{\kappa}_a^2 + \widehat{\theta}^2 \right) u_a^k - \left( \widehat{\kappa}_a^1 + \widehat{\theta}^1 \right) u_a^k \right).$$

If $\mathbf{frac}\,(\Delta\delta) > 0$ and $\widehat{\theta}^2 \geq \widehat{\theta}^1$, then the cut

$$\begin{aligned} q^k + \theta^2 v + \sum_{i \in I} \left( \frac{\min\left\{ \lceil \Delta\alpha_i \rceil \,\mathbf{frac}\,(\Delta\delta), \mathbf{frac}\,(\Delta\alpha_i) + \lfloor \Delta\alpha_i \rfloor \,\mathbf{frac}\,(\Delta\delta) \right\}}{\Delta} - \sum_{a=(n_{it}, n_{i,t+1}) \in A^{\text{idle}}} \widehat{\kappa}_a^1 \right) w_i \\ + \sum_{i \in I} \left( \frac{\min\left\{ \lceil \Delta\beta_i \rceil \,\mathbf{frac}\,(\Delta\delta), \mathbf{frac}\,(\Delta\beta_i) + \lfloor \Delta\beta_i \rfloor \,\mathbf{frac}\,(\Delta\delta) \right\}}{\Delta} - \widehat{\eta}_{i0}^1 \right) x_i \\ - \left( \frac{\lceil \Delta\delta \rceil \,\mathbf{frac}\,(\Delta\delta)}{\Delta} + \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} \left( \widehat{\kappa}_a^1 + \widehat{\theta}^1 \right) u_a^k \right) \geq 0 \end{aligned} \tag{4.17}$$

is valid for $L^k_{\text{CVaR}}(q^k, w, x, v) \geq 0$. □

*Proof.* Let $(q^k, w, x, v)$ be an arbitrary solution satisfying the set of cuts $L^k_{\text{CVaR}}(q^k, w, x, v) \geq 0$ and define

$$q' := q^k - \sum_{i \in I} \left( \sum_{a=(n_{it}, n_{i,t+1}) \in A^{\text{idle}}} \widehat{\kappa}_a^1 \right) w_i - \sum_{i \in I} \widehat{\eta}_{i0}^1 x_i + \widehat{\theta}^2 v - \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} \left( \widehat{\kappa}_a^1 + \widehat{\theta}^1 \right) u_a^k.$$

Then

$$\begin{aligned} q' &= \left[ q^k - \sum_{i \in I} \left( \sum_{a=(n_{it}, n_{i,t+1}) \in A^{\text{idle}}} \widehat{\kappa}_a^1 \right) w_i - \sum_{i \in I} \widehat{\eta}_{i0}^1 x_i + \widehat{\theta}^1 v - \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} \left( \widehat{\kappa}_a^1 + \widehat{\theta}^1 \right) u_a^k \right] + \left( \widehat{\theta}^2 - \widehat{\theta}^1 \right) v \\ &\geq 0 \end{aligned}$$

and

$$\begin{aligned} q' &\geq q' - \left[ q^k - \sum_{i \in I} \left( \sum_{a=(n_{it}, n_{i,t+1}) \in A^{\text{idle}}} \widehat{\kappa}_a^2 \right) w_i - \sum_{i \in I} \widehat{\eta}_{i0}^2 x_i + \widehat{\theta}^2 v - \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} \left( \widehat{\kappa}_a^2 + \widehat{\theta}^2 \right) u_a^k \right] \\ &= \delta - \sum_{i \in I} \alpha_i w_i - \sum_{i \in I} \beta_i x_i. \end{aligned}$$

That is,

$$(q', w, x) \in U := \left\{ (q', w, x) \in \mathbb{R}_+ \times \mathbb{Z}_+^{|I|} \times \mathbb{Z}_+^{|I|} : q' + \sum_{i \in I} \alpha_i w_i + \sum_{i \in I} \beta_i x_i - \delta \geq 0 \right\}.$$

By applying Proposition 4.1, we obtain the cut (4.15) valid for $U$. Finally, we substitute the expression $q'$ to obtain (4.17), which is valid for $L^k_{\text{CVaR}}(q^k, w, x, v) \geq 0$. □ □

Theorem 4.5 is similar to Theorem 4.4, but further requires that $\widehat{\theta}^2 - \widehat{\theta}^1 \geq 0$. However,

this condition can be satisfied with any pair of Benders cuts (4.13), as the indices 1 and 2 used in the theorem can be swapped without loss of generality. We include in Appendix C the implementation of the branch-and-cut algorithm with MIR-enhanced Benders cuts, generalized for both risk-neutral and risk-averse models.

## 4.4 Computational Results

We apply the two-stage stochastic integer programming formulation to optimize the allocation of fleet and parking space in carsharing systems with real-world data. In particular, we focus on the effects of different penetrations of one-way rentals on both reserved-parking and free-floating carsharing systems. Section 4.4.2 analyzes how the ratio of one-way to round-trip demand affects profitability and QoS. In Section 4.4.3, we vary parameter settings in the risk-neutral and risk-averse models, and analyze the effects of lowering relocation costs on the metrics of profitability, QoS, and denied rentals. Section 4.4.4 demonstrates the efficacy of decomposition algorithms and the MIR procedure for enhancing the Benders cuts.

### 4.4.1 Design of Experiments

#### 4.4.1.1 Data Generation

We use Zipcar rental data collected from the Boston-Cambridge area in Massachusetts in our computations. There are a total of 62 days of data, taken from Oct 1 to Dec 1, 2014, which contain the information of the starting and ending times of the rentals and the zipcodes of the origin and destination zones of the rentals. Figure 4.2 illustrates how we divide the Boston-Cambridge area into nine zones.

First, according to the demand's traveling pattern, we divide the Boston-Cambridge area into nine zones. We pre-process the rental data as follows. The rentals are labeled as one-way or round-trip rentals, depending on whether they had different or the same starting and ending zones, respectively. One-way rentals are further categorized by their origin and destination zones and the hour in which the rental started. Round-trip rentals are categorized by their origin (which is also their destination) zone and the hours in which the rental started and ended. One-way rentals are aggregated by their *(origin, destination, starting hour)* triplet, while round-trip rentals are aggregated by their *(origin, starting hour, ending hour)* triplet, for each day, resulting in 62 data points for each of the aforementioned triplets.


Figure 4.2: Division of Boston-Cambridge area into nine zones

We use the Monte Carlo sampling to generate random instances by sampling the number of rentals for each triplet by following Gamma distributions with means and variances equal to the empirical values obtained from the Zipcar data. We conduct hypothesis test of the Gamma distribution because hourly demand for each triplet is non-negative and generally seen to have higher frequencies of small values. We compute the scale and shape parameters of the assumed Gamma distribution for each triplet from the means and variances obtained from each set of 62 data points. Figure 4.3 compares the probability densities of the observed data and the assumed Gamma distribution (computed from the mean and variance of the observations) for two representative types of triplets, which have higher frequencies of low demand and of high demand values, respectively. The assumed Gamma distribution follows the observed distribution closely in both cases.

In our experiments, we target a mean total demand of 1000 rentals. As the experiments require that we use different proportions of one-way versus round-trip rentals, the means and variances are scaled accordingly.

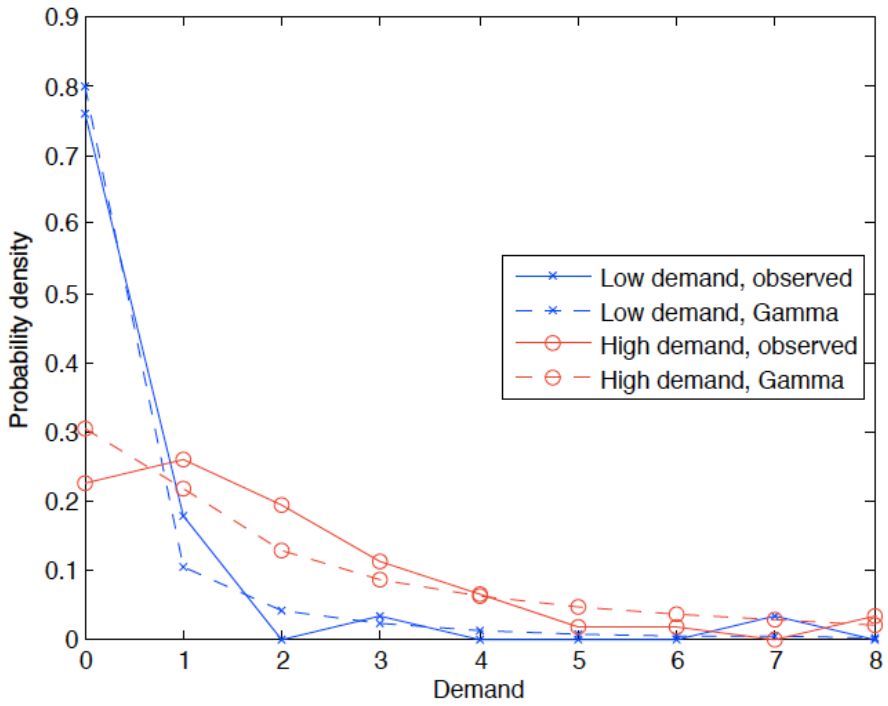



Figure 4.3: Comparison of observed distributions and assumed Gamma distributions

#### 4.4.1.2 Computational Setup

For each mix of one-way and round-trip demand, we use a training sample of 100 scenarios to compute the optimal first-stage decisions $w$ and $x$. We use three models to compute the first-stage solutions $w$ and $x$ for both systems, namely the risk-neutral model (which we refer to as `Stoch`), the CVaR model with $\epsilon = 0.1$ (`CVaR-0.1`), and the CVaR model with $\epsilon = 0.05$ (`CVaR-0.05`). The performance of the first-stage solutions is evaluated using a test sample of 1000 scenarios with the same mix of one-way and round-trip rentals. The model used in this evaluation is (4.6)–(4.7) with the penalty term $g(y^1, \ldots, y^{|K|}) = 0$.

In our numerical experiments, we vary one-way proportion from 0% to 100%, in 20% increments. The granularity of time used is one hour per period, and the experiments are run with $T = 24$ periods. As we do not have data on the fleet size of Zipcar in Boston, we run trial experiments with fleet sizes of 100, 200 and 300 to observe their effect on the results. In the trial experiments, a fleet size of 300 results in almost all demand being satisfied regardless of the other parameters, and consequently perfect or almost-perfect QoS for all experiments. On the other hand, a fleet size of 100 results in trends similar to a fleet size of 200, albeit with smaller absolute values. Consequently, we set $S = 200$ to observe more significant changes when varying other parameters, yet with the fleet not too large as to result in consistently perfect QoS.

For the reserved-parking-based system, we set the parking lot cost $c_i^{\text{lot}} = \$9.6$ per hour in zones $i = 1, 2, 5, 6, 9$, $c_i^{\text{lot}} = \$7.4$ per hour in other zones, and the free-float permit cost $c^{\text{ffp}} = \$0$. For the free-floating system, free-float permit cost is set as $c^{\text{ffp}} = \$9.6$ per hour, and $c_i^{\text{lot}} = \$0$ for all zones $i$. Following the studies by Chesto (2015), these costs are based on annual parking lot reservation costs of \$3,500 per lot in Boston downtown and \$2,700 outside of downtown, and annual free-float permit costs of \$3,500 per car, divided by 365 days.

We use revenue parameters $r^{\text{two}} = \$7.75$ per hour and $r^{\text{one}} = \$12$ per hour based on Zipcar's Boston rental rates for round-trip and one-way in its ONE>WAY program (see Zipcar, 2015). Relocation cost is $c^{\text{rel}} = \$22$ per hour, based on the average annual wage of \$46,481.52 in the U.S. in 2014 (see Social Security Administration, 2015), divided by 52 weeks and 40 hours. Maintenance $c^{\text{mnt}}$ and idling costs $c^{\text{idle}}$ are assumed to be negligible and set to zero. Finally, $G_a = -10 f_a$ for all arcs $a \in A^{\text{one}} \cup A^{\text{two}}$ in `Stoch` and $G_0 = \frac{1}{|A^{\text{one}} \cup A^{\text{two}}|} \sum_{a \in A^{\text{one}} \cup A^{\text{two}}} G_a$ in both `CVaR-0.1` and `CVaR-0.05`. That is, the penalty on each arc is ten times the per unit net revenue of flow and the CVaR penalty factor is the average penalty over all rental arcs.

We implement all the algorithms in Java using Gurobi 6.0.3. The results below are obtained by using a Dell Alienware X51, with an Intel Core i7-3770 dual-core CPU @ 3.4GHz each and 8GB RAM.

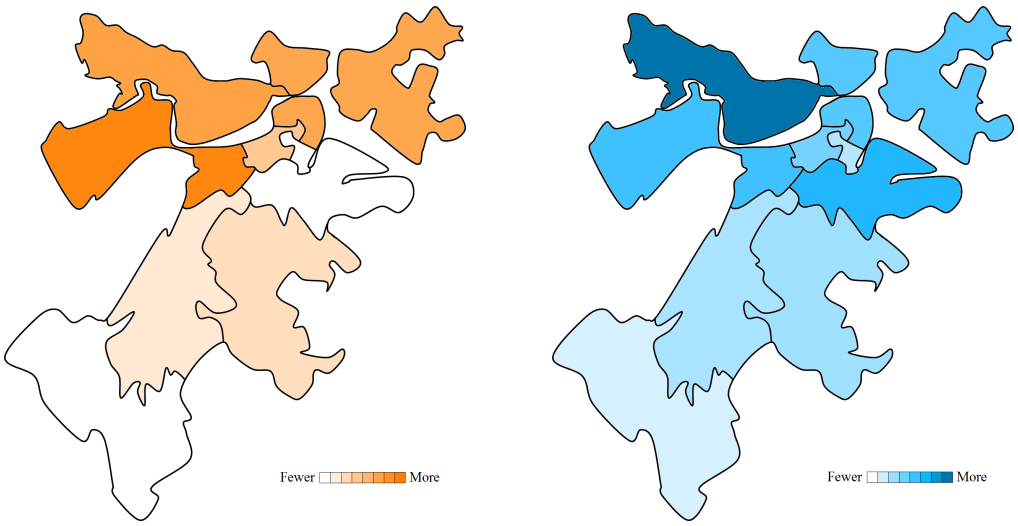
### 4.4.2 Effect of Demand Type Mix

A primary factor affecting the profit and QoS levels of a carsharing service is the mix of one-way and round-trip demand. Volatile pricing of rentals often negatively impacts customer experience, but customer demand, and consequently the ratio of one-way to round-trip rentals, is volatile and changes from day to day. Hence, it is important to understand the impact that this ratio has on a carsharing service. In our first set of experimental results, we evaluate the effects of the mix of one-way and round-trip rentals on carsharing systems with reserved parking lots or free-float parking permits.

#### 4.4.2.1 Demand Concentration versus Vehicle Allocation

First, we compare the average total number of rentals demanded in each zone against the first-stage vehicle allocation obtained through the models. Figure 4.4 shows the relative concentration of demand starting in each zone and the relative allocation of vehicles obtained by solving `Stoch` when the proportion of one-way rentals is 40%. A deeper shade represents a higher demand concentration in Figure 4.4(a) and a larger number of vehicles

allocated in Figure 4.4(b).




(a) Demand concentration (b) Vehicle allocation

Figure 4.4: Visual comparison of demand concentration (by starting zone) versus vehicle allocation

The data indicates that more demand starts from the zones to the north of the Boston-Cambridge area than from the southern zones. Meanwhile, we observe more vehicles allocated to the northern zones. Also, the allocation of vehicles does not match the demand concentrations directly, as the optimal allocation of vehicles also depends on the ending zones of the one-way demand. This validates the realism of our results, and the non-triviality of the problem of allocating a carsharing fleet to initial zones.

#### 4.4.2.2 Profitability

The profitability of the first-stage solutions comprises the following main components: the setup cost of purchasing parking lots or free-float permits, the expected revenue from one-way and round-trip rentals, and the expected cost of relocating cars. Recall that maintenance and idling costs are zero in our experimental setup. Table 4.2 shows these components for each proportion of one-way rentals and each system under the three models used. The last column presents the expected total profit.

In Table 4.2, across the three models and the two systems, the profitability decreases as the proportion of one-way rentals increases. The main contributing factor for this decrease is the decrease in overall revenue. When all rentals are one-way, the revenue gained is only half of that when all rentals are round-trip. This is because one-way rentals are generally shorter in duration (as mentioned in Section 4.4.1.1, they are assumed to be one hour long)

Table 4.2: Profitability metrics of solutions

| One-way proportion | Carsharing system | Model | Setup cost | Revenue from one-way | Revenue from round-trip | Relocation costs | Profit |
|---|---|---|---|---|---|---|---|
| 0% | Reserved Parking | Stoch | 1,887 | 0 | 16,862 | 7 | 14,967 |
| | | CVaR-0.1 | 1,887 | 0 | 16,861 | 6 | 14,968 |
| | | CVaR-0.05 | 1,887 | 0 | 16,861 | 6 | 14,968 |
| | Free Float | Stoch | 1,920 | 0 | 16,864 | 7 | 14,936 |
| | | CVaR-0.1 | 1,920 | 0 | 16,864 | 7 | 14,937 |
| | | CVaR-0.05 | 1,920 | 0 | 16,864 | 7 | 14,937 |
| 20% | Reserved Parking | Stoch | 1,861 | 1,187 | 14,386 | 108 | 13,604 |
| | | CVaR-0.1 | 1,869 | 1,169 | 14,423 | 100 | 13,622 |
| | | CVaR-0.05 | 1,869 | 1,167 | 14,423 | 99 | 13,622 |
| | Free Float | Stoch | 1,920 | 1,177 | 14,399 | 101 | 13,556 |
| | | CVaR-0.1 | 1,920 | 1,166 | 14,425 | 97 | 13,574 |
| | | CVaR-0.05 | 1,920 | 1,164 | 14,430 | 98 | 13,577 |
| 40% | Reserved Parking | Stoch | 1,822 | 3,026 | 11,213 | 246 | 12,172 |
| | | CVaR-0.1 | 1,836 | 3,013 | 11,277 | 213 | 12,239 |
| | | CVaR-0.05 | 1,841 | 3,021 | 11,294 | 222 | 12,252 |
| | Free Float | Stoch | 1,920 | 3,008 | 11,205 | 205 | 12,088 |
| | | CVaR-0.1 | 1,920 | 3,016 | 11,274 | 194 | 12,177 |
| | | CVaR-0.05 | 1,920 | 3,021 | 11,291 | 199 | 12,193 |
| 60% | Reserved Parking | Stoch | 1,770 | 4,804 | 7,475 | 333 | 10,176 |
| | | CVaR-0.1 | 1,801 | 4,841 | 7,505 | 343 | 10,203 |
| | | CVaR-0.05 | 1,805 | 4,810 | 7,511 | 316 | 10,200 |
| | Free Float | Stoch | 1,920 | 4,765 | 7,485 | 278 | 10,053 |
| | | CVaR-0.1 | 1,920 | 4,800 | 7,514 | 287 | 10,107 |
| | | CVaR-0.05 | 1,920 | 4,801 | 7,518 | 286 | 10,114 |
| 80% | Reserved Parking | Stoch | 1,682 | 6,625 | 3,572 | 454 | 8,061 |
| | | CVaR-0.1 | 1,725 | 6,583 | 3,574 | 417 | 8,015 |
| | | CVaR-0.05 | 1,725 | 6,573 | 3,575 | 409 | 8,014 |
| | Free Float | Stoch | 1,872 | 6,601 | 3,574 | 411 | 7,891 |
| | | CVaR-0.1 | 1,920 | 6,570 | 3,578 | 383 | 7,845 |
| | | CVaR-0.05 | 1,920 | 6,564 | 3,578 | 378 | 7,844 |
| 100% | Reserved Parking | Stoch | 1,496 | 8,359 | 0 | 549 | 6,315 |
| | | CVaR-0.1 | 1,535 | 8,297 | 0 | 495 | 6,267 |
| | | CVaR-0.05 | 1,537 | 8,357 | 0 | 544 | 6,276 |
| | Free Float | Stoch | 1,670 | 8,299 | 0 | 462 | 6,166 |
| | | CVaR-0.1 | 1,718 | 8,302 | 0 | 463 | 6,120 |
| | | CVaR-0.05 | 1,718 | 8,308 | 0 | 469 | 6,121 |

as compared to round-trip rentals, which could be several hours long. The cost differential between one-way and round-trip rentals appears to be ineffective in bridging the revenue gap between the two. The additional burden of relocation costs for one-way rentals further suppresses profits. Even though setup cost decreases as one-way proportion increases, this decrease is not sufficient to offset the loss in revenue and increase in relocation cost. The results explain why careshare companies have less incentive to popularize one-way rentals, although they could attract more customers for the more flexible rental form.

The three models show minor differences in their profitability for both systems. The CVaR models have slightly higher setup costs but lower relocation costs compared to the risk-neutral model. Revenue from round-trip rentals is also higher for the CVaR models. Revenue from one-way rentals does not appear to have a direct relationship with the type of model used. This can be attributed to the risk-averse CVaR model being more likely to avoid a higher second-stage cost by incurring more costs in the first stage. Overall, the profitability of solutions from using a risk-neutral model is lower than those obtained from using a CVaR model for smaller proportions of one-way rental demand (up to 60%), with the difference in profitability being greatest at 40% demand being one-way. This relationship is reversed for higher proportions of one-way demand of 80% and 100%.

When comparing the two types of carsharing systems, we see that reserved-parking-based systems had consistently higher profit than free-floating systems. This may be due to the similar cost per parking lot and cost per free-floating permit in downtown zones and lower parking lot costs than permit costs in other zones. Figure 4.5 shows the observed probability density functions of the profits, using a risk-neutral model when the proportion of one-way rentals is 80%. Both probability densities show some positive skewness, likely due to the underlying Gamma distribution of demand. The profit of a reserved-parking-based system has lower density for values below 7,250 and higher density for values above 7,250 when compared to the profit of a free-floating system.

#### 4.4.2.3 QoS Performance

We evaluate the QoS of the solutions given by the models using the expected number of demanded rentals unfulfilled, the expected proportion of unfulfilled rentals out of the total demand, and the expected number of unfulfilled vehicle hours. All three metrics are given in Table 4.3, together with the expected total number of vehicle-hours spent idling.

Under both systems and using any of the three models, the expected number of unfulfilled rentals increases as the proportion of one-way rentals increases, as does the expected proportion of unfulfilled rentals or the expected number of unfulfilled vehicle hours. This further supports our earlier claim that the price differential between one-way and round-

Table 4.3: QoS metrics of solutions

| One-way proportion | Carsharing system | Model | Unfulfilled rentals | | | Idle vehicle-hours |
|---|---|---|---|---|---|---|
| | | | Number | As proportion | In vehicle-hours | |
| 0% | Reserved Parking | Stoch | 146.2 | 14.6% | 290.4 | 2,423.3 |
| | | CVaR-0.1 | 145.9 | 14.6% | 290.4 | 2,423.4 |
| | | CVaR-0.05 | 145.9 | 14.6% | 290.4 | 2,423.4 |
| | Free Float | Stoch | 147.1 | 14.7% | 290.7 | 2,423.5 |
| | | CVaR-0.1 | 147.0 | 14.7% | 290.8 | 2,423.7 |
| | | CVaR-0.05 | 147.0 | 14.7% | 290.8 | 2,423.7 |
| 20% | Reserved Parking | Stoch | 123.9 | 13.0% | 166.4 | 2,633.8 |
| | | CVaR-0.1 | 123.7 | 12.9% | 164.0 | 2,631.8 |
| | | CVaR-0.05 | 123.3 | 12.9% | 163.5 | 2,631.2 |
| | Free Float | Stoch | 123.1 | 12.9% | 167.3 | 2,634.0 |
| | | CVaR-0.1 | 122.9 | 12.9% | 164.0 | 2,631.6 |
| | | CVaR-0.05 | 122.9 | 12.9% | 164.1 | 2,631.8 |
| 40% | Reserved Parking | Stoch | 135.8 | 14.1% | 144.4 | 2,883.1 |
| | | CVaR-0.1 | 129.9 | 13.5% | 134.7 | 2,874.5 |
| | | CVaR-0.05 | 127.9 | 13.3% | 132.2 | 2,871.5 |
| | Free Float | Stoch | 134.3 | 14.0% | 141.9 | 2,876.5 |
| | | CVaR-0.1 | 129.8 | 13.5% | 134.8 | 2,872.6 |
| | | CVaR-0.05 | 127.8 | 13.3% | 131.8 | 2,868.8 |
| 60% | Reserved Parking | Stoch | 177.7 | 18.5% | 178.6 | 3,209.3 |
| | | CVaR-0.1 | 171.8 | 17.9% | 172.0 | 3,201.7 |
| | | CVaR-0.05 | 171.2 | 17.8% | 171.3 | 3,201.2 |
| | Free Float | Stoch | 175.8 | 18.3% | 176.7 | 3,201.9 |
| | | CVaR-0.1 | 169.4 | 17.6% | 169.6 | 3,193.8 |
| | | CVaR-0.05 | 171.4 | 17.8% | 171.5 | 3,198.4 |
| 80% | Reserved Parking | Stoch | 224.4 | 23.4% | 224.5 | 3,432.7 |
| | | CVaR-0.1 | 226.5 | 23.6% | 226.5 | 3,552.6 |
| | | CVaR-0.05 | 226.9 | 23.7% | 226.9 | 3,553.4 |
| | Free Float | Stoch | 222.6 | 23.2% | 222.7 | 3,426.6 |
| | | CVaR-0.1 | 225.8 | 23.5% | 225.9 | 3,548.5 |
| | | CVaR-0.05 | 226.5 | 23.6% | 226.6 | 3,550.0 |
| 100% | Reserved Parking | Stoch | 283.7 | 29.1% | 283.7 | 3,264.2 |
| | | CVaR-0.1 | 283.4 | 29.1% | 283.4 | 3,378.8 |
| | | CVaR-0.05 | 282.9 | 29.0% | 282.9 | 3,377.7 |
| | Free Float | Stoch | 278.6 | 28.6% | 278.6 | 3,250.5 |
| | | CVaR-0.1 | 283.8 | 29.1% | 283.8 | 3,376.1 |
| | | CVaR-0.05 | 278.8 | 28.6% | 278.8 | 3,366.1 |

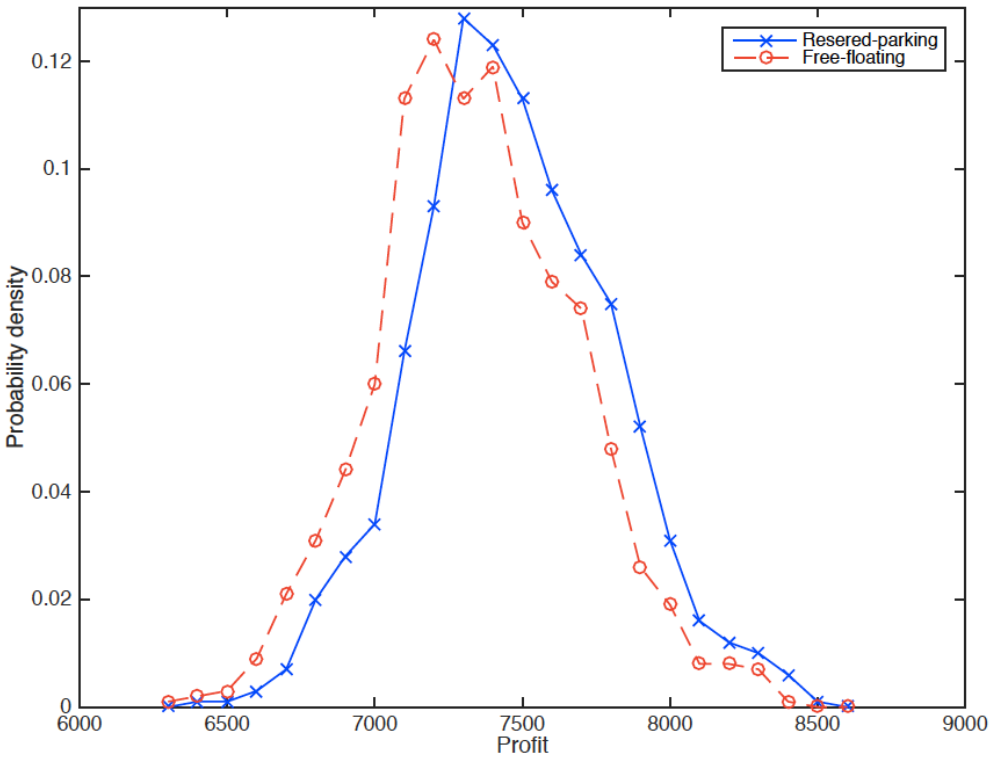



Figure 4.5: Observed probability density of profit

trip rentals is insufficient for the former to be as profitable as the latter. This can also be seen from the fact that the expected number of vehicle hours spent idling increases as the proportion of one-way rentals increases, indicating that with higher one-way proportions, the carsharing company would rather have vehicles idle than satisfying all demand.

However, our results do indicate that a small amount of one-way demand may be beneficial to the improvement of QoS, as the lowest proportion of unfulfilled rentals occurs when the proportion of demand for one-way rentals is 20%. This may be due to one-way rentals being able to provide a limited amount of desirable relocation of vehicles to respond to changes in demand patterns during the demand horizon, as opposed to a static configuration of the car fleet when all demand is round-trip.

There are very small differences between the number of unfulfilled rentals under a reserved-parking-based system and a free-floating system. The values never differ by more than 2.5 in our results, and most differ by no more than 1.5. When comparing between the three models, there are also minor differences, with the differences being more pronounced for moderate proportions of one-way rentals. There is no clear advantage, in terms of QoS, of using the CVaR model over the risk-neutral model. This may be surprising, given that the CVaR model is risk-averse. However, the CVaR model focuses on the worst $100\epsilon\%$

scenarios, and thus may not have better QoS on average.

For clarity, we include Figure 4.6 to illustrate how VaR and CVaR are computed. The figure shows the observed probability density functions of the proportions of rentals unfulfilled by the solutions given by `Stoch` and `CVaR-0.05` when the proportion of one-way rentals is 40%. The vertical thresholds indicate the 0.9-VaR (i.e. the VaR when $\epsilon = 0.1$) for each of the two curves. These thresholds are such that the area under the curve to the right of the threshold is 0.1. In the case of `Stoch`, this area is indicated by the blue and green shaded areas; in the case of `CVaR-0.05`, this area is indicated by the orange and greed shad :cted valu

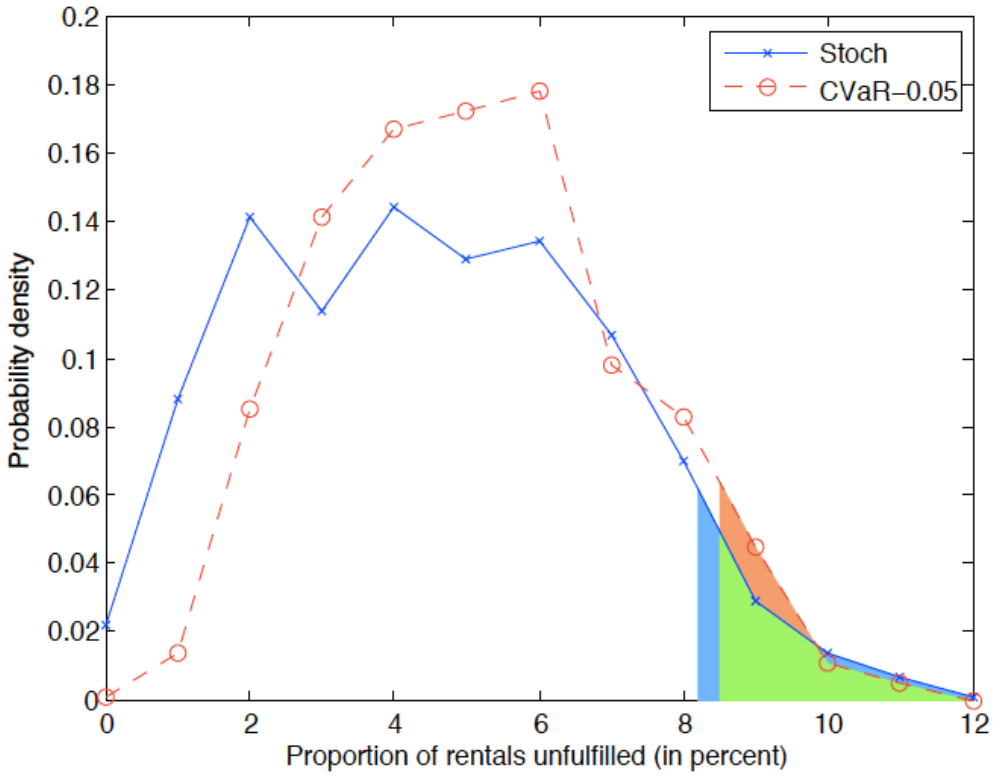



Figure 4.6: Observed distribution of proportion of demanded rentals unfulfilled

#### 4.4.2.4 Denied Rentals

A denied rental is one that is not fulfilled even though there is a car available for use in that zone at that time. Such a phenomenon is unlikely to happen in a free-floating system as the company cannot prevent customers from renting cars if they are available. Hence, denied rentals can be used to measure how realistically the model approximates a free-floating system. However, in a reserved-parking-based system, customers could potentially be blocked from using a car that is available, if doing so might result in greater overall profitability. Denied rentals, in this case, provide another measure of QoS.

Table 4.4 gives the expected proportion of rentals denied and several percentile values of this proportion. The proportion of rentals denied increases with the proportion of one-

Table 4.4: Distribution of proportion of rentals denied when $c^{\text{rel}} = 22$

| One-way proportion | Carsharing system | Model | Mean | Percentile | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | | 25% | 50% | 75% | 90% | 95% | 99% |
| 0% | Reserved Parking | Stoch | 0.24% | 0.10% | 0.20% | 0.33% | 0.52% | 0.64% | 0.91% |
| | | CVaR-0.1 | 0.25% | 0.10% | 0.20% | 0.37% | 0.53% | 0.70% | 1.01% |
| | | CVaR-0.05 | 0.25% | 0.10% | 0.20% | 0.37% | 0.53% | 0.70% | 1.01% |
| | Free Float | Stoch | 0.25% | 0.10% | 0.20% | 0.32% | 0.51% | 0.70% | 1.03% |
| | | CVaR-0.1 | 0.25% | 0.10% | 0.20% | 0.33% | 0.52% | 0.69% | 1.01% |
| | | CVaR-0.05 | 0.25% | 0.10% | 0.20% | 0.33% | 0.52% | 0.69% | 1.01% |
| 20% | Reserved Parking | Stoch | 3.52% | 2.94% | 3.48% | 4.05% | 4.62% | 4.96% | 5.58% |
| | | CVaR-0.1 | 3.20% | 2.66% | 3.17% | 3.67% | 4.17% | 4.48% | 5.10% |
| | | CVaR-0.05 | 3.25% | 2.74% | 3.22% | 3.77% | 4.22% | 4.61% | 5.13% |
| | Free Float | Stoch | 3.69% | 3.13% | 3.66% | 4.18% | 4.74% | 5.15% | 5.73% |
| | | CVaR-0.1 | 3.32% | 2.81% | 3.28% | 3.79% | 4.24% | 4.58% | 5.38% |
| | | CVaR-0.05 | 3.23% | 2.70% | 3.18% | 3.71% | 4.21% | 4.51% | 5.12% |
| 40% | Reserved Parking | Stoch | 7.67% | 6.73% | 7.61% | 8.63% | 9.48% | 9.99% | 10.80% |
| | | CVaR-0.1 | 7.41% | 6.46% | 7.44% | 8.30% | 9.17% | 9.74% | 10.55% |
| | | CVaR-0.05 | 6.87% | 5.88% | 6.90% | 7.73% | 8.57% | 9.06% | 10.01% |
| | Free Float | Stoch | 10.12% | 9.16% | 10.24% | 11.16% | 11.90% | 12.27% | 13.07% |
| | | CVaR-0.1 | 8.28% | 7.30% | 8.27% | 9.28% | 10.09% | 10.64% | 11.30% |
| | | CVaR-0.05 | 7.83% | 6.88% | 7.78% | 8.83% | 9.71% | 10.23% | 11.08% |
| 60% | Reserved Parking | Stoch | 11.64% | 10.43% | 11.65% | 12.88% | 13.89% | 14.42% | 15.47% |
| | | CVaR-0.1 | 10.98% | 9.74% | 10.99% | 12.23% | 13.39% | 14.05% | 14.89% |
| | | CVaR-0.05 | 11.78% | 10.42% | 11.81% | 13.10% | 14.16% | 14.96% | 16.56% |
| | Free Float | Stoch | 16.40% | 15.37% | 16.50% | 17.46% | 18.32% | 18.82% | 19.47% |
| | | CVaR-0.1 | 13.70% | 12.50% | 13.78% | 14.81% | 15.84% | 16.44% | 17.82% |
| | | CVaR-0.05 | 13.75% | 12.69% | 13.76% | 14.91% | 15.91% | 16.40% | 17.63% |
| 80% | Reserved Parking | Stoch | 15.30% | 13.78% | 15.34% | 16.99% | 18.19% | 19.06% | 20.40% |
| | | CVaR-0.1 | 16.58% | 15.09% | 16.61% | 18.15% | 19.46% | 20.36% | 21.39% |
| | | CVaR-0.05 | 17.10% | 15.61% | 17.16% | 18.60% | 19.96% | 20.81% | 22.25% |
| | Free Float | Stoch | 17.16% | 15.71% | 17.26% | 18.67% | 19.83% | 20.52% | 21.82% |
| | | CVaR-0.1 | 18.15% | 16.58% | 18.13% | 19.77% | 20.93% | 21.81% | 22.96% |
| | | CVaR-0.05 | 21.90% | 20.77% | 21.99% | 23.04% | 23.95% | 24.57% | 25.76% |
| 100% | Reserved Parking | Stoch | 19.71% | 18.05% | 19.72% | 21.42% | 22.93% | 23.75% | 25.03% |
| | | CVaR-0.1 | 21.52% | 19.73% | 21.57% | 23.40% | 24.95% | 25.91% | 27.62% |
| | | CVaR-0.05 | 19.95% | 18.08% | 20.08% | 21.77% | 23.49% | 24.33% | 25.98% |
| | Free Float | Stoch | 26.84% | 25.48% | 27.00% | 28.31% | 29.47% | 30.12% | 31.04% |
| | | CVaR-0.1 | 23.12% | 21.30% | 23.27% | 24.92% | 26.42% | 27.23% | 28.97% |
| | | CVaR-0.05 | 22.70% | 21.07% | 22.81% | 24.24% | 25.70% | 26.79% | 28.91% |

way demand. For low proportions of one-way demand, the expected proportion of denied rentals is quite low, at approximately 3% with 20% of demand being one-way and at 7%-10% with 40% of demand being one-way. However, at 60% and higher one-way demand, the proportion of rentals denied can be quite high, reaching up to 26% when all demand is one-way. This suggests that reserved-parking-based and free-floating carsharing systems will cope with high proportions of one-way demand in different ways – while the former

can continue using the model without much repercussion, the latter cannot as it diverges from reality for high one-way demand.

### 4.4.3 Effect of Lowering Relocation Cost

Our results indicate that under the given cost parameters, high demand for one-way rentals can be detrimental to both profitability and QoS, and also decreases the model's ability to realistically approximate customer behavior when used for free-floating carsharing systems. However, this can be remedied by a reduction in relocation costs relative to the revenue from shared car rentals. Firstly, this can be a potential subsidy scheme to encourage carsharing by increasing QoS. This is especially relevant for cities with high demand for one-way rentals relative to round-trip rentals. Secondly, the cost of relocation can be offset by higher pricing of rentals. This may not be desirable for customers, but is still a viable option should the carsharing company face high one-way rental demand. In the following results, we used the same parameters as in Section 4.4.1, but set a lower relocation cost of $c^{\text{rel}} = 10$. (The previous $c^{\text{rel}} = 22$ set according to the average annual wage in the U.S. in 2014.)

#### 4.4.3.1 Profitability

Table 4.5 describes the profitability metrics with lower relocation costs. Total relocation costs are higher, indicating much more relocation occurring. However, revenue from one-way rentals is significantly higher, and can offset the additional relocation costs. Revenue from round-trip rentals is unchanged. Overall, lower relocation cost results in slightly improved profitability for high demand of one-way rentals.

#### 4.4.3.2 QoS Performance

Table 4.6 describes the QoS metrics with lower relocation costs. There is a drastic improvement in the QoS for higher proportions of one-way rentals; when all rentals are one-way, there are almost no unfulfilled rentals. This indicates that the higher the proportion of one-way demand, the greater the impact lower relocation costs has on the QoS.

#### 4.4.3.3 Denied Rentals

Table 4.7 shows the proportion of denied rentals with lower relocation costs. There is a significant improvement in the number of denied rentals, indicating increased reliability of the solutions given by the model. The number of denied rentals appears to decrease as

Table 4.5: Profitability metrics of solutions with lower relocation costs ($c^{\text{rel}} = 10$)

| One-way proportion | Carsharing system | Model | Setup cost | Revenue from one-way | Revenue from round-trip | Relocation costs | Profit |
|---|---|---|---|---|---|---|---|
| 60% | Reserved Parking | Stoch | 1,749 | 6,761 | 7,491 | 1,986 | 10,517 |
| | | CVaR-0.1 | 1,767 | 6,710 | 7,503 | 1,919 | 10,528 |
| | | CVaR-0.05 | 1,765 | 6,711 | 7,503 | 1,921 | 10,527 |
| | Free Float | Stoch | 1,920 | 6,748 | 7,499 | 1,936 | 10,391 |
| | | CVaR-0.1 | 1,920 | 6,653 | 7,514 | 1,834 | 10,413 |
| | | CVaR-0.05 | 1,920 | 6,627 | 7,516 | 1,810 | 10,413 |
| 80% | Reserved Parking | Stoch | 1,635 | 9,282 | 3,570 | 2,696 | 8,521 |
| | | CVaR-0.1 | 1,679 | 9,284 | 3,573 | 2,694 | 8,484 |
| | | CVaR-0.05 | 1,679 | 9,284 | 3,573 | 2,694 | 8,484 |
| | Free Float | Stoch | 1,862 | 9,277 | 3,576 | 2,640 | 8,350 |
| | | CVaR-0.1 | 1,920 | 9,281 | 3,578 | 2,641 | 8,298 |
| | | CVaR-0.05 | 1,920 | 9,281 | 3,578 | 2,641 | 8,298 |
| 100% | Reserved Parking | Stoch | 1,418 | 11,700 | 0 | 3,389 | 6,894 |
| | | CVaR-0.1 | 1,447 | 11,702 | 0 | 3,391 | 6,864 |
| | | CVaR-0.05 | 1,447 | 11,702 | 0 | 3,391 | 6,864 |
| | Free Float | Stoch | 1,670 | 11,697 | 0 | 3,294 | 6,733 |
| | | CVaR-0.1 | 1,718 | 11,701 | 0 | 3,296 | 6,687 |
| | | CVaR-0.05 | 1,718 | 11,701 | 0 | 3,296 | 6,687 |

Table 4.6: QoS metrics of solutions with lower relocation costs ($c^{\text{rel}} = 10$)

| One-way proportion | Carsharing system | Model | Unfulfilled rentals | | | Idle vehicle-hours |
|---|---|---|---|---|---|---|
| | | | Number | As proportion | In vehicle-hours | |
| 60% | Reserved Parking | Stoch | 11.5 | 1.2% | 11.5 | 2,871.4 |
| | | CVaR-0.1 | 14.2 | 1.5% | 14.2 | 2,880.8 |
| | | CVaR-0.05 | 14.2 | 1.5% | 14.2 | 2,880.6 |
| | Free Float | Stoch | 11.6 | 1.2% | 11.6 | 2,876.5 |
| | | CVaR-0.1 | 17.6 | 1.8% | 17.6 | 2,892.6 |
| | | CVaR-0.05 | 19.4 | 2.0% | 19.4 | 2,896.9 |
| 80% | Reserved Parking | Stoch | 1.5 | 0.2% | 1.5 | 2,981.3 |
| | | CVaR-0.1 | 1.0 | 0.1% | 1.0 | 3,096.0 |
| | | CVaR-0.05 | 1.0 | 0.1% | 1.0 | 3,096.0 |
| | Free Float | Stoch | 1.3 | 0.1% | 1.3 | 2,963.6 |
| | | CVaR-0.1 | 0.6 | 0.1% | 0.6 | 3,100.8 |
| | | CVaR-0.05 | 0.6 | 0.1% | 0.6 | 3,100.8 |
| 100% | Reserved Parking | Stoch | 0.2 | 0.0% | 0.2 | 2,711.1 |
| | | CVaR-0.1 | 0.0 | 0.0% | 0.0 | 2,802.7 |
| | | CVaR-0.05 | 0.0 | 0.0% | 0.0 | 2,802.7 |
| | Free Float | Stoch | 0.5 | 0.0% | 0.5 | 2,697.9 |
| | | CVaR-0.1 | 0.1 | 0.0% | 0.1 | 2,812.3 |
| | | CVaR-0.05 | 0.1 | 0.0% | 0.1 | 2,812.3 |

the proportion of one-way demand increases – at 100% one-way rentals, the proportion of denied rentals is almost zero, with even the 99th percentile being less than 0.5%, a vast improvement from over 20% when relocation costs were higher.

Table 4.7: Distribution of proportion of rentals denied with lower relocation costs ($c^{\text{rel}} = 10$)

| One-way proportion | Carsharing system | Model | Mean | Percentile | |
|---|---|---|---|---|---|
| | | | | 95% | 99% |
| 60% | Reserved Parking | Stoch | 0.08% | 0.42% | 0.82% |
| | | CVaR-0.1 | 0.13% | 0.62% | 1.00% |
| | | CVaR-0.05 | 0.13% | 0.62% | 1.11% |
| | Free Float | Stoch | 0.13% | 0.72% | 1.20% |
| | | CVaR-0.1 | 0.16% | 0.77% | 1.19% |
| | | CVaR-0.05 | 0.16% | 0.86% | 1.27% |
| 80% | Reserved Parking | Stoch | 0.00% | 0.00% | 0.21% |
| | | CVaR-0.1 | 0.00% | 0.00% | 0.00% |
| | | CVaR-0.05 | 0.00% | 0.00% | 0.00% |
| | Free Float | Stoch | 0.01% | 0.10% | 0.41% |
| | | CVaR-0.1 | 0.01% | 0.00% | 0.22% |
| | | CVaR-0.05 | 0.01% | 0.00% | 0.31% |
| 100% | Reserved Parking | Stoch | 0.01% | 0.00% | 0.21% |
| | | CVaR-0.1 | 0.00% | 0.00% | 0.00% |
| | | CVaR-0.05 | 0.00% | 0.00% | 0.00% |
| | Free Float | Stoch | 0.01% | 0.00% | 0.40% |
| | | CVaR-0.1 | 0.01% | 0.00% | 0.20% |
| | | CVaR-0.05 | 0.01% | 0.00% | 0.20% |

### 4.4.4 Computational Efficiency of MIR Procedure

In this section, we evaluate the computational efficiency of the proposed MIR procedure for varying numbers of scenarios, which is equal to the number of subproblems in the branch-and-cut algorithm. We generate samples with 100, 200, 500 and 1000 scenarios from the Boston-Cambridge Zipcar data (following the method in Section 4.4.1.1). Five samples are generated for each problem size, to obtain the average computational times. The samples are generated with a 40% proportion of one-way demand. The parameters used are identical to those described in Section 4.4.1.2. Finally, we use the values of 1 and $1/|\alpha_i^2 - \alpha_i^1|$ for all $i \in I$ for the scaling parameter $\Delta$ when generating MIR-enhanced Benders cuts when solving the risk-neutral and risk-averse models, respectively, where $\alpha_i^1$ and $\alpha_i^2$ are coefficients of $x$ in the pair of cuts used.

We compare the computational times of six different approaches to solving the problem, three with the risk-neutral model (prefixed with `Stoch`) and the other three with the `CVaR-0.05` model (prefixed with `CVaR`). For each model, we either solve directly (no suffix), solve via branch-and-cut without MIR cuts (suffixed with `-Branch`), or solve via branch-and-cut with MIR cuts (suffixed with `-MIR`). We again implement all the algorithms in Java with Gurobi 6.0.3, and do not allow pre-solve nor automatically generated cuts in Gurobi when solving the models directly.

Table 4.8 compares the computation time for each problem size, averaged over the five samples for each problem size. All times in Table 4.8 are reported in milliseconds. The computation times are decomposed into several components. The column "MP solve time" gives the total solution time in the case of solving directly (no suffix). For `-Branch` and `-MIR`, this column gives the total time spent solving of the master problem (MP) *and* updating the flow balance constants according to the current first-stage solution. The next column, relevant only to the `-Branch` and `-MIR` solution methods gives the average solution time for each subproblem (SP). In other words, this is the total amount of time spent solving the subproblems divided by the number of scenarios. Series and parallel solve times are the expected solve times if the subproblems are solved in series or in parallel, respectively. The last column gives the number of iterations required to converge on the solution, also only relevant to `-Branch` and `-MIR`.

Our results show that `-Branch` and `-MIR` are generally quicker than solving directly if their subproblems are computed in parallel. As the number of scenarios increases, the gap in parallel solve time becomes even more significant. Furthermore, both the average solve time per subproblem and the number of iterations does not change significantly as the number of scenarios increased, approximately 30–40 with all scenario sizes. The `CVaR` models generally take longer to solve than their `Stoch` counterparts. This is likely due simply to the fact that the CVaR model has more constraints than the risk-neutral model.

Finally, when comparing `-Branch` against `-MIR`, we observe that `-MIR` usually results in shorter master problem solve times for `Stoch`, but longer master problem solve times for `CVaR`. It also yields shorter solution time with parallel computing. While there is insufficient evidence to definitively show that using the MIR cuts solves the problem quicker than the traditional Benders cuts in our computation, it should be noted that the scaling parameter $\Delta$ was not, in fact, tuned for this algorithm, and that other values of $\Delta$ may result in better performance with MIR cuts.

Table 4.8: Computational time (in milliseconds) comparison between models for different problem sizes

| # scenarios (subproblems) | Model | MP solve time | Avg solve time per SP | Series solve time | Parallel solve time | # iterations |
|---|---|---|---|---|---|---|
| 100 | Stoch | 4,251 | - | 4,251 | 4,251 | - |
| | Stoch-Branch | 181 | 78 | 9,311 | 259 | 31 |
| | Stoch-MIR | 173 | 75 | 8,953 | 248 | 30 |
| | CVaR | 5,646 | - | 5,646 | 5,646 | - |
| | CVaR-Branch | 198 | 168 | 18,530 | 366 | 30 |
| | CVaR-MIR | 261 | 169 | 18,472 | 430 | 28 |
| 200 | Stoch | 13,436 | - | 13,436 | 13,436 | - |
| | Stoch-Branch | 602 | 80 | 19,703 | 682 | 32 |
| | Stoch-MIR | 659 | 82 | 20,338 | 741 | 34 |
| | CVaR | 19,537 | - | 19,537 | 19,537 | - |
| | CVaR-Branch | 516 | 166 | 36,417 | 682 | 28 |
| | CVaR-MIR | 476 | 177 | 38,554 | 653 | 27 |
| 500 | Stoch | 65,231 | - | 65,231 | 65,231 | - |
| | Stoch-Branch | 2,839 | 76 | 48,933 | 2,915 | 29 |
| | Stoch-MIR | 2,284 | 78 | 49,251 | 2,363 | 28 |
| | CVaR | 67,312 | - | 67,312 | 67,312 | - |
| | CVaR-Branch | 2,173 | 156 | 84,758 | 2,328 | 22 |
| | CVaR-MIR | 3,371 | 189 | 104,970 | 3,560 | 29 |
| 1,000 | Stoch | 236,207 | - | 236,207 | 236,207 | - |
| | Stoch-Branch | 18,529 | 252 | 291,389 | 18,781 | 39 |
| | Stoch-MIR | 16,015 | 126 | 160,272 | 16,141 | 37 |
| | CVaR | 190,072 | - | 190,072 | 190,072 | - |
| | CVaR-Branch | 16,812 | 222 | 254,038 | 17,034 | 30 |
| | CVaR-MIR | 17,416 | 307 | 343,550 | 17,723 | 39 |

## 4.5 Extensions and Future Research

We briefly describe two extensions to the carsharing model. The first is a potential improvement to the current spatial-temporal network, which may capture customer behavior more accurately. The second is an example extension illustrating the versatility of the model for making generic strategic decisions in the first stage. As these extensions have not been fully developed or tested, future research will test the benefits and viability of applying them to the carsharing model. In a separate paper, Chang et al. (2016) describes a third extension that utilizes this carsharing model to study the effect of carsharing fleet composition (e.g. gasoline, electric, hybrid) on carbon emissions and, consequently, the environmental sustainability of carsharing systems.

### 4.5.1 Parking Capacity and Waiting Arcs

This extension includes a representation a typical situation faced by customers (or employees who are relocating vehicles), in which customers are forced to wait to park their cars as there is insufficient parking capacity in the zone. This is more common in large cities, which are usually target markets for carsharing companies due to the greater inconvenience and cost of owning a vehicle.

For this extension, we start with the spatial-temporal network as constructed in Section 4.2.1. Each spatial-temporal node $n_{it}$ is divided into two nodes – the first "in" node $n_{it}^{\text{in}}$ denotes the state of zone immediately before vehicles park, and the second "out" node $n_{it}^{\text{out}}$ denotes the state of the zone immediately after vehicles park in zone $i$ in period $t$. This is illustrated visually in Figure 4.7, which represents the same spatial-temporal network in Figure 4.1. In Figure 4.7, each ellipse represents the original spatial-temporal node in Figure 4.1, with the white and black circles representing the "in" and "out" nodes of the original node, respectively. Note that the arcs on this spatial-temporal network are similar to those in the original network, except one-way, round-trip, relocation and idle arcs now connect black nodes of their original start node to white nodes of their original end nodes. That is, all arcs that were previously denoted $(n_{it}, n_{js})$ are now denoted $(n_{it}^{\text{out}}, n_{js}^{\text{in}})$.

Subsequently, two additional types of spatial-temporal arcs are appended to the network.

- Parking capacity arcs $(n_{it}^{\text{in}}, n_{it}^{\text{out}})$ for each zone $i$ and period $0 \leq t \leq T$, with capacity $w_i$ and zero flow cost.
- Waiting arcs $(n_{it}^{\text{in}}, n_{i,t+1}^{\text{in}})$ for each zone $i$ and period $0 \leq t \leq T-1$, with infinite capacity and cost $c^{\text{wait}} > c^{\text{idle}}$ per unit flow.

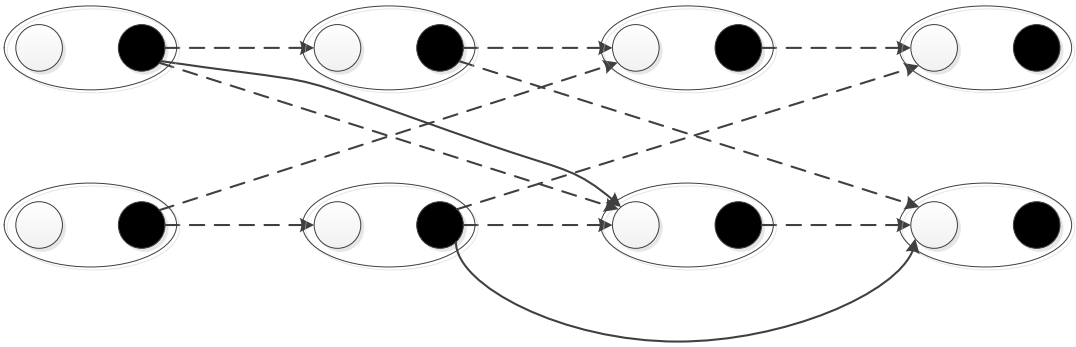


Figure 4.7: Example spatial-temporal network with split nodes

Parking capacity arcs limit the number of cars entering a zone in any one period, while waiting arcs allow any cars that cannot be parked (due to insufficient capacity) to wait until the next period to attempt to park again. Each period that a car waits (i.e. each failed parking attempt) incurs a penalty on customer satisfaction. Figure 4.8 illustrates the previous network in Figure 4.7 with parking capacity and waiting arcs appended.

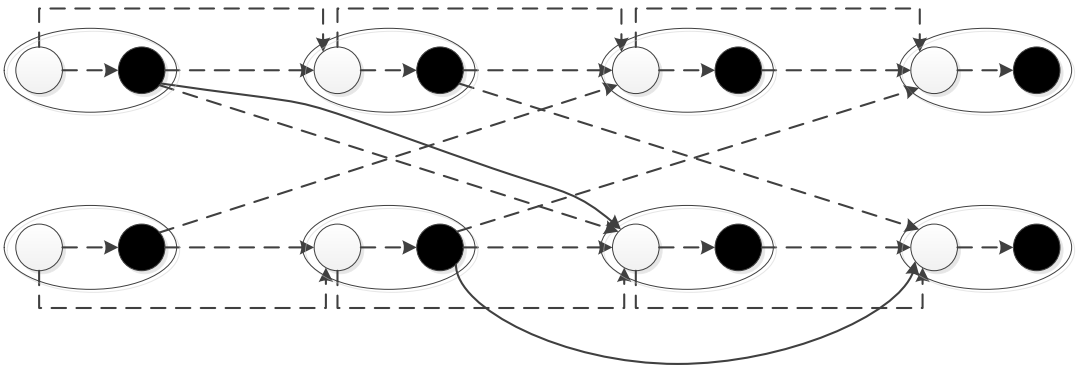


Figure 4.8: Extended spatial-temporal network

This spatial-temporal network remains compatible with the original carsharing model proposed in this chapter. With the appropriate definition of costs for waiting (e.g. $c^{\text{wait}} > c^{\text{idle}}$), this extension allows a more accurate approximation of customer behavior, particularly in cities with limited parking. Additionally, the number of waiting customers can be used as a measure of QoS and can be penalized accordingly, to encourage the purchase of more parking lots.

However, such a model is more suitable for a finer granularity of time, as it may not be desirable to impose a long minimum wait time (e.g. one hour, for a granularity of time of one hour). The finer granularity of time may render the formulation computationally

intractable due to the larger network from increasing the number of periods to represent the same horizon, in addition to the already enlarged network due to the reformulation from this extension.

### 4.5.2 Study of One-way and Round-trip Pricing

The carsharing model is very versatile in its use as a strategic decision making tool. This extension applies the same second-stage problem, but determines the pricing levels for one-way and round-trip rentals in the first stage. In other words, $r^{\text{one}}$ and $r^{\text{two}}$ are the decision variables. For ease of illustration, we only consider revenue from one-way and round-trip rentals and relocation costs in the objective function – other costs can be added to the objective as with the original model.

To optimize $r^{\text{one}}$ and $r^{\text{two}}$, we assume that we are given several candidate values for each of them, namely $r_1^{\text{one}}, \ldots, r_{m_1}^{\text{one}}$ and $r_1^{\text{two}}, \ldots, r_{m_2}^{\text{two}}$. We justify this assumption by a general preference of pricing rentals at rounded values (e.g. pricing at \$8.40 as opposed to pricing at \$8.37), and by the fact that the minimum denomination of the price is necessarily one cent. We introduce binary variables $x_1^{\text{one}}, \ldots, x_{m_1}^{\text{one}}$ and $x_1^{\text{two}}, \ldots, x_{m_2}^{\text{two}}$ such that $x_m^{\text{one}}$ takes the value 1 if the price $r_m^{\text{one}}$ is used and 0 otherwise, and $x_m^{\text{two}}$ takes the value 1 if the price $r_m^{\text{two}}$ is used and 0 otherwise. Each of the sets $\{x_m^{\text{one}}\}_{m=1,\ldots,m_1}$ and $\{x_m^{\text{two}}\}_{m=1,\ldots,m_2}$ are sets of SOS1 variables (i.e. the variables in the sets sum to one), so exactly one price each is selected for one-way and round-trip rentals.

Under these assumptions, the optimization problem we intend to solve is

$$\min_{x^{\text{one}}, x^{\text{two}}, y^1, \ldots, y^{|K|}} \quad \sum_{k \in K} p^k \left( -\left( \sum_{m=1}^{m_1} r_m^{\text{one}} x_m^{\text{one}} \right) \left( \sum_{a \in A^{\text{one}}} l_a y_a^k \right) - \left( \sum_{m=1}^{m_2} r_m^{\text{two}} x_m^{\text{two}} \right) \left( \sum_{a \in A^{\text{two}}} l_a y_a^k \right) + c^{\text{rel}} \sum_{a \in A^{\text{rel}}} l_a y_a^k \right) + g(y^1, \ldots, y^{|K|}) \tag{4.18}$$

$$\text{s.t.} \quad \sum_{m=1}^{m_1} x_m^{\text{one}} = 1 \tag{4.19}$$

$$\sum_{m=1}^{m_2} x_m^{\text{two}} = 1 \tag{4.20}$$

$$x^{\text{one}} \in \{0,1\}^{m_1} \tag{4.21}$$

$$x^{\text{two}} \in \{0,1\}^{m_2} \tag{4.22}$$

$$y^k \in Y(w, x, u^k) \qquad \forall k \in K, \tag{4.23}$$

where $l_a$ is the number of periods that arc $a$ travels over (i.e. $l_a = s - t$ for all arcs $a =$

$(n_{it}, n_{js}) \in A$). Constraints (4.19) and (4.20) ensure that exactly one pricing level each is selected for one-way and round-trip rentals, respectively. It should be noted that this model as it is currently stated will trivially set the pricing at the highest possible for maximum profit. However, this could be remedied if the QoS term ($g(y^1, \ldots, y^{|K|})$) penalizes the objective inversely proportional to price as a result of customer dissatisfaction with price, in addition to other QoS penalties.

This formulation contains bilinear terms $x_m^{\text{one}} y_a^k$ and $x_m^{\text{two}} y_a^k$ in the objective. To linearize the objective, we introduce vectors of auxiliary variables $z^{\text{one}}$ and $z^{\text{two}}$, for which the equivalence relations $z_{ma}^{\text{one},k} \equiv x_m^{\text{one}} y_m^k$ for all $m = 1, \ldots, m_1$ and $z_{ma}^{\text{two},k} \equiv x_m^{\text{two}} y_m^k$ for all $m = 1, \ldots, m_2$ hold for all $a \in A$ and $k \in K$. Then using the capacity bounds of $0 \leq y_a^k \leq f_a$, we can reformulate the problem with McCormick inequalities (see Section 3.32), yielding

$$\min_{x^{\text{one}}, x^{\text{two}}, y^1, \ldots, y^{|K|}, z^{\text{one}}, z^{\text{two}}} \quad \sum_{k \in K} p^k \left( - \sum_{a \in A^{\text{one}}} \sum_{m=1}^{m_1} r_m^{\text{one}} l_a z_{ma}^{\text{one},k} - \sum_{a \in A^{\text{two}}} \sum_{m=1}^{m_1} r_m^{\text{two}} l_a z_{ma}^{\text{two},k} + c^{\text{rel}} \sum_{a \in A^{\text{rel}}} l_a y_a^k \right) + g(y^1, \ldots, y^{|K|}) \tag{4.24}$$

s.t. (4.19)–(4.23)

$$z_{ma}^{\text{one},k} \geq y_a^k + f_a (1 - x_m^{\text{one}}) \quad \forall a \in A^{\text{one}}, k \in K, m = 1, \ldots, m_1 \tag{4.25}$$

$$z_{ma}^{\text{one},k} \leq y_a^k \quad \forall a \in A^{\text{one}}, k \in K, m = 1, \ldots, m_1 \tag{4.26}$$

$$z_{ma}^{\text{one},k} \leq f_a x_m^{\text{one}} \quad \forall a \in A^{\text{one}}, k \in K, m = 1, \ldots, m_1 \tag{4.27}$$

$$z^{\text{one}} \geq 0 \tag{4.28}$$

$$z_{ma}^{\text{two},k} \geq y_a^k + f_a (1 - x_m^{\text{two}}) \quad \forall a \in A^{\text{two}}, k \in K, m = 1, \ldots, m_2 \tag{4.29}$$

$$z_{ma}^{\text{two},k} \leq y_a^k \quad \forall a \in A^{\text{two}}, k \in K, m = 1, \ldots, m_2 \tag{4.30}$$

$$z_{ma}^{\text{two},k} \leq f_a x_m^{\text{two}} \quad \forall a \in A^{\text{two}}, k \in K, m = 1, \ldots, m_2 \tag{4.31}$$

$$z^{\text{two}} \geq 0 \tag{4.32}$$

where constraints (4.25)–(4.28) and (4.29)–(4.32) are the McCormick inequalities that establish the relations $z_{ma}^{\text{one},k} \equiv x_m^{\text{one}} y_m^k$ and $z_{ma}^{\text{two},k} \equiv x_m^{\text{two}} y_m^k$ respectively.

As a further extension of this pricing determination model, we can modify the model to optimize the pricing levels for different times of the day as well. This can be achieved fairly trivially by further indexing each $x_m^{\text{one}}$ and $x_m^{\text{two}}$ by time period $t$ as well. In other words,

we can replace $x_m^{\text{one}}$ (similarly, $x_m^{\text{two}}$) with the variables $x_{mt}^{\text{one}}$ (correspondingly, $x_{mt}^{\text{two}}$), which take value 1 if the price $r_m^{\text{one}}$ (correspondingly, $r_m^{\text{two}}$) is selected for use in time period $t$ and 0 otherwise, for all $t = 0, \ldots, T-1$. Similar to the above model, the sets $\{x_{mt}^{\text{one}}\}_{m=1,\ldots,m_1}$ and $\{x_{mt}^{\text{two}}\}_{m=1,\ldots,m_1}$ are also sets of SOS1 variables for all $t = 0, \ldots, T-1$.

## 4.6 Concluding Remarks

We proposed a two-stage stochastic integer programming model and related cutting-plane approaches for optimizing strategic vehicle allocation and parking design for carsharing systems under uncertain one-way and round-trip demand. Representing a carsharing system as a spatial-temporal network is convenient as the network is simple to characterize. The resulting problem of optimizing the movement of cars is also a convenient minimum cost flow problem, given the first-stage allocation of cars and parking lots/permits. Our results from using the model on the Boston-Cambridge Zipcar data indicated decreased profitability and QoS with higher proportions of one-way demand, corroborating the reluctance of carsharing companies in servicing one-way demand.

However, our model also indicated that with a reduction of relocation costs relative to revenue from rentals, carsharing companies could potentially enjoy increased profits and QoS. Hence, in cities with heavy one-way demand, city governments can increase the quality of carsharing service by reducing the cost burden of carsharing companies, by targeting relocation costs specifically through subsidies and grants. Companies themselves could consider decreasing relocation costs relative to their revenue by increasing the pricing of their rental gradually, to mitigate averse customer reactions to the increased prices.

Our comparisons between the risk-neutral and risk-averse model showed small but expected differences between using a risk-neutral stochastic approach and a more risk-averse CVaR approach. More importantly, the proportion of one-way rentals plays a large role in determining profits and QoS, provided the pricing per unit time of one-way and round-trip do not result in customers trivially favoring one over the other. This can be attributed to the facts that one-way rentals are generally shorter and are hence more sensitive to pricing, and that one-way rentals may unbalance the allocation of vehicles at each zone, consequently incurring relocation costs to rebalance the allocation.

Our proposed approach of solving via branch-and-cut with MIR cuts resulted in quicker solution times than using only branch-and-cut. Furthermore, solving the subproblems in parallel will always be beneficial when the number of scenarios becomes too large to be stored in memory and solving the problem directly becomes impossible.

Through the modification of the spatial-temporal network or first-stage problem, this

model can serve as a tool for studying other types of carsharing problems. Section 4.5 detailed two extensions to the current model, one of which can be used to study the effect of pricing on the profitability and QoS provided to carsharing customers. Of further interest are also the studies of the environmental and consumer impact of renting out different types of vehicles, such as cars, minivans and trucks, to serve heterogeneous customer needs.

# CHAPTER 5

# Conclusions

In this thesis, we proposed three main models for network optimization under demand uncertainty. The first model, PNDP, utilizes chance constraints to limit the probability of demand loss. Depending on the application of the PNDP, the chance constraint can join the flow balance constraints on the demand nodes in four ways, joining all constraints, joining the constraints by node or by commodity, or not joining the constraints in any way, each resulting in its own variant of PNDP. For the last variant, PNDP-cont-nc, or in special cases of PNDP-cont-n and PNDP-cont-c, we developed a polynomial-time algorithm, which solved the PNDP far more efficiently than the generic MILP reformulation.

PNDP-cont-joint was compared against PNDP-cont-nc in computational tests on random networks. The results showed that customizing QoS levels for different commodity-node pairs could result in cost savings, as opposed to having a single QoS level in joint chance constraints. The PNDP was also benchmarked against the SNDP to solve an NDP on the Sioux Falls network; it was found that PNDP solutions were less sensitive to input QoS parameters than SNDP, and the relationship between the performance of the solutions and the input QoS parameters also indicated a lesser need to tune the parameters for PNDP than for SNDP.

The second model, DR-NDP, approached NDPs with distributionally robust optimization, with the aim of tackling NDPs in which historical data is scarce. The proposed ambiguity set is marginal moment-based – its input parameters (mean and variance vectors) can be efficiently computed even for a large number of data points, and the efficiency of solving DR-NDP is independent of the size of the data. Yet, DR-NDP is more conservative than solving NDPs via traditional stochastic optimization means, and also uses more of the input data than traditional robust optimization.

We approximated the DR-NDP as a mixed-integer program, with an MILP embedded within an optimization problem. Using this approximate model, we developed a cutting-plane algorithm to obtain approximate solutions to DR-NDP. Testing on grid networks

with randomly generated demands showed that the solutions computed by DR-NDP had the desired traits of robustness and insensitivity to the observed level of demand in the small data set. However, price of this robustness is computational cost, with DR-NDP solving slower than an SAA-based model. Hence, it was proposed that DR-NDP be further tested with specific real-world applications in which high QoS is desired, yet demand data is scarce (e.g. humanitarian aid).

Finally, the third chapter proposes a carsharing framework that integrates the ability to optimize one-way and round-trip rentals, and optimize the strategic decisions in one model. Although the proposed model only optimizes the purchasing of parking and free-float permits in the first-stage problem, it is versatile enough to optimize other strategic decisions (e.g. the pricing of rentals, briefly described in the extensions for the model) without change to the spatial-temporal network in the second stage.

A risk-neutral and a risk-averse CVaR penalty approach were used in two separate models to manage QoS of the carsharing company. We reformulated both models as MILPs and presented a branch-and-cut algorithm with a MIR procedure to solve them. In addition to the advantage of being able to solve the carsharing model efficiently via parallel computing, the branch-and-cut algorithm was also shown to be more efficient with MIR cuts than without.

Our computational results on Zipcar data in the Boston-Cambridge area gave much insight to how carsharing companies might mitigate the decreased profitability and QoS brought by one-way rentals. We proposed a reduction of relocation costs relative to revenue (via government subsidies or increasing rental pricing) as a means to reduce the impact of one-way rentals. Through our results, we also found that the proportion of one-way trips plays an important role in determining profits and QoS, implying a greater need for the reduction of relocation costs in cities with high one-way demand.

The three models proposed have provided different risk-averse approaches to solving NDPs. Through this thesis, we have shown the effectiveness of these risk-averse models in obtaining conservative solutions to the NDP. Even though the DR-NDP may have limited situations in which it can be applied, the PNDP and carsharing model have potential to be customized to solve a larger variety of problems. In particular, the carsharing model can be easily yet extensively modified both in its first-stage strategic decisions and in its second-stage approximation of customer behavior. With the great potential to specialize the carsharing model to make a wide variety of strategic decisions, it is this writer's hope that this research can be further expanded to solve other stochastic problems in network design.

# APPENDIX A

# Benders Decomposition Approach for all SNDPs (Chapter 1)

We note that SNDPs require that their flow variables for each scenario be optimal for that particular scenario, but also require the capacity design to be common across all scenarios. This nature of SNDPs lends itself nicely to a Benders decomposition approach when we optimize SNDPs. We use SNDP-bin-wp to illustrate the use of Benders decomposition as it is the most complex of the four SNDP variants. However, this approach can be used for all variants of SNDP.

Let the capacity design variables $\beta$ be the first-stage variables, and the recourse flow variables $y^k$ and variables representing unmet demand $t^k$ be second-stage variables. In the *master problem* **[MP]**, we optimize the objective over relaxed constraints on $\beta$ and on $\theta$, the lower bounds on the subproblem in each scenario. At each iteration, **[MP]** is solved to obtain a trial solution $(\beta, \theta)$, which is passed to the subproblems described later. Initially, the feasibility and optimality cut sets $L_1(\beta) \geq 0$ and $L_2(\beta, \theta) \geq 0$ have no cuts.

**[MP]:**

$$\min \quad \sum_{(i,j)\in A} q_{ij}\beta_{ij} + \sum_{k\in K} p^k\theta_k \tag{A.1}$$

$$\text{s.t.} \quad L_1(\beta) \geq 0 \tag{A.2}$$

$$L_2(\beta, \theta_1, \ldots, \theta_{|K|}) \geq 0 \tag{A.3}$$

$$\beta_{ij} \in \{0, 1\} \quad \forall (i,j) \in A \tag{A.4}$$

We denote the *subproblems* by **[SP($k, \beta$)]**. For each scenario $k$, the trial solution obtained from **[MP]** is used as a parameter to obtain an optimal $(y^k, t^k)$ through **[SP($k, \beta$)]**. A feasibility or optimality cut is generated depending on whether **[SP($k, \beta$)]** is infeasible, or has an optimal value that is greater than $\theta_k$, and the cut is appended to the appropriate

cut set. (Problem **[SP($k, \beta$)]** is never unbounded since its optimal value is non-negative.) If the optimal value for **[SP($k, \beta$)]** is at most $\theta_k$, then the solution on hand is declared to be optimal for SNDP-bin-wp.

We note that while **[SP($k, \beta$)]** is not required in the algorithm to generate cuts, we provide its formulation for the sake of completeness.

**[SP($k, \beta$)]:**

$$\min \quad \sum_{w \in W} \left( \sum_{(i,j) \in A} a_{ijw} y^k_{ijw} + \sum_{i \in D_w} G_{iw} t^k_{iw} \right) \tag{A.5}$$

$$\text{s.t.} \quad \sum_{w \in W} y^k_{ijw} \leq u_{ij} \beta_{ij} \quad \forall (i,j) \in A \tag{A.6}$$

$$\sum_{j:(i,j) \in A} y^k_{ijw} - \sum_{j:(j,i) \in A} y^k_{jiw} \leq s_{iw} \quad \forall i \in S_w,\ w \in W \tag{A.7}$$

$$\sum_{j:(i,j) \in A} y^k_{ijw} - \sum_{j:(j,i) \in A} y^k_{jiw} = 0 \quad \forall i \in N \backslash (S_w \cup D_w),\ w \in W \tag{A.8}$$

$$- \sum_{j:(i,j) \in A} y^k_{ijw} + \sum_{j:(j,i) \in A} y^k_{jiw} + t^k_{iw} \geq d^k_{iw} \quad \forall i \in D_w,\ w \in W \tag{A.9}$$

$$y^k \geq 0,\ t^k \geq 0 \tag{A.10}$$

To determine whether **[SP($k, \beta$)]** is feasible, we determine whether the *dual* **[D-SP**$(k, \beta)$**]** of the subproblem is unbounded.

**[D-SP**$(k, \beta)$**]:**

$$\max \quad \sum_{(i,j) \in A} (u_{ij} \beta_{ij}) \mu^k_{ij} + \sum_{w \in W} \left( \sum_{i \in S_w} s_w \pi^k_{iw} - \sum_{i \in D_w} d^k_{iw} \pi^k_{iw} \right) \tag{A.11}$$

$$\text{s.t.} \quad \mu^k_{ij} + \pi^k_{iw} - \pi^k_{jw} \leq a_{ijw} \quad \forall (i,j) \in A,\ w \in W \tag{A.12}$$

$$\mu^k_{ij} \leq 0 \quad \forall (i,j) \in A \tag{A.13}$$

$$\pi^k_{iw} \leq 0 \quad \forall i \in S_w,\ w \in W \tag{A.14}$$

$$\pi^k_{iw} \leq v_{iw} \quad \forall i \in D_w,\ w \in W \tag{A.15}$$

where $\mu^k_{ij}$ and $\pi^k_{iw}$ are the dual variables associated with the constraints (A.6) and (A.7)–(A.9) respectively. In other words, we determine if there exists an unbounded dual direction. To do this, we solve the *separation problem* **[S-SP**$(k, \beta)$**]**.

**[S-SP$(k, \beta)$]:**

$$
\begin{aligned}
\max \quad & \sum_{(i,j)\in A} (u_{ij}\beta_{ij})\mu_{ij}^k + \sum_{w\in W}\left(\sum_{i\in S_w} s_w \pi_{iw}^k - \sum_{i\in D_w} d_{iw}^k \pi_{iw}^k\right) \\
\text{s.t.} \quad & \mu_{ij}^k + \pi_{iw}^k - \pi_{jw}^k \leq 0 \quad \forall (i,j)\in A,\ w\in W \\
& \mu_{ij}^k \leq 0 \quad \forall (i,j)\in A \\
& \pi_{iw}^k \leq 0 \quad \forall i \in S_w \cup D_w,\ w\in W
\end{aligned}
$$

Its feasible region consists of vectors that, when added to feasible solutions of **[D-SP$(k, \beta)$]**, does not change their feasibility in **[D-SP$(k, \beta)$]**. If its objective function is positive, then an unbounded dual direction exists, and the optimal solution in this case would be an unbounded dual direction of **[D-SP$(k, \beta)$]**. Algorithm A.1 describes the algorithm in greater detail.

**Remark A.2.** It can be observed that Algorithm A.1 can easily be modified to solve the other three variants of SNDP by removing the penalty term in the case without penalty, or by changing the binary design variables to continuous design variables. For either modification (or both together), the second stage is still a linear program, so Benders decomposition remains a viable algorithm to solve the problem.

The Benders approach can also be applied to optimize PNDP-bin-nc. As mentioned in Section 2.3.3, replacing the continuous variables $x$ with the binary variables $\beta$ increases the difficulty of the problem. To reduce the difficulty in solving PNDP-bin-nc, we can decompose the problem instead of solving the MILP directly, again with the network design variables as the first-stage variables and the flow variables as the second-stage variables, resulting in a linear program in the second stage. The above methodology can then be applied appropriately, with careful consideration to the fact that there is only one subproblem when decomposing PNDP-bin-nc, instead of the $|K|$ subproblems when decomposing SNDP-bin-wp in Algorithm A.1. □

**Remark A.3.** It is also important to note that **[SP($k, \beta$)]** is, in fact, always feasible, because there always exists the feasible solution of having zero flow and all demands unsatisfied. This is also true of the subproblems of SNDP-cont-wp. However, the feasibility check is included in Algorithm A.1 to illustrate a more general Benders approach that can be applied to the other problems mentioned in Remark A.2. □

**Algorithm A.1** Generalized Benders decomposition algorithm for SNDP-bin-wp

Initialize iteration number $m = 0$
Initialize **[MP]** without any cuts in $L_1(\beta) \geq 0$ and $L_2(\beta, \theta) \geq 0$.
Solve **[MP]** and obtain an optimal solution $(\beta^m, \theta^m)$. If there are no cuts in $L_2(\beta, \theta) \geq 0$, let $\theta_k^m = -\infty$ for all $k \in K$.
**repeat**
  Increment $m$ by 1.
  **for all** $k \in K$ **do**
    **Feasibility check:**
    Solve **[S-SP**$(k, \beta^m)$**]**.
    **if** the optimal value of **[S-SP**$(k, \beta^m)$**]** is positive **then**
      Let $(\mu^{km}, \pi^{km})$ be the optimal solution of **[S-SP**$(k, \beta^m)$**]**, and generate

$$\sum_{(i,j)\in A} (u_{ij}\mu_{ij}^{km})\beta_{ij} + \sum_{w\in W} \left( \sum_{i\in S_w} s_w \pi_{iw}^{km} - \sum_{i\in D_w} d_{iw}^k \pi_{iw}^{km} \right) \leq 0$$

      into the cut set $L_1(\beta) \geq 0$ in **[MP]**.
    **else**
      **Optimality check:**
      Solve **[D-SP**$(k, \beta^m)$**]**.
      **if** the optimal value of **[D-SP**$(k, \beta^m)$**]** $> \theta_k^m$ **then**
        Let $(\mu^{km}, \pi^{km})$ be the optimal solution of **[D-SP**$(k, \beta^m)$**]**, and generate

$$\sum_{(i,j)\in A} (u_{ij}\mu_{ij}^{km})\beta_{ij} + \sum_{w\in W} \left( \sum_{i\in S_w} s_w \pi_{iw}^{km} - \sum_{i\in D_w} d_{iw}^k \pi_{iw}^{km} \right) \leq \theta_k$$

        into the cut set $L_2(\beta, \theta) \geq 0$ in **[MP]**.
      **end if**
    **end if**
  **end for**
**until** No cuts were added in iteration $m$.
**return** Optimal solution $(\beta^m, \theta^m)$.

# APPENDIX B

# Computational Results for DR-NDP Performance Analysis (Chapter 2)

This appendix fully describes all the computational results that were discussed in Section 3.5. In the table, the results for six different networks are listed, namely the NOBEL-US network, and the 3x3 to 7x7 grid networks. For each observed demand level and model type, the table lists the solve time, the number of cuts (relevant only to DR-NDP), the capacity cost of the solution computed by the model, the optimal value and second stage values as computed by the model, and the expected second stage, flow cost, penalty cost and total cost values from computing the performance of the solution with the reference set. Note that the quantities in the table have the relation

$$\begin{aligned}
&\mathbb{E}[\text{Total cost}] \\
&= \text{Capacity cost} + \mathbb{E}[\text{2nd stage cost}] \\
&= \text{Capacity cost} + \mathbb{E}[\text{Flow cost}] + \mathbb{E}[\text{Penalty cost}].
\end{aligned}$$

Table B.1: Computational results for DR-NDP performance analysis

| Network | Demand | Model | Solve time | # cuts | Capacity cost | Model opt value | Model 2nd stage | $\mathbb{E}$[2nd stage] | $\mathbb{E}$[Flow cost] | $\mathbb{E}$[Penalty cost] | $\mathbb{E}$[Total cost] |
|---|---|---|---|---|---|---|---|---|---|---|---|
| NOBEL-US | normal | drndp | 110,928 | 272 | 3,484.15 | 36,867.33 | 33,383.17 | 26,688.95 | 2,047.28 | 24,641.67 | 30,173.10 |
| NOBEL-US | normal | gamma50 | 25 | 0 | 5,206.80 | 20,188.13 | 14,981.33 | 18,645.38 | 2,361.93 | 16,283.45 | 23,852.18 |
| NOBEL-US | normal | gamma1000 | 4,566 | 0 | 5,172.70 | 20,808.81 | 15,636.11 | 18,577.70 | 2,402.14 | 16,175.56 | 23,750.40 |
| NOBEL-US | normal | uniform50 | 36 | 0 | 5,347.70 | 23,523.47 | 18,175.77 | 18,797.72 | 2,420.52 | 16,377.20 | 24,145.42 |
| NOBEL-US | normal | uniform1000 | 4,571 | 0 | 5,300.80 | 22,947.25 | 17,646.45 | 18,537.01 | 2,386.56 | 16,150.45 | 23,837.81 |
| NOBEL-US | low | drndp | 103,113 | 310 | 3,380.12 | 18,775.75 | 15,395.63 | 27,196.63 | 2,054.57 | 25,142.06 | 30,576.75 |
| NOBEL-US | low | gamma50 | 20 | 0 | 4,310.40 | 12,944.73 | 8,634.33 | 21,624.79 | 2,285.91 | 19,338.88 | 25,935.19 |
| NOBEL-US | low | gamma1000 | 2,605 | 0 | 4,606.20 | 9,900.92 | 5,294.72 | 19,967.30 | 2,337.80 | 17,629.49 | 24,573.50 |
| NOBEL-US | low | uniform50 | 25 | 0 | 4,335.20 | 7,081.80 | 2,746.60 | 21,646.05 | 2,244.25 | 19,401.79 | 25,981.25 |
| NOBEL-US | low | uniform1000 | 3,870 | 0 | 4,365.20 | 9,651.45 | 5,286.25 | 21,402.57 | 2,241.08 | 19,161.49 | 25,767.77 |
| NOBEL-US | very low | drndp | 71,919 | 313 | 3,094.48 | 6,293.87 | 3,199.39 | 30,863.29 | 1,973.34 | 28,889.95 | 33,957.78 |
| NOBEL-US | very low | gamma50 | 10 | 0 | 2,982.80 | 4,125.74 | 1,142.94 | 41,678.74 | 1,852.44 | 39,826.30 | 44,661.54 |
| NOBEL-US | very low | gamma1000 | 1,240 | 0 | 2,698.80 | 4,060.60 | 1,361.80 | 42,897.88 | 1,766.28 | 41,131.60 | 45,596.68 |
| NOBEL-US | very low | uniform50 | 15 | 0 | 2,173.20 | 3,143.96 | 970.76 | 59,566.05 | 1,473.39 | 58,092.66 | 61,739.25 |
| NOBEL-US | very low | uniform1000 | 860 | 0 | 2,247.50 | 3,260.93 | 1,013.43 | 53,853.54 | 1,556.56 | 52,296.97 | 56,101.04 |
| Grid 3x3 | normal | drndp | 62,280 | 153 | 4,730.17 | 12,497.46 | 7,767.29 | 5,835.94 | 2,474.93 | 3,361.01 | 10,566.12 |
| Grid 3x3 | normal | gamma50 | 15 | 0 | 5,040.10 | 11,448.65 | 6,408.55 | 5,301.38 | 2,505.70 | 2,795.68 | 10,341.48 |
| Grid 3x3 | normal | gamma1000 | 2,181 | 0 | 4,775.20 | 11,137.85 | 6,362.65 | 5,515.83 | 2,570.70 | 2,945.14 | 10,291.03 |
| Grid 3x3 | normal | uniform50 | 20 | 0 | 4,998.70 | 10,377.15 | 5,378.45 | 5,399.61 | 2,593.56 | 2,806.04 | 10,398.31 |
| Grid 3x3 | normal | uniform1000 | 1,665 | 0 | 4,999.90 | 10,786.90 | 5,787.00 | 5,331.57 | 2,578.93 | 2,752.63 | 10,331.47 |
| Grid 3x3 | low | drndp | 42,800 | 150 | 4,030.68 | 8,729.62 | 4,698.94 | 7,220.50 | 2,226.96 | 4,993.53 | 11,251.18 |
| Grid 3x3 | low | gamma50 | 15 | 0 | 4,560.20 | 8,238.40 | 3,678.20 | 7,029.63 | 2,306.21 | 4,723.42 | 11,589.83 |
| Grid 3x3 | low | gamma1000 | 1,895 | 0 | 4,181.10 | 7,467.56 | 3,286.46 | 6,868.22 | 2,320.49 | 4,547.73 | 11,049.32 |
| Grid 3x3 | low | uniform50 | 10 | 0 | 4,311.30 | 7,181.62 | 2,870.32 | 6,702.92 | 2,381.18 | 4,321.74 | 11,014.22 |
| Grid 3x3 | low | uniform1000 | 1,431 | 0 | 4,443.10 | 7,383.72 | 2,940.62 | 6,407.87 | 2,406.43 | 4,001.44 | 10,850.97 |
| Grid 3x3 | very low | drndp | 7,424 | 140 | 1,645.20 | 3,005.04 | 1,359.85 | 15,788.81 | 1,277.19 | 14,511.62 | 17,434.01 |
| Grid 3x3 | very low | gamma50 | 10 | 0 | 1,497.90 | 2,473.99 | 976.09 | 17,610.20 | 1,154.45 | 16,455.75 | 19,108.10 |
| Grid 3x3 | very low | gamma1000 | 1,325 | 0 | 1,597.00 | 2,658.22 | 1,061.22 | 16,807.20 | 1,180.39 | 15,626.80 | 18,404.20 |
| Grid 3x3 | very low | uniform50 | 10 | 0 | 1,527.10 | 2,496.19 | 969.09 | 17,700.50 | 1,117.44 | 16,583.06 | 19,227.60 |
| Grid 3x3 | very low | uniform1000 | 1,395 | 0 | 1,580.20 | 2,559.19 | 978.99 | 17,113.73 | 1,149.79 | 15,963.94 | 18,693.93 |
| Grid 4x4 | normal | drndp | 144,597 | 355 | 3,325.40 | 22,726.98 | 19,401.58 | 16,066.44 | 3,022.78 | 13,043.65 | 19,391.84 |
| Grid 4x4 | normal | gamma50 | 27 | 0 | 3,676.10 | 15,009.59 | 11,333.49 | 13,762.76 | 3,028.13 | 10,734.63 | 17,438.86 |
| Grid 4x4 | normal | gamma1000 | 6,236 | 0 | 3,987.20 | 15,312.96 | 11,325.76 | 12,731.33 | 3,052.59 | 9,678.73 | 16,718.53 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Grid 4x4 | normal | uniform50 | 40 | 0 | 3,974.30 | 12,854.96 | 8,880.66 | 13,421.12 | 3,019.60 | 10,401.52 | 17,395.42 |
| Grid 4x4 | normal | uniform1000 | 5,796 | 0 | 3,954.40 | 15,018.26 | 11,063.86 | 13,361.19 | 3,025.63 | 10,335.57 | 17,315.59 |
| Grid 4x4 | low | drndp | 115,689 | 369 | 3,190.53 | 12,713.25 | 9,522.72 | 16,398.36 | 2,985.49 | 13,412.88 | 19,588.89 |
| Grid 4x4 | low | gamma50 | 25 | 0 | 3,582.10 | 8,087.54 | 4,505.44 | 15,582.29 | 2,919.09 | 12,663.20 | 19,164.39 |
| Grid 4x4 | low | gamma1000 | 4,421 | 0 | 3,603.40 | 8,095.16 | 4,491.76 | 14,268.75 | 2,982.62 | 11,286.14 | 17,872.15 |
| Grid 4x4 | low | uniform50 | 35 | 0 | 3,369.70 | 7,765.12 | 4,395.42 | 17,459.82 | 2,878.18 | 14,581.64 | 20,829.52 |
| Grid 4x4 | low | uniform1000 | 5,731 | 0 | 3,357.70 | 7,371.70 | 4,014.00 | 16,767.55 | 2,906.60 | 13,860.95 | 20,125.25 |
| Grid 4x4 | very low | drndp | 43,233 | 247 | 1,876.82 | 4,358.05 | 2,481.23 | 32,941.72 | 2,174.60 | 30,767.12 | 34,818.54 |
| Grid 4x4 | very low | gamma50 | 10 | 0 | 1,493.60 | 2,735.20 | 1,241.60 | 43,567.18 | 1,586.91 | 41,980.27 | 45,060.78 |
| Grid 4x4 | very low | gamma1000 | 845 | 0 | 1,591.60 | 2,834.55 | 1,242.95 | 41,405.77 | 1,567.57 | 39,838.19 | 42,997.37 |
| Grid 4x4 | very low | uniform50 | 10 | 0 | 1,416.50 | 2,429.96 | 1,013.46 | 45,611.13 | 1,513.35 | 44,097.78 | 47,027.63 |
| Grid 4x4 | very low | uniform1000 | 1,200 | 0 | 1,406.70 | 2,437.48 | 1,030.78 | 45,728.96 | 1,513.28 | 44,215.69 | 47,135.66 |
| Grid 5x5 | normal | drndp | 185,190 | 688 | 6,193.99 | 45,395.61 | 39,201.63 | 25,241.09 | 4,236.08 | 21,005.01 | 31,435.07 |
| Grid 5x5 | normal | gamma50 | 80 | 0 | 5,973.10 | 23,397.92 | 17,424.82 | 22,687.10 | 4,247.56 | 18,439.54 | 28,660.20 |
| Grid 5x5 | normal | gamma1000 | 11,642 | 0 | 6,971.70 | 28,867.92 | 21,896.22 | 20,600.94 | 4,353.00 | 16,247.94 | 27,572.64 |
| Grid 5x5 | normal | uniform50 | 120 | 0 | 6,274.10 | 28,311.92 | 22,037.82 | 21,457.91 | 4,316.27 | 17,141.64 | 27,732.01 |
| Grid 5x5 | normal | uniform1000 | 15,167 | 0 | 6,255.70 | 27,808.06 | 21,552.36 | 21,757.30 | 4,292.96 | 17,464.34 | 28,013.00 |
| Grid 5x5 | low | drndp | 294,715 | 908 | 5,527.98 | 30,469.83 | 24,941.85 | 25,242.73 | 4,171.82 | 21,070.91 | 30,770.71 |
| Grid 5x5 | low | gamma50 | 65 | 0 | 5,886.20 | 15,363.11 | 9,476.91 | 22,107.90 | 4,278.02 | 17,829.88 | 27,994.10 |
| Grid 5x5 | low | gamma1000 | 9,747 | 0 | 5,904.40 | 17,003.96 | 11,099.56 | 22,249.02 | 4,259.00 | 17,990.02 | 28,153.42 |
| Grid 5x5 | low | uniform50 | 75 | 0 | 5,314.10 | 14,774.21 | 9,460.11 | 26,055.32 | 4,142.15 | 21,913.16 | 31,369.42 |
| Grid 5x5 | low | uniform1000 | 13,602 | 0 | 5,173.40 | 14,479.17 | 9,305.77 | 25,818.92 | 4,176.68 | 21,642.24 | 30,992.32 |
| Grid 5x5 | very low | drndp | 185,899 | 904 | 4,694.49 | 12,028.99 | 7,334.50 | 30,208.16 | 4,061.02 | 26,147.14 | 34,902.65 |
| Grid 5x5 | very low | gamma50 | 35 | 0 | 3,772.30 | 6,228.07 | 2,455.77 | 51,999.67 | 3,859.44 | 48,140.23 | 55,771.97 |
| Grid 5x5 | very low | gamma1000 | 4,720 | 0 | 3,835.00 | 6,719.40 | 2,884.40 | 38,478.44 | 3,973.93 | 34,504.50 | 42,313.44 |
| Grid 5x5 | very low | uniform50 | 30 | 0 | 2,995.10 | 5,321.31 | 2,326.21 | 68,953.03 | 3,421.35 | 65,531.68 | 71,948.13 |
| Grid 5x5 | very low | uniform1000 | 3,956 | 0 | 2,935.80 | 5,314.37 | 2,378.57 | 66,166.33 | 3,513.28 | 62,653.05 | 69,102.13 |
| Grid 6x6 | normal | drndp | 531,350 | 1,409 | 5,827.97 | 104,044.47 | 98,216.50 | 76,251.09 | 3,397.06 | 72,854.03 | 82,079.06 |
| Grid 6x6 | normal | gamma50 | 240 | 0 | 9,368.10 | 68,469.30 | 59,101.20 | 53,110.28 | 3,599.54 | 49,510.74 | 62,478.38 |
| Grid 6x6 | normal | gamma1000 | 42,977 | 0 | 8,569.40 | 57,827.31 | 49,257.91 | 53,980.65 | 3,592.83 | 50,387.83 | 62,550.05 |
| Grid 6x6 | normal | uniform50 | 255 | 0 | 8,092.70 | 59,923.24 | 51,830.54 | 60,599.33 | 3,577.04 | 57,022.29 | 68,692.03 |
| Grid 6x6 | normal | uniform1000 | 36,025 | 0 | 7,788.90 | 54,835.40 | 47,046.50 | 60,088.17 | 3,549.50 | 56,538.66 | 67,877.07 |
| Grid 6x6 | low | drndp | 463,441 | 1,738 | 6,187.79 | 50,891.19 | 44,703.39 | 77,175.93 | 3,382.24 | 73,793.69 | 83,363.72 |
| Grid 6x6 | low | gamma50 | 125 | 0 | 7,225.60 | 16,465.35 | 9,239.75 | 61,259.11 | 3,529.87 | 57,729.24 | 68,484.71 |
| Grid 6x6 | low | gamma1000 | 20,859 | 0 | 7,401.40 | 21,143.66 | 13,742.26 | 57,357.55 | 3,578.58 | 53,778.97 | 64,758.95 |
| Grid 6x6 | low | uniform50 | 170 | 0 | 6,155.90 | 16,829.31 | 10,673.41 | 71,791.93 | 3,393.00 | 68,398.94 | 77,947.83 |

| | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Grid 6x6 | low | uniform1000 | 40,061 | 0 | 5,815.60 | 18,391.51 | 12,575.91 | 71,085.99 | 3,409.44 | 67,676.55 | 76,901.59 |
| Grid 6x6 | very low | drndp | 462,732 | 1,606 | 6,561.47 | 29,928.26 | 23,366.79 | 75,111.81 | 3,417.97 | 71,693.84 | 81,673.29 |
| Grid 6x6 | very low | gamma50 | 95 | 0 | 5,421.50 | 10,162.62 | 4,741.12 | 72,770.76 | 3,419.71 | 69,351.05 | 78,192.26 |
| Grid 6x6 | very low | gamma1000 | 11,607 | 0 | 5,649.00 | 9,646.66 | 3,997.66 | 68,406.49 | 3,441.42 | 64,965.07 | 74,055.49 |
| Grid 6x6 | very low | uniform50 | 65 | 0 | 3,160.60 | 5,181.03 | 2,020.43 | 114,545.11 | 3,054.02 | 111,491.09 | 117,705.71 |
| Grid 6x6 | very low | uniform1000 | 14,237 | 0 | 3,506.90 | 5,846.36 | 2,339.46 | 105,437.73 | 3,097.29 | 102,340.44 | 108,944.63 |
| Grid 7x7 | normal | drndp | 453,094 | 1,241 | 4,915.98 | 138,540.07 | 133,624.09 | 87,795.36 | 2,235.85 | 85,559.51 | 92,711.34 |
| Grid 7x7 | normal | gamma50 | 585 | 0 | 8,395.90 | 82,820.90 | 74,425.00 | 58,390.69 | 2,466.83 | 55,923.86 | 66,786.59 |
| Grid 7x7 | normal | gamma1000 | 45,573 | 0 | 7,644.40 | 71,069.51 | 63,425.11 | 59,515.29 | 2,452.86 | 57,062.42 | 67,159.69 |
| Grid 7x7 | normal | uniform50 | 701 | 0 | 5,367.90 | 73,438.15 | 68,070.25 | 64,516.52 | 2,374.61 | 62,141.91 | 69,884.42 |
| Grid 7x7 | normal | uniform1000 | 114,202 | 0 | 5,589.90 | 69,927.92 | 64,338.02 | 64,084.76 | 2,383.54 | 61,701.22 | 69,674.66 |
| Grid 7x7 | low | drndp | 507,749 | 1,646 | 4,713.25 | 75,696.23 | 70,982.98 | 87,544.33 | 2,160.79 | 85,383.54 | 92,257.57 |
| Grid 7x7 | low | gamma50 | 330 | 0 | 7,275.20 | 41,829.15 | 34,553.95 | 64,400.59 | 2,380.53 | 62,020.06 | 71,675.79 |
| Grid 7x7 | low | gamma1000 | 26,434 | 0 | 5,434.40 | 31,968.97 | 26,534.57 | 63,571.82 | 2,401.82 | 61,170.01 | 69,006.22 |
| Grid 7x7 | low | uniform50 | 360 | 0 | 4,054.90 | 23,769.96 | 19,715.06 | 83,638.38 | 2,230.56 | 81,407.82 | 87,693.28 |
| Grid 7x7 | low | uniform1000 | 36,295 | 0 | 4,312.20 | 31,256.76 | 26,944.56 | 83,073.45 | 2,233.40 | 80,840.05 | 87,385.65 |
| Grid 7x7 | very low | drndp | 467,700 | 1,320 | 4,716.65 | 47,420.66 | 42,704.02 | 82,013.79 | 2,269.03 | 79,744.76 | 86,730.43 |
| Grid 7x7 | very low | gamma50 | 290 | 0 | 5,552.70 | 12,366.15 | 6,813.45 | 92,760.35 | 2,442.72 | 90,317.63 | 98,313.05 |
| Grid 7x7 | very low | gamma1000 | 18,098 | 0 | 5,328.50 | 14,866.38 | 9,537.88 | 74,544.93 | 2,509.00 | 72,035.93 | 79,873.43 |
| Grid 7x7 | very low | uniform50 | 275 | 0 | 3,618.90 | 12,548.05 | 8,929.15 | 130,786.47 | 2,313.92 | 128,472.55 | 134,405.37 |
| Grid 7x7 | very low | uniform1000 | 96,905 | 0 | 3,650.50 | 8,760.34 | 5,109.84 | 130,173.17 | 2,276.27 | 127,896.90 | 133,823.67 |

# APPENDIX C

# Branch-and-Cut Algorithm with MIR Procedure (Chapter 3)

We outline the branch-and-cut algorithm with MIR below in pseudo-code. Algorithms C.1 and C.2 describe the algorithms for the risk-neutral model and CVaR model, respectively. Both algorithms are similar, with differences mainly in the master problem and subproblem formulations and the form of the Benders cuts added. For ease of referencing the coefficients in the cuts, we define $\alpha$, $\beta$, and $\gamma$ as the coefficients of variables $w$, $x$ and $v$, respectively, and $-\delta$ as the constant in the cut. In other words, the cuts for the risk-neutral model are of the form

$$q^k + \sum_{i \in I} \alpha_i w_i + \sum_{i \in I} \beta_i x_i - \delta \geq 0$$

while the cuts for the risk-averse CVaR model are of the form

$$q^k + \sum_{i \in I} \alpha_i w_i + \sum_{i \in I} \beta_i x_i + \gamma v - \delta \geq 0.$$

There are two main parts to the algorithm. The outer algorithm is a regular branch-and-cut algorithm that branches on fractional $x_i$'s and adds cuts generated from the subproblems. The inner algorithm (lines 14–37 in both algorithms) is a Benders decomposition algorithm with an additional MIR procedure that pairs Benders cuts to generate additional valid cuts. The most violated cut is generated to the master problem, while the remaining cuts are stored for subsequent pairings with the MIR procedure.

**Algorithm C.1** Branch-and-cut algorithm with MIR for risk-neutral model

Initialize **MP-Stoch** with no cuts
Initialize `sol`:=`null`
Initialize `optval`:= 0
**for** $k \in K$ **do**
    Initialize $\texttt{cutlist}^k$
    Add cut $q^k \geq 0$ to $\texttt{cutlist}^k$
    Add cut $q^k \geq 0$ to $L^k_{\text{Stoch}} \geq 0$
**end for**
Initialize $S := \{0\}$
Initialize $s := 0$
Define $\texttt{Problem}_0$ as **MP-Stoch**
**while** $S \neq \emptyset$ **do**
    Define $\overline{s} := \min\{s : s \in S\}$
    **repeat**
        Solve $\texttt{Problem}_{\overline{s}}$ to obtain optimal solution $(\widetilde{q}, \widetilde{w}, \widetilde{x})$ and objective $\widetilde{\texttt{obj}}$
        **for** $k \in K$ **do**
            Solve **SP-Stoch**$_k(\widetilde{w}, \widetilde{x})$ to obtain optimal dual solution $(\widehat{\pi}, \widehat{\lambda})$ and optimal objective $\widehat{q}^k$
            **if** $\widehat{q}^k > \widetilde{q}^k$ **then**
                Define `cutA` as cut (4.12)
                Add `cutA` to $\texttt{cutlist}^k$
                **for** `cutB` $\in \texttt{cutlist}^k$ **do**
                    $\Delta \leftarrow 1$
                    Apply Theorem 4.4 to `cutA` and `cutB` to generate $\texttt{cutC}_0$
                    Add $\texttt{cutC}_0$ to $\texttt{cutlist}^k$
                    **for** $i \in I$ **do**
                        $\Delta \leftarrow \frac{1}{|\alpha_i^{\texttt{B}} - \alpha_i^{\texttt{A}}|}$, where $\alpha^{\texttt{A}}$ and $\alpha^{\texttt{B}}$ are coefficients of $w$ in `cutA` and `cutB` respectively
                        Apply Theorem 4.4 to `cutA` and `cutB` to generate $\texttt{cutC}_i$
                        Add $\texttt{cutC}_i$ to $\texttt{cutlist}^k$
                        $\Delta \leftarrow \frac{1}{|\beta_i^{\texttt{B}} - \beta_i^{\texttt{A}}|}$, where $\beta^{\texttt{A}}$ and $\beta^{\texttt{B}}$ are coefficients of $x$ in `cutA` and `cutB` respectively
                        Apply Theorem 4.4 to `cutA` and `cutB` to generate $\texttt{cutD}_i$
                        Add $\texttt{cutD}_i$ to $\texttt{cutlist}^k$
                    **end for**
                **end for**
                Among `cutA` and $\texttt{cutC}_i$ and $\texttt{cutD}_i$ $\forall i \in I$, add to $L^k_{\text{Stoch}} \geq 0$ the cut with the smallest

$$\frac{\delta - (\widetilde{q}^k + \sum_{i \in I} \alpha_i \widetilde{w}_i + \sum_{i \in I} \beta_i \widetilde{x}_i)}{1 + \sum_{i \in I} (\alpha_i)^2 + \sum_{i \in I} (\beta_i)^2}$$

            **end if**
        **end for**
    **until** No cut is added
    **if** $\widetilde{\texttt{obj}}$<`optval` **then**
        **if** $\exists i : \widetilde{x}_i - \lfloor \widetilde{x}_i \rfloor \neq 0$ **then**
            Define $\overline{i} := \min\{i : \widetilde{x}_i - \lfloor \widetilde{x}_i \rfloor \neq 0\}$
            Define $\texttt{Problem}_{s+1}$ as $\texttt{Problem}_{\overline{s}}$ with additional constraint $x_{\overline{i}} \leq \lfloor \widetilde{x}_{\overline{i}} \rfloor$
            Define $\texttt{Problem}_{s+2}$ as $\texttt{Problem}_{\overline{s}}$ with additional constraint $x_{\overline{i}} \geq \lceil \widetilde{x}_{\overline{i}} \rceil$
            $S \leftarrow S \cup \{s+1, s+2\}$
            $s \leftarrow s + 2$
        **else if** $\exists i : \widetilde{w}_i - \lfloor \widetilde{w}_i \rfloor \neq 0$ **then**
            Define $\overline{i} := \min\{i : \widetilde{w}_i - \lfloor \widetilde{w}_i \rfloor \neq 0\}$
            Define $\texttt{Problem}_{s+1}$ as $\texttt{Problem}_{\overline{s}}$ with additional constraint $w_{\overline{i}} \leq \lfloor \widetilde{w}_{\overline{i}} \rfloor$
            Define $\texttt{Problem}_{s+2}$ as $\texttt{Problem}_{\overline{s}}$ with additional constraint $w_{\overline{i}} \geq \lceil \widetilde{w}_{\overline{i}} \rceil$
            $S \leftarrow S \cup \{s+1, s+2\}$
            $s \leftarrow s + 2$
        **else**
            `optval`$\leftarrow \widetilde{\texttt{obj}}$
            `sol`$\leftarrow (\widetilde{q}, \widetilde{x}, \widetilde{w})$
        **end if**
    **end if**
    $S \leftarrow S \backslash \{\overline{s}\}$
**end while**
`optval` is optimal objective function value and `sol` is optimal solution

**Algorithm C.2** Branch-and-cut algorithm with MIR for CVaR model

Initialize **MP-CVaR** with no cuts
Initialize `sol`:=`null`
Initialize `optval`:= 0
**for** $k \in K$ **do**
    Initialize $\texttt{cutlist}^k$
    Add cut $q^k \geq 0$ to $\texttt{cutlist}^k$
    Add cut $q^k \geq 0$ to $L^k_{\text{CVaR}} \geq 0$
**end for**
Initialize $S := \{0\}$
Initialize $s := 0$
Define $\texttt{Problem}_0$ as **MP-CVaR**
**while** $S \neq \emptyset$ **do**
    Define $\overline{s} := \min\{s : s \in S\}$
    **repeat**
        Solve $\texttt{Problem}_{\overline{s}}$ to obtain optimal solution $(\widetilde{q}, \widetilde{w}, \widetilde{x}, \widetilde{v})$ and objective $\widetilde{\texttt{obj}}$
        **for** $k \in K$ **do**
            Solve **SP-CVaR**$_k(\widetilde{w}, \widetilde{x}, \widetilde{v})$ to obtain optimal dual solution $(\widehat{\pi}, \widehat{\lambda}, \widehat{\theta})$ and optimal objective $\widehat{q}^k$
            **if** $\widehat{q}^k > \widetilde{q}^k$ **then**
                Define `cutA` as cut (4.13)
                Add `cutA` to $\texttt{cutlist}^k$
                **for** `cutB` $\in \texttt{cutlist}^k$ **do**
                    $\Delta \leftarrow 1$
                    Apply Theorem 4.4 to `cutA` and `cutB` to generate $\texttt{cutC}_0$
                    Add $\texttt{cutC}_0$ to $\texttt{cutlist}^k$
                    **for** $i \in I$ **do**
                        $\Delta \leftarrow \frac{1}{|\alpha_i^{\texttt{B}} - \alpha_i^{\texttt{A}}|}$, where $\alpha^{\texttt{A}}$ and $\alpha^{\texttt{B}}$ are coefficients of $w$ in `cutA` and `cutB` respectively
                        Apply Theorem 4.4 to `cutA` and `cutB` to generate $\texttt{cutC}_i$
                        Add $\texttt{cutC}_i$ to $\texttt{cutlist}^k$
                        $\Delta \leftarrow \frac{1}{|\beta_i^{\texttt{B}} - \beta_i^{\texttt{A}}|}$, where $\beta^{\texttt{A}}$ and $\beta^{\texttt{B}}$ are coefficients of $x$ in `cutA` and `cutB` respectively
                        Apply Theorem 4.4 to `cutA` and `cutB` to generate $\texttt{cutD}_i$
                        Add $\texttt{cutD}_i$ to $\texttt{cutlist}^k$
                    **end for**
                **end for**
                Among `cutA` and $\texttt{cutC}_i$ and $\texttt{cutD}_i$ $\forall i \in I$, add to $L^k_{\text{CVaR}} \geq 0$ the cut with the smallest

$$\frac{\delta - (\widetilde{q}^k + \sum_{i \in I} \alpha_i \widetilde{w}_i + \sum_{i \in I} \beta_i \widetilde{x}_i + \gamma \widetilde{v})}{1 + \sum_{i \in I} (\alpha_i)^2 + \sum_{i \in I} (\beta_i)^2 + \gamma^2}$$

            **end if**
        **end for**
    **until** No cut is added
    **if** $\widetilde{\texttt{obj}} <$ `optval` **then**
        **if** $\exists i : \widetilde{x}_i - \lfloor \widetilde{x}_i \rfloor \neq 0$ **then**
            Define $\overline{i} := \min\{i : \widetilde{x}_i - \lfloor \widetilde{x}_i \rfloor \neq 0\}$
            Define $\texttt{Problem}_{s+1}$ as $\texttt{Problem}_{\overline{s}}$ with additional constraint $x_{\overline{i}} \leq \lfloor \widetilde{x}_{\overline{i}} \rfloor$
            Define $\texttt{Problem}_{s+2}$ as $\texttt{Problem}_{\overline{s}}$ with additional constraint $x_{\overline{i}} \geq \lceil \widetilde{x}_{\overline{i}} \rceil$
            $S \leftarrow S \cup \{s+1, s+2\}$
            $s \leftarrow s + 2$
        **else if** $\exists i : \widetilde{w}_i - \lfloor \widetilde{w}_i \rfloor \neq 0$ **then**
            Define $\overline{i} := \min\{i : \widetilde{w}_i - \lfloor \widetilde{w}_i \rfloor \neq 0\}$
            Define $\texttt{Problem}_{s+1}$ as $\texttt{Problem}_{\overline{s}}$ with additional constraint $w_{\overline{i}} \leq \lfloor \widetilde{w}_{\overline{i}} \rfloor$
            Define $\texttt{Problem}_{s+2}$ as $\texttt{Problem}_{\overline{s}}$ with additional constraint $w_{\overline{i}} \geq \lceil \widetilde{w}_{\overline{i}} \rceil$
            $S \leftarrow S \cup \{s+1, s+2\}$
            $s \leftarrow s + 2$
        **else**
            `optval`$\leftarrow \widetilde{\texttt{obj}}$
            `sol`$\leftarrow (\widetilde{q}, \widetilde{w}, \widetilde{x}, \widetilde{v})$
        **end if**
    **end if**
    $S \leftarrow S \backslash \{\overline{s}\}$
**end while**
`optval` is optimal objective function value and `sol` is optimal solution

# BIBLIOGRAPHY


Ahuja, R. K., Magnanti, T. L., Orlin, J. B., 1993. Network Flows: Theory, Algorithms, and Application. Prentice Hall.

Ben-Tal, A., Nemirovski, A., 1998. Robust convex optimization. Mathematics of Operations Research 23 (4), 769–805.

Benders, J. F., 1962. Partitioning procedures for solving mixed-variables programming problems. Numerische Mathematik 4 (1), 238–252.

Bertsimas, D., Sim, M., 2003. Robust discrete optimization and network flows. Mathematical Programming 98 (1), 49–71.

Bertsimas, D., Sim, M., 2004. The price of robustness. Operations Research 52 (1), 35–53.

Bodur, M., Luedtke, J. R., 2014. Mixed-integer rounding enhanced Benders decomposition for multiclass service system staffing and scheduling with arrival rate uncertainty. Available at Optimization-Online http://www.optimization-online.org/DB_FILE/2013/10/4080.pdf.

Calafiore, G., El Ghaoui, L., 2006. On distributionally robust chance-constrained linear programs. Journal of Optimization Theory and Applications 130 (1), 1–22.

Carsharing, 2009. One thousand world cities where you can carshare this morning. http://ecoplan.org/carshare/general/cities.htm.

Chang, J., Shen, S., Xu, M., 2016. Carsharing fleet location design with mixed vehicle types for carbon emissions reduction. Working paper.

Chang, M.-S., Tseng, Y.-L., Chen, J.-W., 2007. A scenario planning approach for the flood emergency logistics preparation problem under uncertainty. Transportation Research Part E 43 (6), 737–754.

Charnes, A., Cooper, W., Symonds, G., 1958. Cost horizons and certainty equivalents: An approach to stochastic programming of heating oil. Management Science 4 (3), 235–263.

Chen, A., Kim, J., Zhou, Z., Chootinan, P., 2007. Alpha reliable network design problem. Transportation Research Record 2029, 49–57.

Chen, A., Yang, C., 2004. Stochastic transportation network design problem with spatial equity constraint. Transportation Research Record 1882, 97–104.

Chen, A., Zhou, Z., Chootinan, P., Ryu, S., Yang, C., Wong, S., 2011. Transport network design problem under uncertainty: A review and new developments. Transport Reviews 31 (6), 743–768.

Chesto, J., 2015. Zipcar seeks 150 park-anywhere permits. The Boston Globe.

Crainic, T. G., Fu, X., Gendreau, M., Rei, W., Wallace, S. W., 2011. Progressive hedging-based metaheuristics for stochastic network design. Networks 58 (2), 114–124.

Daskin, M. S., 2011. Network and Discrete Location: Models, Algorithms, and Applications. John Wiley & Sons.

de Almeida Correia, G. H., Antunes, A. P., 2012. Optimization approach to depot location and trip selection in one-way carsharing systems. Transportation Research Part E: Logistics and Transportation Review 48 (1), 233–247.

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

Fan, W. D., 2014. Optimizing strategic allocation of vehicles for one-way car-sharing systems under demand uncertainty. Journal of the Transportation Research Forum 53 (3), 7–20.

Fan, W. D., Machemehl, R. B., Lownes, N. E., 2008. Carsharing: Dynamic decision-making problem for vehicle allocation. Transportation Research Record: Journal of the Transportation Research Board 2063 (1), 97–104.

Febbraro, A., Sacco, N., Saeednia, M., 2012. One-way carsharing: Solving the relocation problem. Transportation Research Record: Journal of the Transportation Research Board (2319), 113–120.

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

He, L., Mak, H. Y., Rong, Y., Shen, Z. J. M., 2015. Service region design for urban electric vehicle sharing systems. Working paper.

Katzev, R., 2003. Car sharing: A new approach to urban transportation problems. Analyses of Social Issues and Public Policy 3 (1), 65–86.

Kek, A. G. H., Cheu, R. L., Meng, Q., Fung, C. H., 2009. A decision support system for vehicle relocation operations in carsharing systems. Transportation Research Part E: Logistics and Transportation Review 45 (1), 149–158.

Kleywegt, A. J., Shapiro, A., Homem-de Mello, T., 2002. The sample average approximation method for stochastic discrete optimization. SIAM Journal on Optimization 12 (2), 479–502.

Laporte, G., Louveaux, F. V., 1993. The integer L-shaped method for stochastic integer programs with complete recourse. Operations Research Letters 13 (3), 133–142.

LeBlanc, L. J., Morlok, E. K., Pierskalla, W. P., 1975. An efficient approach to solving the road network equilibrium traffic assignment problem. Transportation Research 9 (5), 309–318.

Lium, A.-G., Crainic, T. G., Wallace, S. W., 2009. A study of demand stochasticity in service network design. Transportation Science 43 (2), 144–157.

Lo, H. K., Tung, Y.-K., 2003. Network with degradable links: Capacity analysis and design. Transportation Research Part B 37 (4), 345–363.

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

McCormick, G. P., 1976. Computability of global solutions to factorable nonconvex programs: Part I – Convex underestimating problems. Mathematical Programming 10 (1), 147–175.

MirHassani, S., Lucas, C., Mitra, G., Messina, E., Poojari, C., 2000. Computational solution of capacity planning models under uncertainty. Parallel Computing 26 (5), 511–538.

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

Nemhauser, G. L., Wolsey, L. A., 1988. Integer and Combinatorial Optimization. Wiley, New York.

Oh, S. C., Haghani, A., 1997. Testing and evaluation of a multi-commodity multi-modal network flow model for disaster relief management. Journal of Advanced Transportation 31 (3), 249–282.

Orlowski, S., Wessäly, R., Pióro, M., Tomaszewski, A., 2010. SNDlib 1.0 – Survivable Network Design Library. Networks 55 (3), 276–286.

Patil, G., Ukkusuri, S., 2007. System-optimal stochastic transportation network design. Transportation Research Record: Journal of the Transportation Research Board 2029, 80–86.

Pfrommer, J., Warrington, J., Schildbach, G., Morari, M., 2014. Dynamic vehicle redistribution and online price incentives in shared mobility systems. Intelligent Transportation Systems, IEEE Transactions on 15 (4), 1567–1578.

Pióro, M., Medhi, D., 2004. Routing, Flow, and Capacity Design in Communication and Computer Networks. Elsevier.

Pishvaee, M. S., Rabbani, M., Torabi, S. A., 2011. A robust optimization approach to closed-loop supply chain network design under uncertainty. Applied Mathematical Modelling 35 (2), 637–649.

Rockafellar, R. T., Uryasev, S., 2000. Optimization of conditional value-at-risk. Journal of Risk 2, 21–42.

Ruszczyński, A., 2002. Probabilistic programming with discrete distributions and precedence constrained knapsack polyhedra. Mathematical Programming 93 (2), 195–215.

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

Shaheen, S., Cohen, A., 2007. Growth in worldwide carsharing: An international comparison. Transportation Research Record: Journal of the Transportation Research Board 1992 (1), 81–89.

Shaheen, S., Sperling, D., Wagner, C., 1998. Carsharing in Europe and North America: Past, present, and future. Transportation Quarterly 52 (3), 35–52.

Shapiro, A., Homem-de-Mello, T., 2000. On the rate of convergence of optimal solutions of Monte Carlo approximations of stochastic programs. SIAM Journal of Optimization 11 (1), 70–86.

Shen, S., Chen, Z., 2013. Optimization models for differentiating quality of service levels in probabilistic network capacity design problems. Transportation Research Part B: Methodological 58, 71–91.

Sheu, J., 2007. Challenges of emergency logistics management. Transportation Research Part E 43 (6).

Social Security Administration, 2015. National average wage index. https://www.ssa.gov/oact/cola/AWI.html.

Soyster, A. L., 1973. Convex programming with set-inclusive constraints and applications to inexact linear programming. Operations Research 21 (5), 1154–1157.

Tsiakis, P., Shah, N., Pantelides, C. C., 2001. Design of multi-echelon supply chain networks under demand uncertainty. Industrial & Engineering Chemistry Research 40 (16), 3585–3604.

Ukkusuri, S. V., Mathew, T. V., Waller, S. T., 2007. Robust transportation network design under demand uncertainty. Computer-Aided Civil and Infrastructure Engineering 22 (1), 6–18.

Ukkusuri, S. V., Patil, G., 2009. Multi-period transportation network design under demand uncertainty. Transportation Research Part B: Methodological 43 (6), 625–642.

Van Slyke, R. M., Wets, R., 1969. L-shaped linear programs with applications to optimal control and stochastic programming. SIAM Journal on Applied Mathematics 17 (4), 638–663.

Waller, S. T., Ziliaskopoulos, A. K., 2001. Stochastic dynamic network design problem. Transportation Research Record 1771, 106–113.

Weikl, S., Bogenberger, K., 2013. Relocation strategies and algorithms for free-floating car sharing systems. Intelligent Transportation Systems Magazine, IEEE 5 (4), 100–111.

Wolsey, L. A., 1998. Integer Programming. Wiley, New York, NY.

Yang, H., H. Bell, M. G., 1998. Models and algorithms for road network design: A review and some new developments. Transport Reviews 18 (3), 257–278.

Zhou, T., June 2015. Network design for integrated vehicle-sharing and public transportation service. Master's thesis, Massachusetts Institute of Technology.

Zipcar, 2015. Car sharing: An alternative to car rental with Zipcar. http://www.zipcar.com/.