UNIVERSITY OF CALIFORNIA,
IRVINE

Accelerating Benders Decomposition: Theory and Applications

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Management

by

Seyed Mojtaba Hosseini

Dissertation Committee:
Associate Professor John G. Turner, Chair
Professor Rick (Kut) So
Associate Professor Luyi Gui

2022

DEDICATION

To my loving wife and best friend, Arezou.

TABLE OF CONTENTS

LIST OF FIGURES

Page

LIST OF TABLES

LIST OF ALGORITHMS

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest appreciation to my advisor, Dr. John Turner. He has been a source of inspiration for me from the very first day at UCI, and a tremendous mentor and a true friend throughout my career. I am deeply indebted to his continuous support, encouragement, and invaluable advice on my research, career and life. This thesis would not have been possible without his exemplary mentorship and guidance throughout this journey.

I am specially thankful to Dr. Vijay Vazirani, for his support and giving me the opportunity to collaborate with him on matching markets. I am also grateful to the Operations and Decision Technologies faculty: Dr. Carlton Scott, Dr. Rick So, Dr. Robin Keller, Dr. Shuya Yin, Dr. Luyi Gui, and Dr. Kenneth Murphy, from whom I have learned a lot. I am also thankful to Noel Negrete for her valued service to the PhD program.

Special thanks goes to the co-authors of the appended papers Dr. Gita Taherkhani and Dr. Sibel A. Alumur for the knowledge exchange and cooperation we had in preparing the papers. I am also grateful to Dr. Christiane Barz for her support and the insightful discussions we had during my time in Switzerland.

I would like to take the opportunity to thank all my colleagues at the ODT group, particularly my lifelong friends Ali Hassanzadeh and Ali Esmaeeli and their families for their friendship, memories, and fruitful discussions.

Last but not least, I would like to give my deepest gratitude to my wife, Arezou, for her endless love and caring, and my parents for their spiritual support.

VITA

Seyed Mojtaba Hosseini

EDUCATION

Doctor of Philosophy in Management **2022**
University of California Irvine *Irvine, California*

Master of Science in Industrial Engineering **2016**
Koç University *Istanbul, Turkey*

Bachelor of Science in Industrial Engineering **2013**
Sharif University of Technology *Tehran, Iran*

RESEARCH EXPERIENCE

Graduate Research Assistant **2016–2022**
University of California, Irvine *Irvine, California*

Graduate Research Assistant **2014–2016**
Koç University *Istanbul, Turkey*

TEACHING EXPERIENCE

Teaching Assistant **2016–2022**
University of California, Irvine *Irvine, California*

Teaching Assistant **2014–2016**
Koç University *Istanbul, Turkey*

REFEREED JOURNAL PUBLICATIONS

- "Robust-Stochastic Models for Profit Maximizing Hub Location Problems" (2021), *Transportation Science*, (with Gita Taherkhani and Sibel A. Alumur).
- "Benders Decomposition for Profit Maximizing Hub Location Problems with Multiple Demand Classes" (2020), *Transportation Science*, (with Gita Taherkhani and Sibel A. Alumur).

REFEREED CONFERENCE PUBLICATIONS

- "Nash-Bargaining-Based Models for Matching Markets: One-Sided and Two-Sided; Fisher and Arrow-Debreu" (2022), in *The 13th Innovations in Theoretical Computer Science Conference (ITCS 2022)* proceedings, (with Vijay V. Vazirani)

ABSTRACT OF THE DISSERTATION

Accelerating Benders Decomposition: Theory and Applications

By

Seyed Mojtaba Hosseini

Doctor of Philosophy in Management

University of California, Irvine, 2022

Associate Professor John G. Turner, Chair

Since its inception, Benders Decomposition (BD) has been successfully applied to a wide range of large-scale mixed-integer (linear) problems that lie at the heart of operations research and supply chain management. The inherent capacity of BD for exploiting the structural properties of problems with complicating variables has made it one of the most prominent exact algorithms for solving large-scale optimization problems. Over the years, BD has grown in its ability to solve a wide range of challenging problems including variants of facility location problems, supply chain and network design problems, scheduling and routing problems, healthcare operations, machine learning, and variants of stochastic programming problems among several other applications. This dissertation is structured into three research papers. First, we introduce a general acceleration technique for BD by introducing deepest Benders cuts. As an application of BD on problems arising in the transportation and logistics sector, we introduce efficient and novel implementations of BD for variants of hub location problems. We further analyze effects of uncertainty in demand and revenues in hub location problems and propose novel modelling and solution methods based on robust and stochastic optimization techniques.

The key element of BD is the derivation of Benders cuts, which are often not unique. In the first chapter, we introduce a novel unifying Benders cut selection technique based on a geometric interpretation of cut "depth", produce deepest Benders cuts based on ℓ_p-norms, and study their properties. Specifically, we show that deepest cuts resolve infeasibility through minimal deviation from the incumbent point, are relatively sparse, and may produce optimality cuts even when classical Benders would require a feasibility cut. Leveraging the duality between separation and projection, we develop a Guided Projections Algorithm for producing deepest cuts while exploiting the combinatorial structure or decomposablity of problem instances. We then propose a generalization of our Benders separation problem that brings several well-known cut selection strategies under one umbrella. In particular, by establishing its connection to our method, we provide systematic ways of selecting the normalization coefficients in the Minimal Infeasible Subsystems method. We also provide general implementation guidelines, which are useful beyond the scope of this study. Finally, in our tests on facility location problems, we show deepest cuts often reduce both runtime and number of Benders iterations, as compared to other cut selection strategies; and relative to classical Benders, use 1/3 the number of cuts and 1/2 the runtime.

As an application of accelerating Benders decomposition, we model capacity allocation decisions within profit maximizing hub location problems to satisfy demand of commodities from different market segments. We present a strong deterministic formulation of the problem and describe two exact algorithms based on a Benders reformulation to solve large-size instances of the problem. We show that the subproblems can be broken into smaller and simpler problems in two phases. We prove that the first phase can be efficiently solved using a cutting-plane algorithm. To produce strong cuts, we cast the second phase as a multi-objective optimization problem. We show that non-dominated solutions to the second phase can be obtained by either a set of continuous maximum weighted matching problems, or by a series of continuous knapsack problems in a sequential manner. We further enhance the performance of the algorithms by integrating improved variable fixing techniques. We evaluate the efficiency and robustness of the algorithms through extensive computational experiments. Computational results show that large-scale instances with up to 500 nodes and three demand segments can be solved to optimality, and that the proposed algorithms generate cuts that provide significant speedups compared to using Pareto-optimal cuts. The proposed two-phase methodology for solving the Benders subproblem as well as the variable fixing and acceleration techniques can be used to solve other discrete location and network design problems.

Finally, we extend the deterministic model by considering uncertainty associated with the demand to develop a two-stage stochastic program. To solve the stochastic version, we develop a Monte-Carlo simulation-based algorithm that integrates a sample average approximation scheme with the proposed Benders decomposition algorithms. We then extend the models by simultaneously incorporating two sources of uncertainty including stochastic demand and uncertain revenue. To incorporate uncertain revenues into the problem, we use robust optimization techniques and investigate two particular cases including interval uncertainty and discrete scenarios. We formulate robust-stochastic models with a max-min profit criterion and a min-max regret objective for the former and latter cases, respectively. We enhance the Benders decomposition algorithms coupled with sample average approximation scheme by novel acceleration techniques. We also conduct extensive computational experiments to analyze the effects of uncertainty under different settings and compare the quality of the solutions obtained from different modeling approaches under different parameter settings. Computational results demonstrate the efficiency of the proposed algorithms and justify the need for embedding both sources of uncertainty in decision making to provide robust solutions.

Chapter 1

Introduction

1.1 Benders Decomposition

Since Benders (1962) originally proposed a procedure for solving Mixed-Integer Linear Programming (MILP) problems that temporarily fixes some variables to produce one or more much easier-to-solve subproblems at the expense of additional inference and algorithm iterations, Benders Decomposition (BD) has increasingly attracted the attention of researchers in the last five decades. Of note, BD has proven very effective in tackling several classes of challenging MILP problems through both the classical as well as the generalized and logic-based variants of the BD algorithm.

The inherent capacity of BD for exploiting the structural properties of problems with complicating variables has made it one of the most prominent exact algorithms for solving large-scale optimization problems. Over the years, BD has grown in its ability to solve a wide range of challenging problems including variants of facility location problems (Magnanti and Wong 1981, Fischetti et al. 2016, 2017), supply chain and network design problems (Keyvanshokooh et al. 2016, Alshamsi and Diabat 2018, Fontaine and Minner 2018, Pearce and Forbes

2018), hub location problems (Contreras et al. 2011a, 2012, Maheo et al. 2017, Taherkhani et al. 2020), scheduling and routing problems (Mercier 2008, Papadakos 2009, Adulyasak et al. 2015, Bodur and Luedtke 2016, Bayram and Yaman 2017), healthcare operations (Cho et al. 2014, Naderi et al. 2021), machine learning (Rahimi and Gönen 2021), and variants of stochastic programming problems (Santoso et al. 2005, Adulyasak et al. 2015, Bodur et al. 2016, Rahmaniani et al. 2018, Khassiba et al. 2020, Taherkhani et al. 2021) among several other applications.

BD, at its core, is a relax and "learn from mistakes" procedure (Hooker and Ottosson 2003). In classical BD, this learning mechanism is naturally manifested through Linear Programming (LP) duality and mistakes are "corrected" via Benders feasibility and optimality cuts. These cuts are obtained by solving the dual of the subproblem induced by fixing the complicating variables. The learning mechanism, however, need not be restricted to cuts based on LP duality. Geoffrion (1972) laid the foundation for extending BD to general nonlinear optimization problems, Hooker and Ottosson (2003) introduced logic-based BD for tackling problems with logical constraints, and Codato and Fischetti (2006) tailored this idea to MILP problems involving big-M constraints.

Despite its promising structure, a naïve implementation of BD may suffer from slow convergence and other computational deficiencies. A wealth of studies have addressed different drawbacks of BD from different angles (see e.g., Rahmaniani et al. 2017, and references therein for recent advancements). As with any other cutting-plane algorithm, the convergence rate is directly tied to the effectiveness of the generated cuts. Given that there is typically more than one way to generate a Benders cut, an important theoretical and practical question is how to select the most "effective" cut(s) in each iteration, with the aim of speeding up convergence. This question has spawned a stream of research, which we contribute to.

In their seminal paper, Magnanti and Wong (1981) introduced a general-purpose cut selection strategy for selecting a nondominated (or Pareto-optimal) optimality cut among the alternative optimal solutions of the subproblem. More recently, Fischetti et al. (2010) cast the Benders subproblem as a feasibility problem, and proposed an alternative cut selection criterion that approximately identifies a minimal source of infeasibility from the derived feasibility problem. Saharidis and Ierapetritou (2010) introduced the Maximum Feasibility Subsystem (MFS) cut generation strategy for accelerating BD in problems where the majority of cuts generated are feasibility (as opposed to optimality) cuts. Sherali and Lunday (2013) treated cut generation as a multi-objective optimization problem and proposed generating maximal nondominated cuts which they showed can be produced by perturbing the right-hand-side of the primal subproblem. Recently, Bodur and Luedtke (2016) and Bodur et al. (2016) proposed methods for sharpening Benders cuts using mixed-integer rounding schemes.

1.2 Classical Benders Decomposition

We begin with a brief outline of the classical Benders Decomposition (BD) algorithm. Consider the MILP problem

$$\begin{aligned} \text{[OP]}\ \min \quad & c^\top x + f^\top y \\ \text{s.t.} \quad & Ax + By \geq b \\ & x \geq 0, y \in Y, \end{aligned} \tag{1.1}$$

where $f \in \mathbb{R}^n$, $c \in \mathbb{R}^{n'}$, $b \in \mathbb{R}^m$, matrices A and B are conformable, and $Y \subset \mathbb{Z}^n$ is the domain of the y-variables. In what follows, we reserve i and j for indexing the rows and columns of B, respectively. For the sake of generality, we do not make any specific assumptions about the structure of the problem, except that it is a general bounded MILP.

The idea behind BD is to project the original problem (OP) from the space of the (x, y)-variables onto the space of the y-variables in the form of

$$\min\{Q(y) : y \in Y \cap \mathrm{dom}(Q)\}, \tag{1.2}$$

where $Q(y) = f^\top y + \tilde{Q}(y)$ and $\tilde{Q}(y)$ accounts for the contribution of the x-variables to the objective function and is defined as

$$[\text{PSP}]\ \tilde{Q}(y) = \min\ \{c^\top x : Ax \geq b - By, x \geq 0\}. \tag{1.3}$$

Problem (1.3) is known as the primal subproblem (PSP) and $\mathrm{dom}(Q)$ is the set of y values that induce a feasible PSP. Since OP is bounded, PSP is also bounded for any $y \in Y$. The classical BD algorithm works as follows. First, problem (1.2) is reformulated in epigraph form as

$$\min\ \{\eta : (y, \eta) \in \mathcal{E}, y \in Y\}, \tag{1.4}$$

where $\mathcal{E}$ is the epigraph of Q defined as

$$\mathcal{E} = \{(y, \eta) \in \mathbb{R}^{n+1} : \eta \geq Q(y), y \in \mathrm{dom}(Q)\}.$$

Then, a relaxation of (1.4) is successively tightened by progressively outer-approximating $\mathcal{E}$ with supporting hyperplanes obtained by evaluating, at given y values, the dual of (1.3) formulated as

$$[\text{DSP}]\ \tilde{Q}(y) = \max\ \{u^\top(b - By) : u^\top A \leq c^\top, u \geq 0\}, \tag{1.5}$$

which is known as the dual subproblem (DSP). From this dual formulation, we can observe that $\tilde{Q}(y)$ is a piece-wise linear convex function of y. Thus, $Q(y) = f^\top y + \tilde{Q}(y)$ is a piece-wise linear convex function and $\mathcal{E}$ is a closed convex set. Let $\mathcal{U}$ denote the polyhedron defining the set of feasible solutions of DSP, with $\mathcal{U}^*$ its extreme points. For $y \in \mathrm{dom}(Q)$, the DSP induced by y is bounded and its optimal value is attained at one of the extreme points of $\mathcal{U}$. Additionally, since $\tilde{Q}(y)$ is the optimal value of DSP, it follows from weak duality that

$$Q(y) = f^\top y + \tilde{Q}(y) \geq f^\top y + \hat{u}^\top (b - By) \qquad \forall \hat{u} \in \mathcal{U}.$$

On the other hand, by Farkas lemma, the values of y that induce an infeasible PSP (i.e., an unbounded DSP) are the ones for which $\hat{v}^\top (b - By) > 0$ for some (extreme) ray $\hat{v}$ of $\mathcal{U}$. Hence, $\mathrm{dom}(Q)$ may be defined as $\mathrm{dom}(Q) = \{y : 0 \geq \hat{v}^\top (b - By) \;\; \forall \hat{v} \in \mathcal{V}^*\}$, where $\mathcal{V}^*$ is the set of extreme rays of $\mathcal{U}$. Putting these pieces together, we can rewrite (1.4) as

$$\begin{aligned}
\text{[CMP]} \;\; \min \quad & \eta && && (1.6)\\
\text{s.t.} \quad & \eta \geq f^\top y + \hat{u}^\top (b - By) && \forall \hat{u} \in \mathcal{U}^* && (1.7)\\
& 0 \geq \hat{v}^\top (b - By) && \forall \hat{v} \in \mathcal{V}^* && (1.8)\\
& \eta \in \mathbb{R}, y \in Y, && && (1.9)
\end{aligned}$$

which we refer to as the classical Benders master problem (CMP). Constraint sets (1.7) and (1.8) are known as the *Benders optimality* and *feasibility* cuts, respectively. The classical BD algorithm solves CMP by initially relaxing these constraints, and at each iteration posts one or more cuts of the form (1.7) or (1.8) to this relaxation of CMP until the optimality gap is sufficiently closed. For a complete description of the BD algorithm the reader is referred to Rahmaniani et al. (2017).

1.3 Hub Location Problems

In transportation, telecommunication and computer networks, hub-and-spoke structures are used to collect, sort, switch or consolidate different flows of commodities. Hubs are special types of facilities that connect a large number of origin-destination (O-D) pairs via a small number of links. Hubs consolidate flows of different O-D pairs and by doing so reduce the transportation costs by exploiting economies of scale. A hub location problem (HLP) is a network design problem and generally consists of two main decision levels: selection of a set of nodes to locate the hubs and to allocate the demand nodes to these hubs (Campbell et al. 2002, Alumur and Kara 2008). HLPs are a difficult class of NP-hard problems, where the interrelated decision process is the main source of difficulty in HLPs (Contreras et al. 2011a).

In classical hub location problems, it is usually required that the demand of each commodity must be fully satisfied and so are formulated as cost minimization problems. It may, however, be more advantageous from a revenue management perspective not to fully serve the demand of some commodities, especially if the cost of serving a commodity is higher than the revenue associated with satisfying its demand. In such a setting, the right objective is to maximize profit rather than minimizing cost so that the decision on how much demand of each commodity to satisfy depends on the trade-off between revenue and cost. Moreover, the demand of commodities usually consists of different classes (e.g., regular and priority service). The decision maker thus needs to consider how to allocate the available capacity to these different demand segments, while determining the proportion of the demand to serve for each class.

The problem under study is motivated from airline passenger and freight transportation, and express shipment delivery networks in which the amount of demand of different commodities to serve is dependent on hub locations. Even though passengers choose their own routes in airline networks, airlines still have to design their hub networks considering the forecasted demand. Different classes of demand may include the demand for, for example, the first, business, and economy class service. In express shipment delivery networks, on the other hand, there is a demand for services such as priority, express, and standard mail.

1.3.1 Related Works

In the last few decades, many variants of hub location problems have been studied in the literature. The reader may refer to reviews on this area by Campbell et al. (2002), Alumur and Kara (2008), and Contreras (2015). Capacitated versions of hub location problems are first formulated by Campbell (1994) using path-based mixed integer programs that impose capacity constraints on the total incoming flow at hubs (i.e., flow arriving from both hub and non-hub nodes). A variant of this problem arises when capacities are applied only to the traffic arriving directly from non-hub nodes. This variant is motivated from the postal-delivery applications and has been studied by Boland et al. (2004) and Contreras et al. (2012) among others. In some postal-delivery and express shipment networks (e.g. UPS, Canada Post), however, total incoming flow (regardless that it is from a hub or a non-hub node) is sorted at every hub-stop. Moreover, the limiting capacity of a hub may not necessarily be on sorting or material handling, but, for example, on the available number of docks or gates. Hence, in this study, we model the generic case and impose capacity constraints on the total incoming flow at hubs, as introduced in Campbell (1994). As we elaborate in §3.2.2, this generic definition of capacity usage brings on extra challenges for the implementation of Benders decomposition.

There are only a few studies considering a profit-oriented objective in hub location problems. Alibeyg et al. (2016) and the companion paper Alibeyg et al. (2018) present hub network design problems with profits including the decisions on the O-D pairs that will be served.

They employ Lagrangean relaxation within a branch-and-bound algorithm, and also adopt reduction tests and partial enumeration to reduce the size of the problems as well as the computation times. Lin and Lee (2018) consider a hub network design problem for time definite LTL freight transportation in which the carrier aims to determine hub locations, under price elasticity of demand, that maximize total profit. They show that profit optimization builds a denser hub network than cost minimization. In the present study, we additionally consider demands of commodities from different market segments and incorporate capacity allocation decisions.

From a methodological point of view, Benders decomposition (BD) has received increased attention, particularly for solving multiple allocation hub location problems. Camargo et al. (2008) is the first work using a Benders reformulation to solve the uncapacitated multiple allocation hub location problem. Rodriguez-Martin and Salazar-Gonzalez (2008) consider a capacitated multiple allocation hub location problem in which the arcs connecting the hubs are not assumed to create a complete graph. They provide a formulation and design two exact solution algorithms relying on BD. Contreras et al. (2011a) employ a Benders reformulation for the uncapacitated multiple allocation hub location problem which is enhanced through the use of a multicut reformulation, the generation of (approximated) Pareto-optimal cuts, the integration of reduction tests, and the execution of a heuristic procedure as a warm start point of their algorithms. Contreras et al. (2012) provide an extension of the BD approach proposed by Contreras et al. (2011a) to solve capacitated multiple allocation hub location problems.

1.4 Hub Location Problems under Uncertainty

In strategic planning, decisions need to be held for a considerable time frame. During this time, in real world, many unpredictable causes may lead to changes in operating conditions.

For example, the amount of demand may be greater or less than its expected value. Changes may also occur in the amount of revenue obtained from the satisfied demand due to some unpredictable variations in a competitive environment. In these conditions, solving a deterministic model may result in wrong and costly decisions. Hence, taking uncertainty into account in the decision process is a necessity. To provide more reliable models, we consider two typical sources of uncertainty in our problem. We assume that demand of commodities and revenues are not precisely known and the optimal decisions have to be anticipated under uncertainty.

Optimization under uncertainty generally consists of two streams of research: stochastic and robust optimization. In stochastic optimization, there are some known probability distributions describing the behavior of uncertain parameters and these distributions can be used to optimize the expected value of the objective function. In robust optimization, on the other hand, no probabilistic information is available for the uncertain parameters. In this case, uncertainty can be described by using a finite set of scenarios or can be modeled assuming that the values of the uncertain parameters can change within predefined intervals (for more information on robust optimization see, e.g., Bertsimas and Sim 2003, Ben-Tal et al. 2004, Bertsimas et al. 2011, and Correia and Saldanha-da Gama 2015).

1.4.1 Uncertainty Considerations in Hub Location Problems

Yang (2009) proposes a two-stage stochastic programming model to address the issues of air freight hub location and flight-route planning under stochastic demand. They solve the deterministic equivalent of this problem considering three scenarios. In the same year, Sim et al. (2009) present the stochastic p-hub center problem, in which service level considerations are incorporated by using chance constraints when travel times are normally distributed. Contreras et al. (2011b) study the uncapacitated multiple allocation hub location problem under uncertain demands and transportation costs. When demand is analyzed as the source of uncertainty, they show that the stochastic problem will be equivalent to its associated deterministic expected value problem where uncertain demand is replaced by its expectation. However, this equivalence does not hold in the capacitated cases. The authors develop an SAA method to solve their stochastic problems and apply it to instances involving up to 50 demand nodes.

More recently, Meraklı and Yaman (2016) model the robust uncapacitated multiple allocation p-hub median problem under polyhedral demand uncertainty with two different uncertainty sets; hose and hybrid. The hose model assumes that the only available information is the upper limit on the total flow adjacent at each node, while the hybrid model imposes lower and upper bounds on each pairwise demand. They adopt a min-max robustness criterion for a cost-minimization objective function and develop two exact algorithms based on Benders decomposition. Meraklı and Yaman (2017) extend this study by incorporating capacity constraints for hubs and devise two different Benders decomposition algorithms capable of solving instances with up to 50 nodes. To the best of our knowledge, Meraklı and Yaman (2017) is the first paper that incorporates robustness into capacitated hubs problems.

From a methodological point of view, Benders decomposition (BD) has received increased attention, particularly for solving multiple allocation hub location problems. Camargo et al. (2008) is the first work using a Benders reformulation to solve the uncapacitated multiple allocation hub location problem. Rodriguez-Martin and Salazar-Gonzalez (2008) consider a capacitated multiple allocation hub location problem in which the arcs connecting the hubs are not assumed to create a complete graph. They provide a formulation and design two exact solution algorithms relying on BD. Contreras et al. (2011a) employ a Benders reformulation for the uncapacitated multiple allocation hub location problem which is enhanced through the use of a multicut reformulation, the generation of (approximated) Pareto-optimal cuts, the integration of reduction tests, and the execution of a heuristic procedure as a warm start point of their algorithms. Contreras et al. (2012) provide an extension of the BD approach proposed by Contreras et al. (2011a) to solve capacitated multiple allocation hub location problems.

1.5 Organization of the Dissertation

In Chapter 2, we introduce a general framework for producing effective cuts for accelerating Benders decomposition. By treating the separation problem as a feasibility problem, we establish a duality between separation and projection, which we use for deriving what we call *deepest Benders cuts*. With this perspective, the learning component in our Benders procedure can be viewed as resolving infeasibility in this feasibility problem through minimal deviation from the incumbent point. Our approach departs from previous studies by (i) taking the "depth" of the candidate cuts explicitly into account, (ii) providing a unifying framework for producing deep optimality and feasibility cuts, and (iii) introducing Benders distance functions that bring several cut selection strategies under one umbrella.

In Chapter 3, we present a strong mixed-integer programming formulation of the profit maximizing capacitated hub location problem. We propose two exact algorithms based on a Benders decomposition of the deterministic formulation. Since the subproblem is nonseparable, solving it can be as challenging as solving the original problem. Moreover, due to degeneracy in the subproblem, straightforward implementation of Benders decomposition suffers from low convergence. Alleviating these deficiencies, in this study, we propose a general methodology for decomposing inseparable subproblems into smaller problems in a two-phase fashion, where optimality and strength of the cuts are guaranteed in Phase I and Phase II, respectively. More specifically, for the profit maximizing capacitated hub location problem, we prove that the second phase can be solved as a set of LP-relaxations of maximum weighted matching problems, or as a series of LP-relaxations of knapsack problems. We enhance these algorithms by incorporating improved variable fixing techniques.

In Chapter 4, we consider two sources of uncertainty in the profit maximizing hub location problems: demand and revenue. Because of the availability of historical data, we assume that demand is described by a known probability distribution. On the other hand, since revenue might be affected by unpredictable external sources (e.g., economical conditions or competition) and historical data may fail to effectively describe such variations, it may not make sense to assume a known probability distribution for the revenue describing its behavior. Hence, we use robust optimization techniques to incorporate uncertain revenues into the problem by considering both interval representation and discrete scenarios. Modeling profit maximizing hub location problems using both robust and stochastic optimization techniques surely brings on extra computational challenges, yet we believe this is a much more realistic problem setting with respect to information availability. We integrate the proposed Benders decomposition algorithms with a sample average approximation (SAA) scheme to solve the stochastic problem with a continuous demand distribution and an infinite number of scenarios. Inspired by the repetitive nature of SAA, we additionally propose novel acceleration techniques to improve the convergence of the algorithms employed for the stochastic problem. Moreover, we perform extensive computational experiments to evaluate the efficiency and robustness of the proposed algorithms and solve large-scale instances of the problem.

Finally, Chapter 5 summarizes the contributions of this thesis.

Chapter 2

Deepest Cuts for Benders Decomposition

In this chapter, we introduce a general framework for producing effective cuts for accelerating Benders decomposition. We begin §2.1.1 with an alternative decomposition scheme, which paves the way for us to formally define what we mean by "deep" Benders cuts. In §2.1, we introduce a procedure to produce a so-called "deepest Benders cut" by taking the Euclidean depth of the candidate cuts as a measure of cut quality. Then we extend the notion of depth using general ℓ_p-norms in §2.1.3 and provide a comprehensive study of the properties of deepest cuts in §2.1.4. In §2.2, we introduce Benders distance functions and establish an important monotonicity property tied to convexity that generalizes geometric distance. In §2.3, we (i) present some useful reformulations of the separation problem we use to generate deep cuts, (ii) introduce distance functions based on linear normalization functions, and (iii) present several ways of deriving effective normalization coefficients for our linear normalization functions, which connect our method to other cut selection strategies by Fischetti et al. (2010), Magnanti and Wong (1981) and Conforti and Wolsey (2019). Next, we introduce our Guided Projections Algorithm (GPA) in §2.4, which leverages combinatorial structure to further accelerate deep cut generation. Then, in §2.5, we run computational experiments on the capacitated facility location problem to test the performance of Benders' cuts generated using several families of distance functions and demonstrate that deepest cuts produced using GPA require 1/3 the number of cuts and 1/2 the runtime of classical Benders cuts. Finally, we summarize our conclusions in §2.6. Technical details and proofs are provided in Appendix A.

2.1 Deepest Benders Cuts

2.1.1 A Unifying Decomposition Scheme

In classical BD, y is the only piece of information passed from the master problem to the subproblems, and η is merely used to obtain a lower bound on OP. But ignoring η is a little like generating cuts with one hand tied behind your back; you can do it if you have to, but you'll get better results if you don't. We instead begin by reformulating the original problem (1.1) in epigraph form as

$$\begin{aligned} \min \quad & \eta \\ \text{s.t.} \quad & \eta \geq c^\top x + f^\top y \\ & Ax + By \geq b \\ & x \geq 0, y \in Y, \end{aligned} \tag{2.1}$$

and apply BD to it by taking (y, η) as the master problem variables. While this reformulation is sometimes prone to numerical instabilities in the primal space (Bonami et al. 2020), if treated carefully, it provides a framework for unifying the classical Benders optimality and feasibility cuts (Fischetti et al. 2010). Taking this viewpoint, the primal subproblem induced by trial solution $(\hat{y}, \hat{\eta})$ is

$$\begin{aligned} \text{[FSP]}\ \min\quad & 0 \\ \text{s.t.}\quad & -c^\top x \geq f^\top \hat{y} - \hat{\eta} \\ & Ax \geq b - B\hat{y} \\ & x \geq 0, \end{aligned} \tag{2.2}$$

which is a feasibility problem for any given $(\hat{y}, \hat{\eta})$, hence we call it the *feasibility subproblem* (FSP). Assigning the dual variable π_0 to the first constraint and the dual vector π to the second set of constraints, a Farkas certificate for infeasibility of FSP can be produced using

$$\text{[CGSP]}\ \max_{(\pi,\pi_0)\in\Pi}\quad \pi^\top(b - B\hat{y}) + \pi_0(f^\top \hat{y} - \hat{\eta}), \tag{2.3}$$

which we refer to as the *certificate generating subproblem (CGSP)*, where

$$\Pi = \{(\pi, \pi_0) : \pi^\top A \leq \pi_0 c^\top,\ \pi \geq 0, \pi_0 \geq 0\}$$

is the cone of feasible solutions (rays). If FSP is feasible, then the optimal value of both FSP and CGSP is zero. Otherwise, CGSP is unbounded and a ray $(\hat{\pi}, \hat{\pi}_0)$ exists such that $\hat{\pi}^\top(b - B\hat{y}) + \hat{\pi}_0(f^\top \hat{y} - \hat{\eta}) > 0$; hence, the infeasible solution $(\hat{y}, \hat{\eta})$ violates the constraint

$$\hat{\pi}^\top(b - By) + \hat{\pi}_0(f^\top y - \eta) \leq 0.$$

Given $(\hat{\pi}, \hat{\pi}_0) \in \Pi$, we define $\mathcal{H}(\hat{\pi}, \hat{\pi}_0)$ and $\partial(\hat{\pi}, \hat{\pi}_0)$ as the half-space and hyperplane defined by $(\hat{\pi}, \hat{\pi}_0)$, respectively, i.e.,

$$\begin{aligned} \mathcal{H}(\hat{\pi}, \hat{\pi}_0) &= \{(y, \eta) : \hat{\pi}^\top(b - By) + \hat{\pi}_0(f^\top y - \eta) \leq 0\}, \\ \partial(\hat{\pi}, \hat{\pi}_0) &= \{(y, \eta) : \hat{\pi}^\top(b - By) + \hat{\pi}_0(f^\top y - \eta) = 0\}. \end{aligned}$$

Consequently, OP can be restated as the following modified master problem (MP):

$$\begin{aligned} [\text{MP}]\ \min\quad & \eta && && (2.4) \\ \text{s.t.}\quad & (y,\eta) \in \mathcal{H}(\hat{\pi},\hat{\pi}_0) && \forall(\hat{\pi},\hat{\pi}_0) \in \Pi && (2.5) \\ & \eta \in \mathbb{R}, y \in Y. && && (2.6) \end{aligned}$$

More precisely, this is a projection of (2.1) onto the (y, η) space. With this representation of Benders decomposition, at iteration t, we produce a candidate point $(y^{(t)}, \eta^{(t)})$ by solving a relaxation of MP, and test its feasibility using CGSP. If the test proves $(y^{(t)}, \eta^{(t)})$ infeasible, we generate a certificate $(\hat{\pi}, \hat{\pi}_0)$ and add a cut of the form (2.5) to the relaxed MP to avoid producing the infeasible $(y^{(t)}, \eta^{(t)})$ again; otherwise, we conclude that $(y^{(t)}, \eta^{(t)})$ is an optimal solution for MP. As noted in Proposition 2.1 below, cuts of the form (2.5) represent both Benders optimality and feasibility cuts; when $\hat{\pi}_0 > 0$, the cut corresponds to a classical Benders optimality cut, while $\hat{\pi}_0 = 0$ corresponds to a classical Benders feasibility cut. An overview of the BD algorithm based on this decomposition scheme is presented in Algorithm 1. Proofs for all of our theorems and propositions can be found in Appendix A.1.

Proposition 2.1. *Solution (y, η) satisfies constraints (2.5) if and only if $(y, \eta) \in \mathcal{E}$.*

Algorithm 1 Overview of Benders Decomposition algorithm

```
1: $t \leftarrow 1$, $\hat{\Pi}_t \leftarrow \emptyset$.
2: Solve MP with $\hat{\Pi}_t$ in place of $\Pi$ and obtain master solution $(y^{(t)}, \eta^{(t)})$.
3: Find a certificate $(\hat{\pi}, \hat{\pi}_0)$ for infeasibility of $(y^{(t)}, \eta^{(t)})$ using CGSP (2.3).
4: if certificate $(\hat{\pi}, \hat{\pi}_0)$ exists then
5:     Set $\hat{\Pi}_{t+1} \leftarrow \hat{\Pi}_t \cup \{(\hat{\pi}, \hat{\pi}_0)\}$, $t \leftarrow t + 1$ and go to step 2.
6: else
7:     Stop. $(y^{(t)}, \eta^{(t)})$ is an optimal solution for MP.
8: end if
```

At step 3 of Algorithm 1, CGSP provides a logical answer to whether the current master problem solution $(y^{(t)}, \eta^{(t)})$ is feasible (and hence optimal) for MP. But, not every logical answer is equally useful. In other words, to prove suboptimality of $(y^{(t)}, \eta^{(t)})$, CGSP produces a certificate $(\hat{\pi}, \hat{\pi}_0) \in \Pi$ such that $\hat{\pi}^\top(b - By^{(t)}) + \hat{\pi}_0(f^\top y^{(t)} - \eta^{(t)}) > 0$, without providing further information about how "far" $(y^{(t)}, \eta^{(t)})$ is from being optimal. Moreover, not only do we want to discard the trial solution $(y^{(t)}, \eta^{(t)})$, but we also want to rule out as many other unacceptable values of (y, η) as possible. Hence, we may phrase the key question of the BD algorithm as: *How should we select a certificate* $(\hat{\pi}, \hat{\pi}_0) \in \Pi$ *that conveys additional information about the sub-optimality of* $(y^{(t)}, \eta^{(t)})$*, so that we may exploit this information to speed up the convergence of the BD algorithm?* Our order of business in this article is to address this question by introducing selection strategies that exploit the properties of promising cuts in a computationally tractable manner.

At each iteration of the BD Algorithm 1, we wish to separate (if possible) the incumbent point $(y^{(t)}, \eta^{(t)})$ from the epigraph $\mathcal{E}$. As a result of Proposition 2.1, we may equivalently define $\mathcal{E}$ as

$$\mathcal{E} = \{(y, \eta) : (y, \eta) \in \mathcal{H}(\hat{\pi}, \hat{\pi}_0) \quad \forall (\hat{\pi}, \hat{\pi}_0) \in \Pi\}.$$

In cutting-plane theory, the separation problem produces a hyperplane (or a cut) that lies between a given point and a closed convex set. In our application, we want to separate the incumbent point $(y^{(t)}, \eta^{(t)})$ from the closed convex set $\mathcal{E}$ using a hyperplane $\partial(\pi, \pi_0)$ for some $(\pi, \pi_0) \in \Pi$. Note that infinitely many such hyperplanes may exist, thus one needs a selection criterion for producing the cut that "best" separates $(y^{(t)}, \eta^{(t)})$ from $\mathcal{E}$. While there is no universal definition of "best" cut, a "good" cut should satisfy some natural requirements. First, it should be a supporting hyperplane for $\mathcal{E}$, in the sense that it should touch $\mathcal{E}$ at some point. We further postulate that the cut must be *deep*, in the sense that it is as far from the given point $(y^{(t)}, \eta^{(t)})$ as possible. We begin in Section 2.1.2 with the Euclidean distance as our measure of cut depth, then generalize to distances induced by ℓ_p-norms in Section 2.1.3. Finally, in Section 2.1.4 we present an alternative primal perspective of deepest cut generation, and derive some important properties of deepest cuts. Specifically, we show that deepest cuts not only support $\mathcal{E}$, but also (i) minimally resolve infeasibility in the system

FSP, (ii) amount to optimality cuts, and (iii) are relatively flat thus, help close the gap quickly.

2.1.2 Euclidean Deepest Cuts

As our measure of cut depth, we start with the Euclidean distance from the point $(\hat{y}, \hat{\eta})$ to the hyperplane $\partial(\pi, \pi_0)$. Euclidean norm is the standard norm used in convex analysis, and measuring depth using this norm, also known as scaled violation, is also common practice in cutting-plane theory. For example, to produce deep facet-defining cuts for solving mixed-integer programs, Balas et al. (1993) and Cadoux (2010) use the Euclidean distance between the optimal vertex of the current relaxation and candidate separating hyperplanes; in a similar spirit, we also call the cuts we generate *deepest Benders cuts.*

Let $d(\hat{y}, \hat{\eta}|\pi, \pi_0)$ be the Euclidean distance between the point $(\hat{y}, \hat{\eta})$ and the hyperplane $\partial(\pi, \pi_0)$. From basic linear algebra, we know that the Euclidean distance from the point $\hat{z}$ to the hyperplane $\alpha^\top z + \beta = 0$ is $\frac{|\alpha^\top \hat{z} + \beta|}{\|\alpha\|_2}$. Hence,

$$d(\hat{y}, \hat{\eta}|\pi, \pi_0) = \frac{|\pi^\top (b - B\hat{y}) + \pi_0(f^\top \hat{y} - \hat{\eta})|}{\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_2} = \frac{\pi^\top (b - B\hat{y}) + \pi_0(f^\top \hat{y} - \hat{\eta})}{\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_2}, \tag{2.7}$$

where the last equality holds because $(\hat{y}, \hat{\eta})$ must (except at the last Benders iteration) violate the constraint $\pi^\top(b - By) + \pi_0(f^\top y - \eta) \leq 0$, hence $\pi^\top(b - B\hat{y}) + \pi_0(f^\top \hat{y} - \hat{\eta}) \geq 0$. To produce a deepest cut, we choose $(\pi, \pi_0) \in \Pi$ which maximizes this distance (i.e., depth) via

$$\text{[SSP] } d^*(\hat{y}, \hat{\eta}) = \max_{(\pi, \pi_0) \in \Pi} \quad d(\hat{y}, \hat{\eta}|\pi, \pi_0),$$

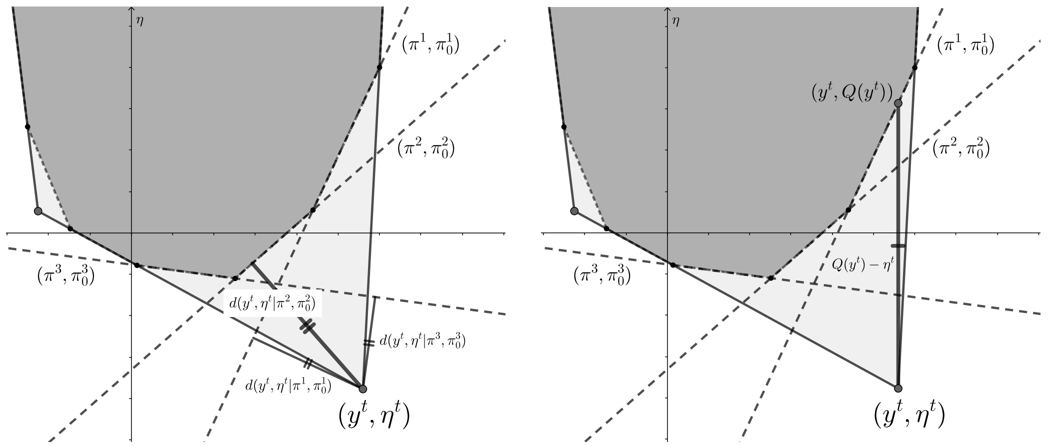
which we refer to as the *separation subproblem (SSP).* As we will show in §2.1.4, maximizing the distance of a separating hyperplane from the point $(\hat{y}, \hat{\eta})$ coincides with finding the minimum distance of $(\hat{y}, \hat{\eta})$ from the epigraph $\mathcal{E}$; thus we call $d^*(\hat{y}, \hat{\eta})$ the (Euclidean) distance of $(\hat{y}, \hat{\eta})$ from the epigraph $\mathcal{E}$. At iteration t of the BD algorithm, if $d^*(y^{(t)}, \eta^{(t)}) > 0$, then we can separate $(y^{(t)}, \eta^{(t)})$ from $\mathcal{E}$, otherwise $(y^{(t)}, \eta^{(t)}) \in \mathcal{E}$. Consequently, if $d^*(y^{(t)}, \eta^{(t)}) = 0$, then $(y^{(t)}, \eta^{(t)})$ is an optimal solution for MP.

Figure 2.1(a) illustrates how we select a certificate (π, π_0) by finding the hyperplane that is the maximum Euclidean distance from the master problem's solution $(y^{(t)}, \eta^{(t)})$. For demonstration purposes, we assume y is a continuous one-dimensional variable in this toy example. In this figure, the dark and light polygons represent $\mathcal{E}$ and the approximation of $\mathcal{E}$ at iteration t (i.e., $\mathcal{E}_t$), respectively. $(y^{(t)}, \eta^{(t)})$ is the current solution of the master problem to be separated. The dashed lines represent the hyperplanes associated with the dual solutions. The dual solution selected by the classical Benders subproblem is (π^1, π_0^1) whereas the solution selected based on Euclidean distance is (π^2, π_0^2).

As illustrated in Figure 2.1(b), the hyperplane produced by classical DSP supports $\mathcal{E}$ at $(y^{(t)}, Q(y^{(t)}))$. While the deepest cuts also support $\mathcal{E}$ (see Proposition 2.3), it does not necessarily do so at $(y^{(t)}, Q(y^{(t)}))$, but at a point which we call the *projection of* $(y^{(t)}, \eta^{(t)})$ *onto* $\mathcal{E}$. As illustrated in Figure 2.1(a), dual solutions (π^1, π_0^1), (π^2, π_0^2) and (π^3, π_0^3) (and their convex combinations) are the candidate solutions evaluated based on the Euclidean distance of their associated hyperplanes to the point $(y^{(t)}, \eta^{(t)})$, and (π^2, π_0^2) is selected as the deepest cut. It is worth pointing out that in the literature, the question of which Benders cut to select usually arises when the classical DSP admits alternative optimal solutions (Magnanti and Wong 1981). However, even when the classical DSP admits a unique optimal solution (as in the given example), the deepest cut may not coincide with the classical Benders cut.

2.1.3 ℓ_p-norm Deepest Cuts

We now generalize our notion of distance to that induced by any ℓ_p-norm. Our derivation begins with the observation that the denominator in (2.7) is the ℓ_2-norm of the vector of



(a) Deepest Benders cut

(b) Classical Benders cut

Figure 2.1: Deepest (a) versus classical (b) Benders cut selection.

coefficients $(\pi_0 f^\top - \pi^\top B, \pi_0)$. If we replace this norm with a general ℓ_p-norm for $p \geq 1$, and define $d_{\ell p}$ as

$$d_{\ell p}(\hat{y}, \hat{\eta}|\pi, \pi_0) = \frac{\pi^\top (b - B\hat{y}) + \pi_0 (f^\top \hat{y} - \hat{\eta})}{\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_p}, \tag{2.8}$$

then an *ℓ_p-norm deepest cut* (or *ℓ_p-deepest cut* for short) can be produced by solving the following separation problem

$$[\text{SSP}]\ d^*_{\ell p}(\hat{y}, \hat{\eta}) = \max_{(\pi, \pi_0) \in \Pi} \quad d_{\ell p}(\hat{y}, \hat{\eta}|\pi, \pi_0). \tag{2.9}$$

With this definition, $d_{\ell p}$ still measures the distance of $(\hat{y}, \hat{\eta})$ from the hyperplane $\partial(\pi, \pi_0)$, but unlike for the special case where $p = 2$ (Euclidean distance), the distance measure is no longer the ℓ_p-distance. In fact, as Proposition 2.2 shows below, $d_{\ell p}$ measures the distance between $(\hat{y}, \hat{\eta})$ and the hyperplane $\partial(\pi, \pi_0)$ with respect to the dual norm ℓ_q, where $\frac{1}{p} + \frac{1}{q} = 1$. The proof of this result relies on the definition of dual norm and is proven for general norms (including standard ℓ_p norms), but to be expositionally consistent we state the results for standard ℓ_p norms.

Proposition 2.2. *Given $q \geq 1$ and $\hat{z} \in \mathbb{R}^{n+1}$, the minimum ℓ_q-distance from the point $\hat{z}$*

to the points on the hyperplane $\alpha^\top z + \beta = 0$ is

$$\min_{z:\alpha^\top z+\beta=0} \|z - \hat{z}\|_q = \frac{|\alpha^\top \hat{z} + \beta|}{\|\alpha\|_p},$$

where ℓ_p is the dual norm of ℓ_q (i.e., $\frac{1}{p} + \frac{1}{q} = 1$).

Note that, as given in the proof of Proposition 2.2, we may extend the definition of deepest cuts by replacing the denominator in (2.8) with general norms (e.g., a composition of ℓ_p-norms with different p for different subsets of the components of $(\pi_0 f^\top - \pi^\top B, \pi_0)$). However, for clarity and simplicity of exposition, we restrict consideration in the remainder of this chapter to standard ℓ_p-norms.

Some choices of p for ℓ_p-deepest cuts merit special attention. In particular, for $p = 1$ and $p = \infty$, $d_{\ell p}$ defined by (2.8) measures the ℓ_∞ and ℓ_1 distance of $(\hat{y}, \hat{\eta})$ from the hyperplane $\partial(\pi, \pi_0)$, respectively. As we will show in §2.3.1, these norms are in general computationally favorable over the ℓ_2-norm since they result in linear separation subproblems.

As well, note that π_0 is the coefficient of η and $\pi^\top B - \pi_0 f^\top$ is the coefficient of y in the cut $\pi^\top b \leq (\pi^\top B - \pi_0 f^\top)y + \pi_0 \eta$. Therefore, deepest cuts effectively cut off the point $(\hat{y}, \hat{\eta})$ while minimizing the coefficients of the variables in the produced constraint. In particular, when the ℓ_1-norm is employed, producing deepest cuts mimics the idea of producing maximally nondominated Benders cuts introduced by Sherali and Lunday (2013), where the cut is maximally nondominated in the sense typically used in the cutting-plane theory for integer programs.

2.1.4 A Primal Projection Perspective of the Separation Problem

We now provide another view of deepest cuts, which will be important for analyzing their properties and paves the way for devising algorithms to produce them efficiently. By strong duality, we establish a duality between separation and projection as stated in Theorem 2.1 below.

Theorem 2.1. *Separation problem* (2.9) *is equivalent to the following Lagrangian dual problem*

$$\begin{aligned}
\text{[Primal SSP]} \quad \min \quad & \|(y-\hat{y}, \eta-\hat{\eta})\|_q \\
s.t. \quad & \eta \geq c^\top x + f^\top y \\
& Ax \geq b - By \\
& x \geq 0,
\end{aligned} \tag{2.10}$$

in which (y, x, η) *are the variables and* ℓ_q *is the dual norm of* ℓ_p*.*

The following result follows from strong duality and the definition of $\mathcal{E}$.

Corollary 2.1. $d^*_{\ell p}(\hat{y}, \hat{\eta})$ *measures the* ℓ_q *distance of* $(\hat{y}, \hat{\eta})$ *from* $\mathcal{E}$*. That is,*

$$d^*_{\ell p}(\hat{y}, \hat{\eta}) = \min_{(y,\eta)\in\mathcal{E}} \|y-\hat{y}, \eta-\hat{\eta}\|_q. \tag{2.11}$$

Let $(\tilde{y}, \tilde{\eta})$ be the optimal solution of the Primal SSP. In convex analysis, the solution of (2.11) for $q = 2$ is known as the *projection* of $(\hat{y}, \hat{\eta})$ onto $\mathcal{E}$. Thus, we refer to $(\tilde{y}, \tilde{\eta})$ henceforth as the ℓ_q-projection of $(\hat{y}, \hat{\eta})$, and refer to (2.11) as the *projection subproblem*. Figure 2.2 illustrates these projections for different values of q. In this figure, the dark and light polygons represent $\mathcal{E}$ and the approximation of $\mathcal{E}$ at iteration t, respectively. The point $(y^{(t)}, \eta^{(t)})$ is the current solution of the master problem to be separated. Finding the ℓ_p-deepest cut (i.e., normalizing dual solutions with ℓ_p-norm) accounts for resolving infeasibility in FSP (in the primal space) by finding a point in $\mathcal{E}$ with minimum ℓ_q-distance to $(y^{(t)}, \eta^{(t)})$, where ℓ_q is the dual norm of ℓ_p. The red lines represent the contour lines of the objective value of SSP, which also correspond to ℓ_q-balls around $(y^{(t)}, \eta^{(t)})$. Observe that the ℓ_q-projection or the ℓ_p-deepest cut might not be unique for $p = 1$ ($q = \infty$) or $p = \infty$ ($q = 1$). The following proposition states that deepest cuts support $\mathcal{E}$ at ℓ_q-projections, even when the projection or the cut are not unique.

Proposition 2.3. *Let $(\tilde{y}, \tilde{\eta}) \in \mathcal{E}$ be an ℓ_q-projection of $(\hat{y}, \hat{\eta})$ onto $\mathcal{E}$. Then, any ℓ_p-deepest cut separating $(\hat{y}, \hat{\eta})$ from $\mathcal{E}$ supports $\mathcal{E}$ at $(\tilde{y}, \tilde{\eta})$.*

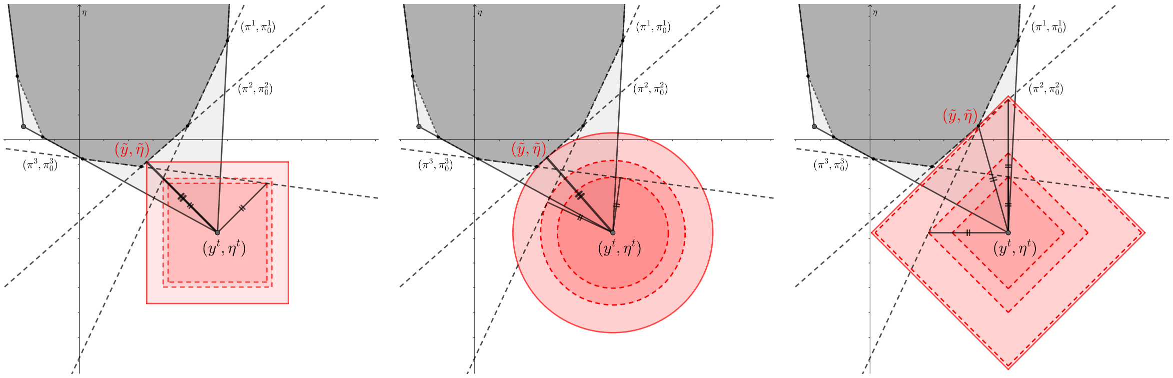
From the duality between separation and projection established in the above Theorem and Proposition, we derive the following important technical results.

Deepest cuts minimally resolve infeasibility in FSP. By Theorem 2.1 and as illustrated in Figure 2.2, producing an ℓ_p-deepest cut amounts to finding the point $(\tilde{y}, \tilde{\eta})$ of least ℓ_q-distance from $(\hat{y}, \hat{\eta})$ for which a feasible solution x exists that satisfies the system

$$\left\{c^\top x \leq \tilde{\eta} - f^\top \tilde{y}; \quad Ax \geq b - B\tilde{y}; \quad x \geq 0\right\}.$$

Hence, producing a deepest cut can be viewed as resolving infeasibility of FSP (2.2) through minimal deviation from $(\hat{y}, \hat{\eta})$ with respect to the ℓ_q-norm. If FSP is feasible for $(\hat{y}, \hat{\eta})$ (i.e., if $\|\tilde{y} - \hat{y}, \tilde{\eta} - \hat{\eta}\|_q = 0$), then $(\hat{y}, \hat{\eta})$ is optimal for MP. Effectively, $d^*_{\ell p}(\hat{y}, \hat{\eta})$ measures how far $(\hat{y}, \hat{\eta})$ is from being optimal by measuring the minimal deviation in $(\hat{y}, \hat{\eta})$ that renders FSP feasible. Thus, producing a deepest cut assesses how inaccurate our current guess of the optimal solution is.

Sparsity, density and flatness of deepest cuts. We have empirically observed that the deepest cuts generated at the early stages of the BD algorithm tend to be (relatively) flat. That is, the coefficients of the y-variables in the cut are mostly zero, and in some cases the cut is completely flat, i.e., all y coefficients are zero. Here, we provide a justification for this observation and discuss its implications. Along this vein, we first note the following



(a) ℓ_1-deepest cut ($p = 1$, $q = \infty$) (b) ℓ_2-deepest cut ($p = 2$, $q = 2$) (c) ℓ_∞-deepest cut ($p = \infty$, $q = 1$)

Figure 2.2: Primal and dual perspectives of the separation problem.

property of ℓ_1-deepest cuts.

Proposition 2.4. *For sufficiently small $\hat{\eta}$, the ℓ_1-deepest cut separating $(\hat{y}, \hat{\eta})$ from $\mathcal{E}$ is the flat cut $\eta \geq Q^*$, where $Q^* = \min_y Q(y)$ is the optimal value of Q for unrestricted y.*

Proposition 2.4 implies that, at early iterations of the BD algorithm, an ℓ_1-deepest cut can provide a lower bound of at least Q^*. Since Q^* is obtained by relaxing $y \in Y$, Q^* is at most equal to the optimal value of the LP relaxation of OP. While the quality of this bound is problem-specific, we have observed that the bound is indeed very close to the optimal value of OP when the integrality gap is low (e.g., in facility location problems).

More generally, for small p (i.e., large q) and relatively small $\hat{\eta}$, we may approximate $\|(y - \hat{y}, \eta - \hat{\eta})\|_q \approx \eta - \hat{\eta}$. Therefore, in line with Proposition 2.4, we can expect that the coefficients of the y-variables in the ℓ_p-deepest cuts (i.e., $\hat{\pi}_0 f^\top - \hat{\pi}^\top B$) will be close to zero, which means that the deepest cuts are relatively *sparse*. This observation is also in line with using Lasso or ℓ_1-regularization in machine learning and statistics for producing sparse solutions (see e.g., Tibshirani 1996). By the same token, large values of p (e.g., $p = \infty$) induce *dense* cuts, in that the coefficients of the y variables are mostly non-zero.

Deepest cuts are more likely to be optimality cuts than feasibility cuts. In our experiments with deepest cuts, we found that they are likely to be optimality cuts, even when classical Benders produces a feasibility cut (i.e., even when $\hat{y} \notin \operatorname{dom}(Q)$). Intuitively, since at each iteration of BD, $\eta^{(t)}$ is an under-estimator of the convex piece-wise linear function Q and deepest cuts support $\mathcal{E}$, the coefficient of η in a deepest cut (namely, π_0) is most likely non-zero (i.e., the cut is an optimality cut). More specifically, Propositions 2.5 and 2.6 below provide sufficient conditions for when deepest cuts are guaranteed to be optimality cuts.

Proposition 2.5. *Let $Q^* = \min_y Q(y)$ be the optimal unrestricted value of Q. Provided that $\hat{\eta} < Q^*$ and $p > 1$, the ℓ_p-deepest cut(s) separating $(\hat{y}, \hat{\eta})$ are optimality cuts for any arbitrary $\hat{y}$ (i.e., even if $\hat{y} \notin \operatorname{dom}(Q)$).*

Proposition 2.6. *Assume the ℓ_q-projection of $(\hat{y}, \hat{\eta})$ onto $\mathcal{E}$ is unique, and denote it by the point $(\tilde{y}, \tilde{\eta})$. If $\hat{\eta} < \tilde{\eta}$, then the ℓ_p-deepest cuts separating $(\hat{y}, \hat{\eta})$ are optimality cuts for any arbitrary $\hat{y}$ (i.e., even if $\hat{y} \notin \operatorname{dom}(Q)$).*

Proposition 2.5 guarantees that deepest cuts are optimality cuts when $\hat{\eta}$ is sufficiently small. Proposition 2.6 further suggests that, even if $\hat{\eta}$ is not very small, deepest cuts are more likely to be optimality cuts (note that the ℓ_q-projection is always unique for $1 < q < \infty$). This is particularly appealing from a practical standpoint, as the contribution of Benders optimality cuts to closing the gap is usually more pronounced than that of feasibility cuts (see e.g., Saharidis and Ierapetritou 2010, de Sá et al. 2013). This property of deepest cuts is in line with the cut generation strategy proposed by Saharidis and Ierapetritou (2010), where to speed up convergence, they produce an additional optimality cut whenever a feasibility cut is needed.

2.2 General Benders Distance Functions

In this section, we introduce a general distance function based on duality theory, which we call a *Benders distance function*. Such a distance function (a) identifies whether the incumbent point $(\hat{y}, \hat{\eta})$ is inside, outside, or on the boundary of the epigraph $\mathcal{E}$, and (b) if outside, conveys how "far" the incumbent point is from the boundary. Crucially, we do not explicitly define the metric on which a Benders distance function is based; this is by design, and interestingly is not needed. In fact, we show in Section 2.2.1 that so long as a monotonicity property linked to convexity holds, then a sufficient notion of distance exists. Then, in Section 2.2.2 we introduce an important special class of Benders distance function, normalized distance functions, which we will use in Section 2.3 to connect deepest cuts with other types of Benders cuts from the literature. Finally, in Section 2.2.3 we introduce a distance-based BD algorithm and study its convergence properties.

2.2.1 Definition

We define a Benders distance function, which is a generalization of the geometric distance functions induced by ℓ_p-norms presented earlier, as follows.

Definition 2.1 (**Benders distance function**). *Function $d(\hat{y}, \hat{\eta}|\pi, \pi_0) : \mathbb{R}^{n+1} \times \Pi \to \mathbb{R}$ is a Benders distance function if (i) it certifies $d(\hat{y}, \hat{\eta}|\pi, \pi_0) > 0$ iff $(\hat{y}, \hat{\eta})$ is in the exterior of $\mathcal{H}(\pi, \pi_0)$, $d(\hat{y}, \hat{\eta}|\pi, \pi_0) = 0$ iff $(\hat{y}, \hat{\eta})$ is on the boundary of $\mathcal{H}(\pi, \pi_0)$, and $d(\hat{y}, \hat{\eta}|\pi, \pi_0) < 0$ iff $(\hat{y}, \hat{\eta})$ is in the interior of $\mathcal{H}(\pi, \pi_0)$, and (ii) $d^*(\hat{y}, \hat{\eta})$ defined below is* ***convex****:*

$$\text{[BSP] } d^*(\hat{y}, \hat{\eta}) = \sup_{(\pi, \pi_0) \in \Pi} \quad d(\hat{y}, \hat{\eta}|\pi, \pi_0). \tag{2.12}$$

Definition 2.2 (**Epigraph distance function**). *For a given Benders distance function d, we call d^* as defined in* (2.12) *the epigraph distance function induced by d.*

Proposition 2.7. *Epigraph distance function d^* certifies $d^*(\hat{y}, \hat{\eta}) > 0$ iff $(\hat{y}, \hat{\eta})$ is in the exterior of $\mathcal{E}$, $d^*(\hat{y}, \hat{\eta}) = 0$ iff $(\hat{y}, \hat{\eta})$ is on the boundary of $\mathcal{E}$, and $d^*(\hat{y}, \hat{\eta}) < 0$ iff $(\hat{y}, \hat{\eta})$ is in the interior of $\mathcal{E}$.*

By Proposition 2.7, d^* maps each point $(\hat{y}, \hat{\eta}) \in \mathbb{R}^{n+1}$ to a value in the extended real number line (i.e., $\mathbb{R} \cup \{\pm\infty\}$) such that the sign of $d^*(\hat{y}, \hat{\eta})$ determines if $(\hat{y}, \hat{\eta})$ is in the exterior (sign $= +1$), interior (sign $= -1$) or on the boundary of $\mathcal{E}$ (sign $= 0$). The convexity requirement for d^* is tied to the epigraph distance function d^* being monotonic in the following sense.

Definition 2.3 **(Monotonicity of epigraph distance function).** *For arbitrary $(\hat{y}, \hat{\eta}) \notin \mathcal{E}$ and $(y^0, \eta^0) \in \partial\mathcal{E}$ (boundary of $\mathcal{E}$) such that the open line segment between $(\hat{y}, \hat{\eta})$ and (y^0, η^0) lies in the exterior of $\mathcal{E}$, define $d^*(\alpha) = d^*((1-\alpha)(y^0, \eta^0) + \alpha(\hat{y}, \hat{\eta}))$. We say the epigraph distance function d^* is* ***monotonic*** *if $d^*(\alpha_1) \leq d^*(\alpha_2)$ for any $0 \leq \alpha_1 < \alpha_2 \leq 1$. We say d^* is* ***strongly monotonic*** *if $d^*(\alpha_1) < d^*(\alpha_2)$ for any $0 \leq \alpha_1 < \alpha_2 \leq 1$.*

Intuitively, and as illustrated in Figure 2.3(b), (strong) monotonicity of d^* implies that as α increases, we move away from the boundary of $\mathcal{E}$ and distance increases (i.e., d^* becomes larger). Theorem 2.2 formally establishes that $d^*(\hat{y}, \hat{\eta})$ can be viewed as how far $(\hat{y}, \hat{\eta})$ is from the boundary of $\mathcal{E}$. The proof of this result uses the fact that d^* is a convex function.

Theorem 2.2. *Epigraph distance functions are monotonic.*

We end this section by pointing out that our definition of Benders distance functions is sufficiently general to allow us to bring several well-known cut selection strategies under one umbrella, which we will discuss in further detail in Section 2.3. However, in that section we will show that classical Benders cuts, which we would not consider particularly "deep", can also be generated using a Benders distance function. In essence, just because a function may be classified as a Benders distance function does not automatically mean that strong, deep cuts will be generated using it. For this reason, we henceforth reserve the term "deepest cuts" to mean cuts that are produced based on geometric distance functions (i.e., $d_{\ell p}$), which correspond to what we believe most people would intuitively consider "deep cuts". Finally, we remark that the epigraph distance functions induced by $d_{\ell p}$ are strongly monotonic, which draws another distinction between these distance functions and general Benders distance functions.

Proposition 2.8. *Epigraph distance function $d^*_{\ell p}$ (2.9) is strongly monotonic for any $p \geq 1$.*

2.2.2 Normalized Distance Functions and Normalization Functions

We now introduce an important special class of Benders distance functions, which we call *normalized Benders distance functions.* This is a generalization of the geometric ℓ_p-norm-based Benders distance functions that we introduced in Section 2.1.3. The distance functions in this class are constructed by replacing the denominator of the distance function $d_{\ell p}$ (2.8) with a general *normalization function* which is only required to be a positive homogeneous function of the dual variables.

Definition 2.4 (Normalized distance function). *Let $d_g(\hat{y}, \hat{\eta}|\pi, \pi_0) = \frac{\pi^\top(b-B\hat{y})+\pi_0(f^\top\hat{y}-\hat{\eta})}{g(\pi,\pi_0)}$ where g is a positive homogeneous function (i.e., $g(\alpha\pi, \alpha\pi_0) = \alpha g(\pi, \pi_0)$ for any $\alpha \geq 0$). We call d_g a **normalized distance function**, and refer to g as its **normalization function**.*

The normalization function g governs the behavior of the distance function, and quantifies our perception of the quality of the cut it produces. Homogeneity of g is critical. Indeed, with constant (i.e., non-homogeneous) $g(\pi, \pi_0) = 1$, BSP is equivalent to the naïve CGSP (2.3), for which $d^*(y^{(t)}, \eta^{(t)}) \in \{0, +\infty\}$. In this case, d^* is simply a characteristic function of $\mathcal{E}$, which, regardless of the quality of the cut, merely provides a binary answer to whether or not $(y^{(t)}, \eta^{(t)})$ is the optimal solution, without any further indication of how far $(y^{(t)}, \eta^{(t)})$ is from being optimal.

For a positive homogeneous function g, let $\Pi_g = \{(\pi, \pi_0) \in \Pi : g(\pi, \pi_0) \leq 1\}$ be the cone Π truncated by the constraint $g(\pi, \pi_0) \leq 1$, and define the normalized separation problem (NSP) as

$$[\text{NSP}] \max_{(\pi,\pi_0)\in\Pi_g} \pi^\top(b - B\hat{y}) + \pi_0(f^\top\hat{y} - \hat{\eta}). \tag{2.13}$$

We first note the following correspondence between normalized distance functions and normalized separation problems, which sheds light on the role and desirable properties of the normalization function g, and paves the way for reformulating the separation problems. It operates by generalizing the Charnes-Cooper transformation (Charnes and Cooper 1962) for linear-fractional programs.

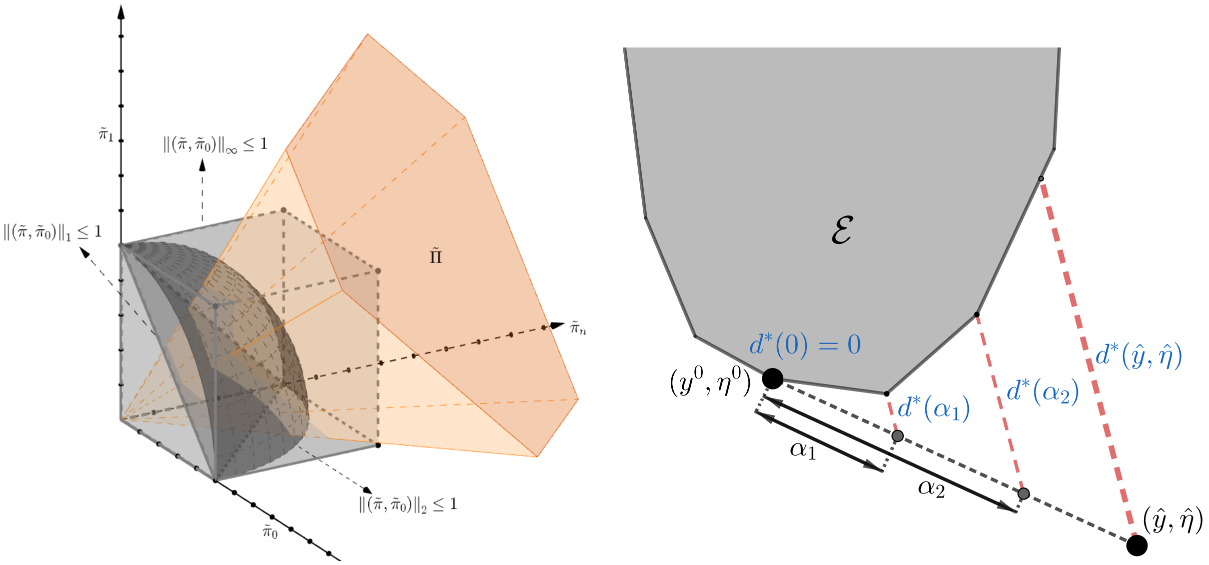
Proposition 2.9. *Let* $d_g(\hat{y}, \hat{\eta}|\pi, \pi_0) = \frac{\pi^\top(b-B\hat{y})+\pi_0(f^\top\hat{y}-\hat{\eta})}{g(\pi,\pi_0)}$ *be a normalized distance function. Then, the separation problem* (2.12) *is equivalent to the normalized separation problem* (2.13)*. That is,*

$$d_g^*(\hat{y}, \hat{\eta}) = \max_{(\pi,\pi_0)\in\Pi_g} \pi^\top(b - B\hat{y}) + \pi_0(f^\top\hat{y} - \hat{\eta}).$$

Additionally, $g(\pi, \pi_0) \leq 1$ *is binding at optimality.*

From a polyhedral perspective, Proposition 2.9 shows the equivalence between choosing a distance function and truncating the cone Π with a specific normalization function. Note that all general norms (including the ℓ_p-norms introduced in §2.1.3) are normalization functions. Figure 2.3(a) illustrates the effect of ℓ_1-, ℓ_2- and ℓ_∞-norms on truncating the cone Π. For illustration, the space of dual variables $(\pi, \pi_0) \in \Pi$ is transformed from $\mathbb{R}^{m+1}$ to the $\mathbb{R}^{n+1}$-space via $(\tilde{\pi}, \tilde{\pi}_0) = (\pi_0 f^\top - \pi^\top B, \pi_0)$ and Π is mapped to this space as $\tilde{\Pi} = \{(\tilde{\pi}, \tilde{\pi}_0) \in \mathbb{R}^{n+1} : (\tilde{\pi}, \tilde{\pi}_0) = (\pi_0 f^\top - \pi^\top B, \pi_0), (\pi, \pi_0) \in \Pi\}$.

From a computational point of view, Proposition 2.9 shows that, for convex g, BSP can be



(a) Normalization functions truncate Π.

(b) Epigraph distance functions are monotonic.

Figure 2.3: (a) Effect of ℓ_1-norm (unit simplex), ℓ_2-norm (sphere) and ℓ_∞-norm (box) on truncating the cone Π. (b) As we move away from the boundary of $\mathcal{E}$, d^* gets larger.

converted into a problem of optimizing a linear function over a convex set Π_g. Additionally, since Π_g does not depend on $(y^{(t)}, \eta^{(t)})$, one may leverage the reoptimization capabilities of the solver whenever possible. For instance, a certificate produced at iteration t of the BD algorithm can be used for warm starting the separation subproblem at iteration $t + 1$. In particular, a convex piece-wise linear function $g(\pi, \pi_0)$ amounts to solving linear programs with different objective function coefficients at each iteration, thus can be reoptimized using a primal simplex method. We provide further details in §2.3.

Finally, as stated in Proposition 2.10 below, any normalized distance function induces a monotonic epigraph distance function.

Proposition 2.10. *Normalized distance function d_g induces a monotonic epigraph distance function d_g^* for any normalization function g.*

2.2.3 Distance-Based Benders Decomposition Algorithm

We present an overview of our proposed Benders decomposition algorithm based on general Benders distance functions in Algorithm 2. Theorem 2.3, below, establishes finite convergence of this algorithm for a specific practical class of Benders distance function.

Theorem 2.3. *Let d_g be a Benders normalized distance function with a convex piece-wise linear normalization function g. Then BD Algorithm 2 converges to an optimal solution or asserts infeasibility of MP in a finite number of iterations.*

In particular, Algorithm 2 is finitely convergent when g is a linear function of (π, π_0) (see §2.3.2 for such linear functions) or when ℓ_1- or ℓ_∞-deepest cuts are produced. For other cases (e.g., Euclidean deepest cuts), one may choose to employ Euclidean deepest cuts in conjunction with known finitely convergent separation routines (e.g., classical BD cuts) to guarantee convergence while continuing to benefit from the desirable properties of deepest cuts.

Algorithm 2 Distance-based Benders Decomposition algorithm

```
1: Select a Benders distance function $d$.
2: $t \leftarrow 1$, $\hat{\Pi}_t \leftarrow \emptyset$
3: Solve MP with $\hat{\Pi}_t$ in place of $\Pi$ and obtain master solution $(y^{(t)}, \eta^{(t)})$.
4: Solve BSP (2.12) to obtain $d^*(y^{(t)}, \eta^{(t)})$ and the optimal solution $(\hat{\pi}, \hat{\pi}_0)$.
5: if $d^*(y^{(t)}, \eta^{(t)}) > 0$ then
6:     Set $\hat{\Pi}_{t+1} \leftarrow \hat{\Pi}_t \cup \{(\hat{\pi}, \hat{\pi}_0)\}$, $t \leftarrow t + 1$ and loop to Step 3.
7: else
8:     Stop. $(y^{(t)}, \eta^{(t)})$ is an optimal solution for MP with optimal value $\eta^{(t)}$.
9: end if
```

2.3 Reformulations and Connections to other Cut Selection Strategies

In this section, we present some reformulations that may be used to implement distance-based cuts, as well as highlight special cases of normalized distance functions which link our concept of Benders distance function to other cut selection strategies in the literature. Section 2.3.1 begins by presenting reformulations of the separation problems introduced in Section 2.1, which can be used to computationally generate ℓ_p-deepest cuts using linear or quadratic programming solvers. Then, Section 2.3.2 considers the special case of normalized distance functions where the normalization function is linear, and examines how specific choices of linear coefficients correspond to cut selection strategies in the literature.

2.3.1 ℓ_p-deepest Cuts

We first show how SSP (2.9) can be cast as linear/quadratic programs using standard reformulation techniques. By homogeneity of ℓ_p-norms, using Proposition 2.9 we may restate SSP (2.9) as

$$\begin{aligned} \max_{(\pi,\pi_0)\in\Pi} \quad & \pi^\top(b - B\hat{y}) + \pi_0(f^\top\hat{y} - \hat{\eta}) \\ \text{s.t.} \quad & \|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_p \leq 1. \end{aligned} \tag{2.14}$$

Using this reformulation, we may express the constraint $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_p \leq 1$ as a set of linear/quadratic constraints depending on the choice of p as follows.

- For $p = \infty$, we have $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_\infty = \max\left\{\pi_0, \max_{j=1,\ldots,n}\{|\pi_0 f_j - \pi^\top B_{\cdot j}|\}\right\}$, where $B_{\cdot j}$ is the j'th column of matrix B. Therefore, $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_\infty \leq 1$ can be represented by the $2n$ linear constraints $-1 \leq \pi_0 f_j - \pi^\top B_{\cdot j} \leq 1$ for each j, and a bound constraint $\pi_0 \leq 1$.

- For $p = 1$, we may rewrite $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_1 = \pi_0 + \sum_{j=1}^n |\pi_0 f_j - \pi^\top B_{.j}| \leq 1$ as $\pi_0 + \sum_{j=1}^n \tau_j \leq 1$ by introducing n new variables $\tau \in \mathbb{R}_+^n$ and $2n$ constraints $-\tau \leq \pi_0 f - \pi^\top B \leq \tau$.

- For $p = 2$, one only needs to rewrite $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_2 \leq 1$ as $\pi_0^2 + \sum_{j=1}^n (\pi_0 f_j - \pi^\top B_{.j})^2 \leq 1$ to cast (2.14) as a convex quadratically constrained linear program.

- For $p > 2$ and integer, note that $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_p \leq 1$ is equivalent to $\pi_0^p + \sum_{j=1}^n \tau_j^p \leq 1$, where $-\tau_j \leq \pi_0 f_j - \pi^\top B_{.j} \leq \tau_j$. The constraint $\pi_0^p + \sum_{j=1}^n \tau_j^p \leq 1$ can be expressed as quadratic constraints using a series of transformations. For instance, with $p = 4$, it is not difficult to see that $\pi_0^4 + \sum_{j=1}^n \tau_j^4 \leq 1$ may be expressed using auxiliary variables $\{\beta_j\}_{j=0}^n$ as the following set of second-order constraints; similar transformations may be used for other values of p.

$$\beta_0^2 + \sum_{j=1}^n \beta_j^2 \leq 1, \quad \pi_0^2 \leq \beta_0, \quad \tau_j^2 \leq \beta_j \quad \forall j.$$

2.3.2 Linear Pseudonorms

Consider the class of normalization functions defined by choosing parameters (w, w_0) such that $g(\pi, \pi_0) = \pi^\top w + \pi_0 w_0 \geq 0$ for all $(\pi, \pi_0) \in \Pi$. In this subsection, we study how different values of the (w, w_0) parameters impact the resulting normalized Benders distance function, as well as how the cuts produced relate to other cut selection strategies in the literature. Note that a linear function g of this form satisfies most axioms of a norm; that is, g is subadditive (i.e., $g(u + v) \leq g(u) + g(v)$), homogeneous (i.e., $g(\alpha u) = \alpha g(u)$ for any $\alpha \geq 0$), and positive over Π, but not necessarily positive definite (i.e., $g(\pi, \pi_0) = 0$ does not necessarily imply $(\pi, \pi_0) = 0$, unless $(w, w_0) > 0$). Hence, we call g a *linear pseudonorm* over Π. With $g(\pi, \pi_0) = \pi^\top w + \pi_0 w_0$, the separation problem (2.13) can be stated as the following linear program

$$\begin{aligned} \max \quad & \pi^\top (b - B\hat{y}) + \pi_0(f^\top \hat{y} - \hat{\eta}) \\ \text{s.t.} \quad & \pi^\top A \le \pi_0 c^\top \\ & \pi^\top w + \pi_0 w_0 = 1 \\ & \pi \ge 0, \pi_0 \ge 0, \end{aligned} \tag{2.15}$$

which contains only one additional variable and one additional constraint compared to DSP (1.5) in the classical BD algorithm. Note that the normalization constraint is an equality constraint since $g(\pi, \pi_0)$ is binding at optimality.

Separation problem (2.15) is similar to the MIS (minimal infeasible subsystems) subproblem proposed by Fischetti et al. (2010). They derive the MIS subproblem by treating the separation problem as approximating the minimal source of infeasibility of FSP (2.2) by minimizing a positive linear function $\pi^\top w + \pi_0 w_0$ over the alternative polyhedron of Π (i.e., Π truncated by constraint $\pi^\top (b - B\hat{y}) + \pi_0(f^\top \hat{y} - \hat{\eta}) = 1$). Therefore, the MIS subproblem can be viewed as a special type of the Benders separation subproblem (2.12) in which the normalization function g takes the form of $g(\pi, \pi_0) = \pi^\top w + \pi_0 w_0$.

As noted by Fischetti et al. (2010), the choice of the normalization coefficients (w, w_0) can have a profound impact on the effectiveness of MIS cuts. In their implementation, the authors set $w_0 = 1$ and initially set $w_i = 1$, for all $i = 1, \ldots, m$. They further suggest that setting $w_i = 0$ for the null rows of B (i.e., row i such that $B_{ij} = 0$ for all j) may lead to substantial improvement in the convergence of the BD algorithm. Below, we propose four ways for choosing parameters (w, w_0) based on the parameters of the problem instance and discuss their implications and connections to other cut selection strategies.

2.3.2.1 Benders pseudonorm.

A trivial choice for parameters w and w_0 is to set $w = 0$ and $w_0 = 1$, which leads to setting $\pi_0 = 1$, thus (2.15) reduces to the dual subproblem (1.5) in the classical BD algorithm. In other words, if we define $d_{\text{CB}}(\hat{y}, \hat{\eta}|\pi, \pi_0) = \frac{\pi^\top(b-B\hat{y})+\pi_0(f^\top\hat{y}-\hat{\eta})}{\pi_0}$, as the classical Benders (CB) distance function, then

$$d^*_{\text{CB}}(\hat{y}, \hat{\eta}) = \max_{(\pi,\pi_0)\in\Pi} d_{\text{CB}}(\hat{y}, \hat{\eta}|\pi, \pi_0) = \tilde{Q}(\hat{y}) + f^\top\hat{y} - \hat{\eta} = Q(\hat{y}) - \hat{\eta}.$$

Therefore, as illustrated in Figure 2.1(b), $d^*_{\text{CB}}(\hat{y}, \hat{\eta})$ can be geometrically interpreted as the distance from the point $(\hat{y}, \hat{\eta})$ to the boundary of $\mathcal{E}$ along the η-axis. Observe that, at iteration t of BD, $Q(y^{(t)}) - \eta^{(t)} \geq UB - LB$, where $LB = \eta^{(t)}$ and UB is the best upper bound identified by the algorithm so far. Hence, $d^*_{\text{CB}}(y^{(t)}, \eta^{(t)})$ estimates how far $(y^{(t)}, \eta^{(t)})$ is from being optimal by overestimating the optimality gap. Thus, $d^*_{\text{CB}}(y^{(t)}, \eta^{(t)}) = 0$ means $(y^{(t)}, \eta^{(t)})$ is an optimal solution to (1.4), which is exactly the stopping criterion used in the classical BD algorithm.

2.3.2.2 Relaxed ℓ_1 pseudonorm.

Expanding the ℓ_1-norm as $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_1 = \pi_0 + \sum_{j=1}^n |\pi_0 f_j - \pi^\top B_{.j}| = \pi_0 + \sum_{j=1}^n |\pi_0 f_j - \sum_{i=1}^m \pi_i B_{ij}|$ and using the triangle inequality, we obtain

$$\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_1 \leq \pi_0(1 + \sum_{j=1}^n |f_j|) + \sum_{i=1}^m \pi_i \sum_{j=1}^n |B_{ij}|.$$

Hence, we refer to $g(\pi, \pi_0) = \pi^\top w + \pi_0 w_0$ with $w_0 = 1 + \sum_{j=1}^n |f_j|$ and $w_i = \sum_{j=1}^n |B_{ij}|$ for $i = 1, \ldots, m$ as the *relaxed ℓ_1 pseudonorm* (and write $R\ell_1$ for short). Note that, since $B_{ij} = 0$ for the null rows of B, $R\ell_1$ automatically sets $w_i = \sum_{j=1}^n |B_{ij}| = 0$ for the null rows of B, which is in line with the intuitive suggestion of Fischetti et al. (2010). Furthermore, observe that $d_{R\ell 1}$ defined as

$$d_{R\ell 1}(\hat{y}, \hat{\eta}|\pi, \pi_0) = \frac{\pi^\top(b - B\hat{y}) + \pi_0(f^\top \hat{y} - \hat{\eta})}{\pi^\top w + \pi_0 w_0}$$

underestimates $d_{\ell 1}$. Thus, maximizing $d_{R\ell 1}$ serves as a surrogate for maximizing $d_{\ell 1}$, but by solving a simpler LP. As stated in Proposition 2.11 below, the choice of (w, w_0) according to $R\ell_1$ not only assigns meaningful values to the normalization parameters in the MIS subproblem, but also leads to a geometric interpretation of the MIS subproblem.

Proposition 2.11. *The following relationship between the epigraph distance functions induced by d_{CB}, $d_{\ell p}$, and $d_{R\ell 1}$ holds:*

$$d^*_{CB}(\hat{y}, \hat{\eta}) = Q(\hat{y}) - \hat{\eta} \geq d^*_{\ell\infty}(\hat{y}, \hat{\eta}) \geq \cdots \geq d^*_{\ell p}(\hat{y}, \hat{\eta}) \geq \cdots \geq d^*_{\ell 1}(\hat{y}, \hat{\eta}) \geq d^*_{R\ell 1}(\hat{y}, \hat{\eta}).$$

2.3.2.3 Magnanti-Wong-Papadakos pseudonorm.

The Magnanti-Wong procedure for producing a Pareto-optimal cut using a given core point $\bar{y}$ (i.e., $\bar{y} \in \text{relint}(Y)$) involves solving the following subproblem (Magnanti and Wong 1981):

$$\max \left\{ u^\top(b - B\bar{y}) : u^\top(b - B\hat{y}) = \tilde{Q}(\hat{y}), u \in \mathcal{U} \right\}, \tag{2.16}$$

where $\mathcal{U} = \{u \geq 0 : u^\top A \leq c^\top\}$ and $\tilde{Q}(\hat{y})$ is obtained by solving DSP (1.5). The constraint $u^\top(b - B\hat{y}) = \tilde{Q}(\hat{y})$ in (2.16) is imposed to guarantee that the dual solution u is one of the alternative optimal solutions of the DSP induced by $\hat{y}$. However, as noted by Papadakos (2008), one can still produce a Pareto-optimal cut by suppressing this constraint and solving

$$\tilde{Q}(\bar{y}) = \max \left\{ u^\top(b - B\bar{y}) : u \in \mathcal{U} \right\}. \tag{2.17}$$

Note that $\tilde{Q}(\bar{y}) - u^\top(b - B\bar{y}) \geq 0$ for any $u \in \mathcal{U}$, and problem (2.17) is equivalent to minimizing $\tilde{Q}(\bar{y}) - u^\top(b - B\bar{y})$. Additionally, $\frac{\pi}{\pi_0} \in \mathcal{U}$ for any $(\pi, \pi_0) \in \Pi$ such that $\pi_0 > 0$. Consequently, one can approximate a Pareto-optimal cut when cutting off the point $(\hat{y}, \hat{\eta})$ by employing

$$d_{\text{MWP}}(\hat{y}, \hat{\eta} | \pi, \pi_0) = \frac{\pi^\top(b - B\hat{y}) + \pi_0(f^\top\hat{y} - \hat{\eta})}{\pi_0\tilde{Q}(\bar{y}) - \pi^\top(b - B\bar{y})},$$

as the distance function, which is equivalent to setting $g(\pi, \pi_0) = \pi^\top w + \pi_0 w_0$ with $(w, w_0) = (B\bar{y} - b, \tilde{Q}(\bar{y}))$. We refer to $g(\pi, \pi_0)$ with this choice of (w, w_0) as the *Magnanti-Wong-Papadakos (MWP) pseudonorm*, which further connects the distance functions to this well-known cut selection strategy. Note that $\tilde{Q}(\bar{y})$ needs to be computed only once and that the normalization function remains the same in the course of BD Algorithm 2.

2.3.2.4 Conforti-Wolsey pseudonorm.

Recently, Conforti and Wolsey (2019) proposed an interesting procedure for producing facet-defining cuts using a core point. Given a core point $\bar{y}$ and its optimal value $Q(\bar{y}) = \tilde{Q}(\bar{y}) + f^\top\bar{y}$, the geometric interpretation of this idea is to find the closest point to $(\hat{y}, \hat{\eta})$ on the line segment between $(\bar{y}, Q(\bar{y}))$ and $(\hat{y}, \hat{\eta})$ that renders FSP (2.2) feasible. In our context, their procedure translates to solving

$$\begin{aligned} \min \quad & \lambda \\ \text{s.t.} \quad & -c^\top x + \lambda\left(Q(\bar{y}) - \hat{\eta} - f^\top(\bar{y} - \hat{y})\right) \geq -\hat{\eta} + f^\top\hat{y} \\ & Ax + \lambda B(\bar{y} - \hat{y}) \geq b - B\hat{y} \\ & x \geq 0, 1 \geq \lambda \geq 0. \end{aligned} \tag{2.18}$$

First, note that we may suppress $\lambda \leq 1$ since $(\bar{y}, Q(\bar{y}))$ is feasible for FSP (2.2). Next, assigning dual variable π_0 to the first constraint and π to the second set of constraints, we may express (2.18) in its dual form as

$$\begin{aligned} \max \quad & \pi^\top(b - B\hat{y}) + \pi_0(f^\top\hat{y} - \hat{\eta}) \\ \text{s.t.} \quad & \pi^\top B(\bar{y} - \hat{y}) + \pi_0\left(Q(\bar{y}) - \hat{\eta} - f^\top(\bar{y} - \hat{y})\right) \leq 1 \\ & (\pi, \pi_0) \in \Pi, \end{aligned}$$

which is equivalent to employing $g(\pi, \pi_0) = \pi^\top w + \pi_0 w_0$ in the normalized separation problem (2.13) with $w_0 = Q(\bar{y}) - \hat{\eta} - f^\top(\bar{y} - \hat{y})$ and $w = B(\bar{y} - \hat{y})$. We refer to $g(\pi, \pi_0)$ with this choice of (w, w_0) as the *Conforti-Wolsey (CW) pseudonorm.* We remark that the coefficients of the CW pseudonorm change as $(\hat{y}, \hat{\eta})$ changes; thus, unlike other normalization functions presented so far, one should update the normalization constraint for each new point being separated.

2.4 Guided Projections Algorithm for ℓ_p-deepest Cuts

The reformulations previously presented in §2.3.1 may be used to produce ℓ_p-deepest cuts using LP/QP blackbox solvers. However, unless the subproblem happens to be small or can be decomposed into smaller subproblems, exploiting the combinatorial structure which otherwise would be present in classical Benders subproblems is not straightforward. Since exploiting combinatorial structure often produces an order of magnitude in speed-up for the BD algorithm, we propose in this section a specialized iterative algorithm that produces or approximates ℓ_p-deepest cuts through a series of projections guided by classical Benders cuts.

As noted in Proposition 2.1, we can express $\mathcal{E}$ in terms of the classical optimality and feasibility cuts, which means deepest cuts can be represented as combination of these cuts. Provided that there exists an oracle for producing classical cuts efficiently, we show how these cuts can be used for producing deepest cuts. Recall from Theorem 2.1 that producing an ℓ_p-deepest cut is equivalent to finding the ℓ_q-projection of the incumbent point $(\hat{y}, \hat{\eta})$ onto the epigraph $\mathcal{E}$. Instead of finding this projection directly, we can iteratively *guide* the projection by moving from the incumbent point to its projection on the epigraph by successively identifing facets of the epigraph using classical Benders dual subproblems; thus, we call this iterative procedure the *Guided Projections Algorithm* (GPA). GPA separates the projection problem (2.10) into two simpler problems in a row generation manner: (a) producing a new facet of $\mathcal{E}$ using classical Benders subproblems, and (b) projecting the incumbent point $(\hat{y}, \hat{\eta})$ onto these half-spaces.

Algorithm 3 provides an overview of GPA. Starting with $\mathcal{C}^{(0)}$ as an initial approximation of $\mathcal{E}$, GPA first projects $(\hat{y}, \hat{\eta})$ onto $\mathcal{C}^{(0)}$ to obtain the intermediate projection $(\tilde{y}^{(0)}, \tilde{\eta}^{(0)})$. GPA then produces a classical cut by solving DSP evaluated at $\tilde{y}^{(0)}$, and adds the cut to $\mathcal{C}^{(0)}$ to obtain $\mathcal{C}^{(1)}$. GPA then iterates by updating the epigraph approximation and producing new dual solutions. Note that at iteration h, given the intermediate projection $(\tilde{y}^{(h)}, \tilde{\eta}^{(h)})$, the values of $\|(\hat{y}, \hat{\eta}) - (\tilde{y}^{(h)}, \tilde{\eta}^{(h)})\|_q$ and $\|(\hat{y}, \hat{\eta}) - (\tilde{y}^{(h)}, Q(\tilde{y}^{(h)}))\|_q$ provide lower- and upper-bounds on $d^*_{\ell p}(\hat{y}, \hat{\eta})$ (i.e., the ℓ_q distance from $(\hat{y}, \hat{\eta})$ to $\mathcal{E}$), respectively. As the algorithm iterates and $\mathcal{C}^{(h)}$ becomes a tighter approximation of $\mathcal{E}$, these bounds converge; thus, the intermediate projections converge to the projection of $(\hat{y}, \hat{\eta})$ onto $\mathcal{E}$.

Figure 2.4 illustrates this procedure in our 1-dimensional example, where $(\tilde{y}^{(0)}, \tilde{\eta}^{(0)}) = (\hat{y}, \hat{\eta})$ is the point to be separated, $(\tilde{y}^{(2)}, \tilde{\eta}^{(2)})$ is its projection onto $\mathcal{E}$ (dark polygon), and $(u^{(1)}, 1)$ corresponds to the deepest cut. To find this projection and the deepest cut, we first obtain the optimality cut defined by $u^{(0)}$ by solving a classical DSP induced by $\tilde{y}^{(0)} = \hat{y}$. We then find the projection of $(\hat{y}, \hat{\eta})$ onto this half-space to obtain $(\tilde{y}^{(1)}, \tilde{\eta}^{(1)})$. Repeating this procedure for another step produces dual solution $(u^{(1)}, 1)$ and primal solution $(\tilde{y}^{(2)}, \tilde{\eta}^{(2)})$.

Once GPA converges, we obtain a sequence of dual solutions $\{(\pi^{(h)}, \pi_0^{(h)})\}$, and by construction, each one of these solutions can be used for separating $(\hat{y}, \hat{\eta})$ from $\mathcal{E}$. Note that the dual

Algorithm 3 Guided Projections Algorithm

```
STEP 0: Initialize $\mathcal{C}^{(0)} \leftarrow \emptyset$, $h \leftarrow 0$;
while not converged do
    STEP 1 (Projection): If $h = 0$, set $(\tilde{y}^{(h)}, \tilde{\eta}^{(h)}) = (\hat{y}, \hat{\eta})$. Otherwise, find the $\ell_q$-projection of $(\hat{y}, \hat{\eta})$ onto $\mathcal{C}^{(h)}$, and let $(\tilde{y}^{(h)}, \tilde{\eta}^{(h)})$ be this projection.
        $$(\tilde{y}^{(h)}, \tilde{\eta}^{(h)}) \leftarrow \underset{(y,\eta)\in\mathcal{C}^{(h)}}{\operatorname{argmin}} \quad \|(y-\hat{y}, \eta-\hat{\eta})\|_q \qquad (2.19)$$
    STEP 2 (Cut Generation): Solve the following classical DSP:
        $$[\text{DSP}] \quad \tilde{Q}(\tilde{y}^{(h)}) = \max\left\{u^\top(b - B\tilde{y}^{(h)}) : u^\top A \le c, u \ge 0\right\}.$$
    if DSP is bounded then
        Let $u^{(h)}$ be an optimal solution to DSP and set $(\pi^{(h)}, \pi_0^{(h)}) \leftarrow (u^{(h)}, 1)$
    else
        Let $v^{(h)}$ be an optimal ray to DSP and set $(\pi^{(h)}, \pi_0^{(h)}) \leftarrow (v^{(h)}, 0)$
    end if
    STEP 3 (Epigraph Approximation): $\mathcal{C}^{(h+1)} \leftarrow \mathcal{C}^{(h)} \cap \mathcal{H}(\pi^{(h)}, \pi_0^{(h)})$
    $h \leftarrow h + 1$
end while
```

solution associated with the deepest cut might not be one of these solutions, but a convex combination of them. Therefore, one can choose to add one or more of the cuts produced by GPA to the BD master problem. Note that GPA guarantees convergence of the BD algorithm, even if we terminate GPA before converging to the true projection. This is because the cut obtained in the first iteration of this algorithm is the cut that one would produce using the classical BD algorithm.

We remark that $\mathcal{C}^{(0)}$ need not be initialized with an empty set, and the algorithm may benefit from initializing $\mathcal{C}^{(0)}$ with a few simple constraints. For instance, one can use the cuts generated before (e.g., in separating master solutions in the previous iterations of BD) to initialize $\mathcal{C}^{(0)}$. Also, in many cases the constraints that define Y are also feasibility cuts. For instance, the non-negativity constraints $y \ge 0$ are necessary to ensure boundedness in DSP for facility location-type problems. Therefore, one can add such constraints to $\mathcal{C}^{(0)}$ as well.

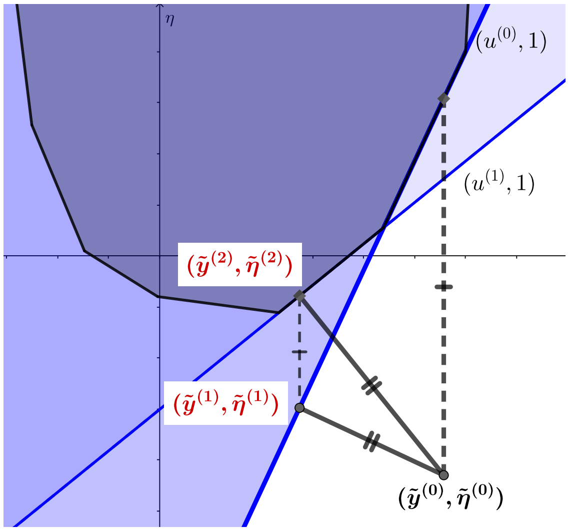

η
$(u^{(0)}, 1)$
$(u^{(1)}, 1)$
$(\tilde{\boldsymbol{y}}^{(2)}, \tilde{\boldsymbol{\eta}}^{(2)})$
$(\tilde{\boldsymbol{y}}^{(1)}, \tilde{\boldsymbol{\eta}}^{(1)})$
$(\tilde{\boldsymbol{y}}^{(0)}, \tilde{\boldsymbol{\eta}}^{(0)})$

Figure 2.4: A geometric view of Guided Projections Algorithm.

2.5 Computational Experiments

In this section, compare the performance of deepest cuts and other variants of Benders cuts on instances of the capacitated facility location problem (CFLP), whose structure is known to be well-suited for BD (Fischetti et al. 2016, 2017). General details for the effective implementation of the BD algorithm are given in Appendix A.2.

2.5.1 Benchmark Instances

We used instances of the CFLP as a testbed for evaluating the performance of the BD algorithm with different choices of distance functions. Facility location problems lie at the heart of network design and planning, and arise naturally in a wide range of applications such as supply chain management, telecommunications systems, urban transportation planning, health care systems and humanitarian logistics to count a few (see e.g., Drezner and Hamacher 2001). Given a set of customers and a set of potential locations for the facilities, CFLP in its simplest form as formulated below, consists of determining which facilities to open and how to assign customers to opened facilities to minimize cost, i.e.,

$$
\begin{aligned}
\min \quad & \sum_{l=1}^{k}\sum_{j=1}^{n} x_{lj} d_l c_{lj} + \sum_{j=1}^{n} f_j y_j && && (2.20)\\
\text{s.t.} \quad & \sum_{j=1}^{n} x_{lj} \geq 1 && \forall l && (2.21)\\
& \sum_{l=1}^{k} x_{lj} d_l \leq s_j y_j && \forall j && (2.22)\\
& x_{lj} \leq y_j && \forall l, j && (2.23)\\
& x \geq 0, y \in Y, && && (2.24)
\end{aligned}
$$

where f_j and s_j are respectively the installation cost and capacity of facility $j = 1, \dots, n$; d_l is the demand of customer $l = 1, \dots, k$; c_{lj} is the cost of serving one unit of demand from customer l using facility j; and $Y = \{y \in \{0,1\}^n : \sum_{l=1}^{k} d_l \leq \sum_{j=1}^{n} s_j y_j\}$ is the domain of the y variables.

We used two sets of benchmark instances from the literature:

CAP: The famous CAP data set from the OR-Library (Beasley 2021) consists of 24 small instances with $k = 50$ customers and $n \in \{16, 25, 50\}$ facilities, and 12 large instances with $n = 100$ facilities and $k = 1000$ customers. The instances are denoted CAPx1–CAPx4, where $x \in \{6, 7, 9, 10, 12, 13\}$ for the small instances and $x \in \{a, b, c\}$ for the large instances.

CST: These are instances that we randomly generated following the procedure proposed by Cornuéjols et al. (1991). We denote each instance by tuple (n, k, r), where (n, k) pairs were selected from {(50,50), (50,100), (100,100), (100,200), (100,500), (500,500), (100,1000), (200,1000), (500,1000), (1000,1000)} and the scaling factor r was selected from $\{5, 10, 15, 20\}$. For each choice of (n, k, r) we randomly generated 4 instances as follows. For each facility $j \in \{1, \dots, n\}$, we randomly drew s_j and f_j from $U[10, 160]$ and $U[0, 90] + U[100, 110]\sqrt{s_j}$, respectively, where $U[a, b]$ represents the uniform distribution on $[a, b]$. For each customer

$l \in \{1, \ldots, k\}$, we randomly drew d_l from $U[5, 35]$. Finally, we scaled the facility capacities using parameter r such that $r = \frac{\sum_{j=1}^{n} s_j}{\sum_{l=1}^{k} d_l}$. To compute the allocation costs, we placed the customers and facilities in a unit square uniformly at random, and set c_{lj} to 10 times the Euclidean distance of facility j from customer l.

We remark that, whenever a core point $\bar{y}$ was needed (e.g., for the MWP and CW pseudonorms) we chose the core point by setting $\bar{y}_j = \frac{1}{r} + \epsilon$ for each j, where $\epsilon = 10^{-3}$ and $r = \frac{\sum_{j=1}^{n} s_j}{\sum_{l=1}^{k} d_l}$. The same core point was used for choosing β for scaling the η variable as described in §A.2.2.

2.5.2 Numerical Results for Standard Reformulations

We start by presenting numerical results for distance-based BD algorithms where the separation problems are reformulated as LP/QP programs using standard transformation techniques presented in §2.3 and solved using a solver without exploiting any combinatorial structures in the problem instances. We considered nine cut selection strategies:

- $\boldsymbol{\ell_1}$ and $\boldsymbol{\ell_\infty}$: Deepest cuts with ℓ_1 and ℓ_∞ norms on the coefficients. Separation problems are transformed into linear programs according to the transformations given in §2.3.1.

- $\boldsymbol{\ell_2}$ and $\boldsymbol{\ell_4}$: Deepest cuts with ℓ_2 and ℓ_4 norms on the coefficients. Separation problems are transformed into quadratic programs according to the transformations given in §2.3.1.

- ***MIS***: Distance function with linear normalization function $g(\pi, \pi_0) = w^\top \pi + w_0 \pi_0$, with (w, w_0) in the MIS subproblem chosen according to the default setting suggested by Fischetti et al. (2010), that is $w_0 = 1$, $w_i = 0$ if the i'th row of B is all zeros and $w_i = 1$ otherwise.

- $\boldsymbol{R\ell_1}$: Distance function with linear normalization function $g(\pi, \pi_0) = w^\top \pi + w_0 \pi_0$, with (w, w_0) in the MIS subproblem chosen according to $R\ell_1$ as described in §2.3.2.2.

- ***MWP***: Distance function with linear normalization function $g(\pi, \pi_0) = w^\top \pi + w_0 \pi_0$, with (w, w_0) in the MIS subproblem chosen according to MWP as described in §2.3.2.3.
- ***CW***: Distance function with linear normalization function $g(\pi, \pi_0) = w^\top \pi + w_0 \pi_0$, with (w, w_0) in the MIS subproblem chosen according to CW as described in §2.3.2.4.
- ***CB***: Classical Benders cuts which correspond to setting $g(\pi, \pi_0) = \pi_0$.

Table 2.1 compares the computational performance of BD with different choices of the normalization functions across CAP and CST instances. Each row in Table 2.1 corresponds to the average performance metrics over four instances. The CAP instances are formatted as "CAPx* (n, k)". For instance CAP6* (16,50) corresponds to CAP instances CAP61–CAP64, which contain $n = 16$ facilities and $k = 50$ customers. The CST instances are formatted as "CST (n, k, r)", which correspond to four randomly generated instances with n facilities, k customers, and capacity/demand scaling factor r. In these experiments, we have used the small and moderately sized instances of these data sets (i.e., with at most $k = 200$ customers) to showcase the quality of the cuts produced based on each normalization function.

The results provided in Table 2.1 highlight the effectiveness of ℓ_p-deepest cuts in reducing the number of cuts compared to CB and MIS cuts with different choices of the normalization coefficients. This is particularly pronounced with $p = 1$ and $p = 2$ in CAP instances, while all choices of $p \in \{1, 2, 4, \infty\}$ perform well in CST instances. A surprising result is that the default choices of normalization coefficients in MIS suffers from producing weak cuts in the CAP instances, but performs well in the CST instances. On the other hand, when the normalization coefficients are chosen according to MWP, CW or $R\ell_1$ pseudonorms, MIS subproblems perform significantly better. In fact, the best performance in the CST instances across all normalization functions is achieved when $R\ell_1$ or CW pseudonorms are used.

In terms of computation time, MWP, CW, $R\ell_1$, and CB yield the best overall computation time in CAP instances, followed by ℓ_1- and ℓ_∞-deepest cuts in second place. This is because although deepest cuts are often more effective (particularly ℓ_1-deepest cuts in the `CAP` instances), generating these cuts is often computationally more expensive than MWP, CW, $R\ell_1$, and CB, which require solving simpler LPs. Similar observations can be made in `CST` instances, except ℓ_∞-deepest cuts along with $R\ell_1$ and CW cuts achieve the best computation times. As expected, since ℓ_2- and ℓ_4-deepest cuts require solving relatively large quadratic programs, their computation times do not justify the effectiveness of the cuts. These observations necessitate resorting to a method such as GPA capable of exploiting the structural properties of the problem.

2.5.3 Numerical Results for the Guided Projections Algorithm

We now present computational results for the BD algorithm when ℓ_p-deepest cuts are generated using the Guided Projections Algorithm 3. Our goal is to assess the effectiveness of GPA in exploiting the combinatorial structure of a problem while producing ℓ_p-deepest cuts for different choices of p. The capacitated facility location problem, which we use in our tests, (2.20)–(2.24) exhibits combinatorial structures that can be exploited when solving classical Benders subproblems. Since our intent is to isolate the effect of deepest cuts, we do not employ stabilization techniques in our numerical tests, which would complicate the dynamics and make conclusions harder to draw. A list of such techniques can be found in (Fischetti et al. 2017) and a comprehensive study of their impacts in conjunction with deepest cuts is left to future work.

For a given (fractional) solution $\hat{y}$, we may derive a Benders cut efficiently by solving the subproblem in the primal space as follows. First, note that constraints (2.22) can be treated as bounds on the primal variables x. Consequently, it suffices to update these bounds based on values of $\hat{y}$ and reduce the number of constraints in the primal subproblem from $n+k+nk$ to $n+k$ constraints. The resulting problem is a transportation problem, which can be solved

	Cuts								Time (sec.)							
Instance	CB	MIS	MWP	CW	$R\ell_1$	ℓ_1	ℓ_∞	ℓ_2	ℓ_4	CB	MIS	MWP	CW	$R\ell_1$	ℓ_1	ℓ_∞	ℓ_2	ℓ_4
`CAP6*` (16,50)	17.5	48.0	10.3	30.0	18.8	7.3	13.8	10.3	54.8	0.04	0.26	0.08	0.72	0.08	0.20	0.08	4.75	19.11
`CAP7*` (16,50)	15.0	383.5	10.3	13.3	12.3	5.5	10.3	8.3	34.3	0.02	0.98	0.05	0.15	0.03	0.11	0.05	3.57	10.28
`CAP9*` (25,50)	44.8	482.3	21.5	30.0	64.5	8.3	16.8	9.8	136.3	0.14	2.44	0.20	0.59	0.36	0.56	0.26	7.99	84.39
`CAP10*` (25,50)	29.3	3363.0	11.5	19.8	43.0	7.3	24.8	11.5	82.8	0.05	127.98	0.10	0.32	0.16	0.28	0.17	9.44	39.32
`CAP12*` (50,50)	125.8	5449.5	36.0	37.8	120.8	14.0	88.3	21.3	21.0	0.92	384.22	0.90	1.94	1.84	2.97	3.96	84.33	13.05
`CAP13*` (50,50)	103.3	6626.0	23.0	29.3	61.8	9.0	60.3	27.8	39.8	0.36	384.14	0.52	1.18	0.64	1.45	1.23	85.93	19.89
Mean	55.9	2725.4	18.8	26.7	53.5	8.6	35.7	14.8	61.5	0.25	150.00	0.31	0.82	0.52	0.93	0.96	32.67	31.01
`CST` (50,50,5)	17.8	12.0	19.0	5.8	5.0	4.3	4.0	4.3	6.5	0.09	0.23	0.50	0.23	0.14	0.61	0.34	6.19	3.32
`CST` (50,50,10)	19.3	11.8	15.0	7.5	3.5	4.0	3.5	4.0	6.8	0.18	0.70	0.48	0.28	0.23	0.54	0.25	5.77	3.94
`CST` (50,50,15)	5.5	5.5	15.8	3.5	2.8	3.0	1.8	2.8	3.8	0.03	0.25	0.59	0.25	0.28	0.34	0.18	3.29	1.96
`CST` (50,50,20)	9.0	8.0	18.5	7.5	4.8	5.5	4.5	5.5	7.5	0.07	0.47	0.77	0.31	0.44	0.87	0.34	8.20	4.52
`CST` (50,100,5)	20.8	14.3	20.3	6.5	6.0	7.0	4.8	6.0	8.8	0.34	1.82	1.45	0.70	0.61	2.54	0.85	3.03	8.00
`CST` (50,100,10)	28.0	17.8	18.3	6.5	4.3	9.3	6.3	9.3	7.0	0.61	4.38	1.89	0.91	0.85	3.79	1.55	4.55	6.85
`CST` (50,100,15)	7.5	4.8	13.5	2.5	2.8	2.8	2.0	3.0	4.0	0.16	0.75	2.11	0.68	0.66	1.01	0.69	1.40	4.36
`CST` (50,100,20)	20.3	13.0	25.0	7.0	5.5	7.5	5.8	7.5	9.8	0.52	5.83	4.35	1.36	2.53	4.46	1.46	3.85	10.06
`CST` (100,100,5)	8.3	7.3	12.8	5.0	1.8	3.3	3.0	3.3	8.5	0.20	0.79	1.75	1.64	1.38	4.16	1.39	7.11	12.99
`CST` (100,100,10)	33.5	13.0	19.3	7.8	6.3	6.5	6.8	7.5	11.0	2.01	1.19	4.29	1.70	3.35	10.49	3.22	14.75	17.23
`CST` (100,100,15)	43.5	17.3	26.5	8.3	5.3	11.3	7.0	7.5	10.5	3.64	22.30	8.49	3.34	4.62	25.56	3.35	13.49	17.37
`CST` (100,100,20)	38.8	13.5	25.0	7.0	5.8	6.5	5.5	9.0	8.3	4.45	15.57	11.32	2.48	5.14	14.43	3.16	15.45	12.02
`CST` (100,200,5)	12.8	7.5	10.5	5.8	2.8	4.3	3.3	4.0	8.8	1.44	3.42	19.61	11.25	6.00	35.16	15.89	24.47	88.24
`CST` (100,200,10)	66.5	24.0	24.3	7.3	7.0	11.5	7.0	12.8	11.3	21.34	126.52	25.57	9.76	12.85	57.48	11.51	61.98	102.33
`CST` (100,200,15)	46.3	37.7	30.3	5.8	8.3	12.7	9.0	11.5	16.5	35.88	520.30	43.14	8.28	22.83	163.59	14.81	45.43	141.73
`CST` (100,200,20)	62.7	30.8	25.3	5.5	7.0	14.5	8.0	12.7	17.3	25.91	450.67	46.81	7.04	26.33	125.50	12.02	58.17	167.40
Mean	27.5	14.9	19.9	6.2	4.9	7.1	5.1	6.9	9.1	6.05	72.20	10.82	3.14	5.51	28.16	4.44	17.32	37.65

Table 2.1: Performance comparison of BD with different distance functions on `CAP` and `CST` instances. Separation problems are solved using a LP/QP solver without exploiting combinatorial structure.

efficiently for large instances using specialized algorithms. In our implementation, however, we have used `CPLEX` for solving the transportation problems since it benefits from better warm starting. Let $\hat{u}^{\mathrm{D}}$ be the optimal dual solution associated with the demand constraints (2.21) obtained by the solver. Given $\hat{u}^{\mathrm{D}}$, as noted by several authors (see e.g., Cornuéjols et al. 1991, Fischetti et al. 2016), the Benders cut takes the form

$$\eta \geq \sum_{l=1}^{k} \hat{u}_l^{\mathrm{D}} + \sum_{j=1}^{n} \left(f_j + \kappa_j(\hat{u}^{\mathrm{D}})\right) y_j, \tag{2.25}$$

where $\kappa_j(\hat{u}^{\mathrm{D}})$ is the optimal value of the continuous knapsack problem

$$\kappa_j(\hat{u}^{\mathrm{D}}) = \min \left\{ \sum_{l=1}^{k} (d_l c_{lj} - \hat{u}_l^{\mathrm{D}}) \alpha_l : \sum_{l=1}^{k} \alpha_l d_l \leq s_j, \alpha \in [0,1]^k \right\},$$

which can be solved efficiently in $\mathcal{O}(k)$ time by finding the weighted median of the ratios $\{c_{lj} - \frac{\hat{u}_l^{\mathrm{D}}}{d_l}\}$ using the procedure given in Balas and Zemel (1980).

Each iteration of GPA (Algorithm 3) for producing an ℓ_p-deepest cut involves finding an ℓ_q-projection (with q such that $\frac{1}{p} + \frac{1}{q} = 1$), and producing a classical cut using (2.25). We terminate Algorithm 3 when the ℓ_q-projection $(\tilde{y}^{(h)}, \tilde{\eta}^{(h)})$ at iteration h of GPA is sufficiently close to $\mathcal{E}$ as mentioned in §2.4, or after 10 iterations. We considered three choices for ℓ_p-deepest cuts with $p \in \{1, 2, \infty\}$. We also considered the option of switching to classical cuts after the optimality gap falls below a certain threshold (5% in our experiments) to validate the intuitions provided in §2.1.4 which suggest that deepest cuts are more effective at the early iterations of the BD algorithm. Table 2.2 compares the computational results of six variants of BD equipped with GPA for producing ℓ_p-deepest cuts (with or without switching to classical cuts) with classical Benders (CB) across the `CAP` and `CST` benchmark instances. To showcase the benefits of exploiting the structural properties of the problem at a large scale, we have included the large instances of these data sets (i.e. up to 1000 facilities and customers) in Table 2.2. For brevity, we have excluded the results for smaller instances, but we highlight that GPA is able to reduce the runtime of BD with ℓ_∞-, ℓ_1-, and ℓ_2-deepest cuts by more than 4, 16, and 35 times on average, respectively, compared to the results reported

	Cuts							Time (sec.)						
		Without switch			Switch at 5%				Without switch			Switch at 5%		
Instance	CB	ℓ_1	ℓ_∞	ℓ_2	ℓ_1	ℓ_∞	ℓ_2	CB	ℓ_1	ℓ_∞	ℓ_2	ℓ_1	ℓ_∞	ℓ_2
CAPa* (100,1000)	105.8	48.8	62.3	46.8	41.5	51.0	44.8	18.66	16.32	15.98	13.98	11.40	14.59	13.39
CAPb* (100,1000)	413.3	155.8	104.5	95.0	115.8	106.8	101.0	52.42	35.72	18.13	19.75	24.90	17.91	21.21
CAPc* (100,1000)	320.8	127.3	97.0	91.8	107.5	92.0	81.5	37.65	27.57	14.90	15.84	19.14	14.36	14.50
Mean	280.0	110.6	87.9	77.8	88.3	83.3	75.8	36.24	26.54	16.33	16.52	18.48	15.62	16.36
CST (100,500,5)	59.0	34.3	20.8	9.5	19.3	13.0	13.8	3.57	2.23	1.31	1.01	1.68	1.10	1.29
CST (100,500,10)	68.5	54.3	36.5	38.8	27.8	30.3	32.0	4.51	3.48	2.67	3.46	2.29	2.53	3.01
CST (100,500,15)	87.5	62.8	29.8	54.5	36.3	19.8	22.8	6.29	4.13	2.65	4.91	2.90	2.11	2.57
CST (100,500,20)	65.3	56.3	40.3	32.0	29.3	23.5	30.8	5.32	4.35	3.12	3.29	2.68	2.51	3.26
CST (500,500,5)	28.8	8.5	8.5	8.8	6.0	5.5	6.8	9.30	4.77	4.53	4.82	4.46	4.08	4.68
CST (500,500,10)	69.3	57.5	63.3	66.0	42.0	28.0	53.8	23.72	19.44	17.60	24.12	15.98	12.57	22.17
CST (500,500,15)	75.7	30.5	38.8	27.3	46.8	7.0	46.3	26.03	14.85	14.89	15.03	20.45	7.46	22.53
CST (500,500,20)	127.5	53.5	40.3	43.3	10.3	27.8	8.8	50.84	23.12	18.67	23.20	9.55	15.57	11.39
CST (100,1000,5)	88.5	36.0	24.3	42.5	26.0	22.0	31.3	11.62	5.23	4.07	6.58	4.49	3.74	5.62
CST (100,1000,10)	125.3	83.3	57.3	48.3	34.0	43.3	44.3	19.40	12.24	10.53	9.67	7.15	9.29	10.14
CST (100,1000,15)	81.8	65.3	66.8	65.5	52.8	60.0	54.8	16.59	12.82	13.54	14.69	11.13	13.02	12.49
CST (100,1000,20)	78.5	64.3	54.0	56.8	47.0	49.8	42.5	19.93	15.67	13.32	13.99	11.39	13.38	12.28
Mean	79.6	50.5	40.0	41.1	31.4	27.5	32.3	16.43	10.20	8.91	10.40	7.84	7.28	9.29
CST (200,1000,5)	56.8	8.3	9.8	9.5	7.8	5.3	5.5	14.44	4.44	4.72	4.78	4.49	3.82	3.79
CST (200,1000,10)	180.3	31.8	41.5	62.0	21.8	37.5	32.8	59.96	14.25	17.86	22.84	12.46	17.94	16.23
CST (200,1000,15)	204.5	83.8	84.3	78.5	52.3	50.3	65.0	70.66	29.23	32.55	34.69	26.40	26.96	33.84
CST (200,1000,20)	170.5	127.3	78.8	76.8	60.8	72.0	105.8	73.83	44.09	35.18	37.34	34.05	39.03	51.87
CST (500,1000,5)	29.3	34.0	14.5	18.0	21.3	4.5	22.3	22.71	25.58	14.98	18.16	20.53	10.30	20.66
CST (500,1000,10)	131.5	37.0	43.3	45.3	40.3	22.0	31.3	122.92	37.28	37.71	45.95	42.04	31.75	38.27
CST (500,1000,15)	127.5	31.5	40.5	18.8	23.3	22.0	26.3	133.71	42.07	48.58	36.23	38.34	39.83	46.70
CST (500,1000,20)	252.0	27.3	69.5	115.8	20.5	39.3	41.8	389.08	48.79	94.76	143.77	45.71	74.17	85.31
CST (1000,1000,5)	3.5	2.0	2.0	2.0	2.0	2.0	2.0	22.82	15.66	15.48	16.07	15.43	14.92	15.27
CST (1000,1000,10)	46.5	5.0	5.0	4.5	4.0	3.0	3.5	97.55	31.17	29.99	27.61	29.53	24.99	25.98
CST (1000,1000,15)	204.3	59.5	67.8	88.8	48.3	27.0	65.5	684.32	171.47	155.73	214.40	144.51	104.71	193.88
CST (1000,1000,20)	103.0	12.3	8.0	9.8	10.0	6.3	7.0	280.88	79.79	78.10	81.37	74.48	77.29	80.41
Mean	125.8	38.3	38.7	44.1	26.0	24.3	34.0	164.41	45.32	47.14	56.94	40.66	38.81	51.01

Table 2.2: Computational performance of BD algorithm using GPA for producing deepest cuts on large instances of CAT and CST data sets, where cuts are generated by exploiting the combinatorial structures of the instances.

in Table 2.1 for these norms.

Comparing GPA and CB, we observe that BD equipped with GPA with all choices of $p \in \{1, 2, \infty\}$ has reduced the number of cuts by more than three times on average compared to BD with CB cuts in the large instances of both `CAP` and `CST` data sets. The results further confirm effectiveness of deepest cuts in closing the optimality gap, despite the classical cuts in the CFLP (when derived using (2.25)) often being deemed sufficiently effective in the literature (Fischetti et al. 2016). More importantly, we observe that the running time of BD with GPA is less than half that of BD with CB cuts, even when Euclidean (ℓ_2) deepest cuts are produced. This highlights the effectiveness of GPA in separating the projection and cut generation steps while producing cuts, even when the projection step requires solving a quadratic program. Finally, we observe that employing deepest cuts at the early iterations of the BD algorithm and switching to classical cuts after a certain threshold (5% optimality gap in our tests) has a positive effect on the overall number of cuts produced and on the computation time, which supports the theoretical insights provided in §2.1.4.

2.6 Conclusions

In this chapter, we proposed and analyzed theoretically and computationally a new method for selecting Benders cuts, aimed at improving the effectiveness of the cuts in closing the gap and reducing the running time of the BD algorithm. Our technique is based on generating Benders cuts that explicitly take cut depth into account. As a measure of cut depth, we considered Euclidean distance from the master solution to the candidate cuts, and then extended this measure to general ℓ_p-norms. We provided a comprehensive study of deepest cuts and unveiled their properties from a primal perspective. We showed that producing an ℓ_p-deepest cut is equivalent to finding an ℓ_q-projection of the point being separated onto the epigraph of the original problem. We also showed how the separation problems can be solved as linear or quadratic programs. Leveraging the duality between ℓ_p-deepest cuts and

ℓ_q-projections, we introduced our Guided Projections Algorithm for producing ℓ_p-deepest cuts in a way that can exploit the combinatorial structure of problem instances.

From a theoretical perspective, we generalized our notion of distance by defining what we call a Benders distance function, and developed a notion of monotonicity which allows these functions to be treated as distance functions despite the fact that they do not necessarily satisfy the axioms of metrics. Then, we illustrated the connection of such distance functions to some well-known cut selection strategies. Specifically, we established the connection to MIS cuts, and provided three novel ways of choosing the normalization coefficients in the MIS subproblem, that connect our distance functions to the Magnanti-Wong procedure for producing Pareto-optimal cuts, as well as the Conforti-Wolsey procedure for producing facet-defining cuts.

Our computational experiments on CFLP instances showed the benefits of deepest cuts and other distance-based cuts, particularly when generated using GPA, in decreasing the number of cuts as well as the runtime of the BD algorithm. Besides the theoretical insights, our results showed that deepest cuts are effective in speeding up convergence of BD. Our results also illustrated the benefits of choosing specific normalization coefficients in the MIS problems based on parameters of the problem instances.

Chapter 3

Benders Decomposition for Profit Maximizing Hub Location Problems with Capacity Allocation

In this chapter[1], we consider revenue management decisions within hub location problems and determine how to allocate available capacities of hubs to demand of commodities from different market segments. *The profit maximizing capacitated hub location problems* introduced in this study seek to find an optimal hub network structure, maximizing total profit to provide services to a set of commodities while considering the design cost of the network. The decisions to be made are the optimal number and locations of hubs, allocation of demand nodes to these hubs, and the optimal routes of flow of different classes of commodities that

[1] **Statement of collaboration:** The following is the summary of a joint work with my coauthors Gita Taherkhani (Department of Information Systems and Supply Chain Management, Quinlan School of Business, Loyola University Chicago, Chicago, IL. Email: gtaherkhani@luc.edu) and Sibel A. Alumur (Department of Management Sciences, University of Waterloo, Waterloo, Ontario, Canada. Email: sibel.alumur@uwaterloo.ca). This chapter contains materials from (Taherkhani et al. 2020). All authors were involved in writing the papers and my main contributions to this study was in developing and implementing the models and algorithms.
Reprint permission acquired from Transportation Science with License ID 1225865-1.

are selected to be served. We consider a hub location problem with multiple assignments, and allow a path of an O-D pair to pass through at most two hubs. Direct connections between non-hub nodes are not allowed; all commodities must be routed via a set of hubs. Demand is segmented into different classes and revenue is obtained from satisfying demands for the commodities of each class. The model is to decide how much demand to serve from each class considering the available capacity. Each demand class from each commodity can be partially satisfied.

This chapter is organized as follows. We introduce the notation and formulate the mixed-integer linear programming models in §3.1. §3.2 embodies the main contribution of the study which is the proposed Benders decomposition algorithm and our introduction of several features that improve the convergence and efficiency for the deterministic version of the problem. We perform extensive computational experiments in §3.3 to test our mathematical models and evaluate our algorithms. The chapter is concluded in §3.4 with concluding remarks. Finally, technical details and supplementary numerical results are provided in Appendix B.

3.1 Mathematical Formulation

This section first introduces the notation and then presents mathematical formulation for the deterministic version of the problem.

3.1.1 Notation

Let $G = (N, A)$ be a complete digraph, where N is the set of nodes and A is the set of arcs. Let $H \subseteq N$ be the set of potential hub locations (in our setting we simply assume $H = N$), and let f_i and Γ_i denote the installation cost and the available capacity of a hub located at node $i \in H$, respectively. $K \subseteq N \times N$ represents the set of commodities whose origin and destination points belong to N (in our setting we assume $K = N \times N$). Demand of commodities are segmented into M classes. For each commodity $k \in K$ of class $m \in M$, w_k^m represents the amount of commodity k of class m to be routed from the origin $o(k) \in N$ to the destination $d(k) \in N$. Satisfying a unit commodity $k \in K$ of class $m \in M$ produces a per unit revenue of r_k^m.

The transportation cost from node $i \in N$ to node $j \in N$ is defined as $c_{ij} = \gamma d_{ij}$, where d_{ij} denotes the distance from node i to node j, and γ is the resource cost per unit distance. Distances are assumed to satisfy the triangle inequality. Each path of an O-D pair contains at least one and at most two hubs. Thus, paths are of the form $(o(k), i, j, d(k))$, where $(i, j) \in H \times H$ represents the ordered pair of hubs to which $o(k)$ and $d(k)$ are allocated, respectively. The transportation cost of routing one unit of commodity k along path $(o(k), i, j, d(k))$ can be calculated by $C_{ijk} = \chi c_{o(k)i} + \alpha c_{ij} + \delta c_{jd(k)}$, where χ, α, δ represent the collection, transfer, and distribution costs, respectively. The economies of scale between hubs is reflected by assuming $\alpha < \chi$ and $\alpha < \delta$.

Before presenting the mathematical formulation, we note the following properties of the capacitated hub location problems.

Property 3.1. *(Boland et al. 2004) In any optimal solution, a commodity can be routed via a path containing two distinct hubs only if it is not cheaper to do so using one of the hubs.*

Property 3.2. *(Contreras et al. 2011a) In any optimal solution, among the paths $(o(k), i, j, d(k))$ and $(o(k), j, i, d(k))$, commodity $k \in K$ uses at most one them and it is the one with the lower transportation cost.*

Consequently, instead of defining hub arc a as an ordered set $a = (a_1, a_2) \in H \times H$, we define a hub arc as $a = \{a_1, a_2\}$ where $a_1, a_2 \in H$. For simplicity, when $a_1 = a_2$ (i.e. (a_1, a_2) is a loop) we denote the loop as $a = \{a_1\}$. We use the notation $|a|$ as the number of hubs that form the hub arc a, and we say $i \in a = \{a_1, a_2\}$ if node i is equal to a_1 or a_2. Additionally, each commodity defines its own set of potential hub arcs. Henceforth, we replace C_{ijk} with $\hat{C}_{ijk} = \min\{C_{ijk}, C_{jik}\}$, and reduce the set of candidate hub arcs for commodity $k \in K$ to A_k defined as

$$A_k = \{(i,j) \in A : i \leq j, \hat{C}_{ijk} \leq \min\{C_{iik}, C_{jjk}\}\}, \tag{3.1}$$

which can be used to reduce the size of the mathematical formulations. In this equation, when $i = j$ we assume that the hub arc is a loop.

3.1.2 Mathematical Model

We first consider the deterministic setting assuming that perfect information on demands and revenues is available. Let y_i equal to 1 if a hub is established at node $i \in H$, and 0 otherwise. Define $Y = \{0,1\}^{|H|}$ as the domain of the y-variables. Moreover, let x^m_{ak} determine the fraction of commodity $k \in K$ of class $m \in M$ that is satisfied through a path with hub arc $a \in A_k$. The *profit maximizing capacitated hub location problem* is then modeled as:

$$\max \quad \sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k}(r^m_k - \hat{C}_{ak})w^m_k x^m_{ak} - \sum_{i\in H} f_i y_i \tag{3.2}$$

$$\text{s.t.} \quad \sum_{a\in A_k} x^m_{ak} \leq 1 \qquad k \in K, m \in M \tag{3.3}$$

$$\sum_{a\in A_k : i\in a} x^m_{ak} \leq y_i \qquad i \in H, k \in K, m \in M \tag{3.4}$$

$$\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k : i\in a} w^m_k x^m_{ak} \leq \Gamma_i y_i \qquad i \in H \tag{3.5}$$

$$x \geq 0, y \in Y. \tag{3.6}$$

The objective function (3.2) represents net profit, which is calculated by subtracting total cost from total revenue. If demand of a commodity from a given class is to be satisfied, then by constraints (3.3) flow must be routed via hub arcs. Note that each demand class from each commodity can be partially satisfied through different paths, hence these constraints are expressed for each $m \in M$. Constraints (3.4) ensure that demand of commodities can be satisfied only through open hubs. Constraints (3.5) restrict capacity on the total incoming flow at a hub via both hub and non-hub nodes. Finally, constraints (3.6) define the non-negative and binary variables, in which $x = \{x_{ak}^m\}$.

Remark 3.1. *For integer values of y, constraints* (3.3) *and* (3.5) *imply constraints* (3.4)*. Hence, constraints* (3.4) *act as valid inequalities for the mathematical model* (3.2)-(3.6)*.*

Since our solution method is based on Benders decomposition, and it is known that Benders decomposition performs better with stronger formulations (Magnanti and Wong 1981), we choose to keep these constraints in our mathematical model.

Note that when revenue from satisfying the commodity $k \in K$ of class $m \in M$ is strictly smaller than the unit transportation cost of routing commodity k along a path containing a hub arc $a \in A_k$, no profit can be obtained from satisfying the demand for commodity (k, m) through that path. Accordingly, the optimal value for the corresponding variable x_{ak}^m can then be set to zero as noted in Property 3.3 below.

Property 3.3. *For every $k \in K$, $m \in M$ and $a \in A_k$, if $r_k^m < \hat{C}_{ak}$, then $x_{ak}^m = 0$ in any optimal solution to* (3.2)-(3.6)*.*

3.2 Benders Decomposition

Motivated by successful implementations of BD to solve hub location problems, in this study, we also develop Benders-based algorithms to solve our problems to optimality. To date, the largest capacitated hub location instances that could be solved optimally contain 300 nodes (Contreras et al. 2012). As we will show through our computational experiments, we are able to solve instances of size 500 nodes, while considering an even more difficult problem setting with generic capacity constraints, multiple demand segments, and a profit maximizing objective function. Moreover, we show that the BD algorithms that we introduce in this study is more efficient in terms of computation time, using memory and in generating stronger cuts than those in the previous literature.

Benders decomposition is well suited for hub location problems especially with multiple allocation structure as the problem can be decomposed into linear subproblems by fixing the integer variables for the location of hubs. In this section, we first present the Benders reformulation of the deterministic model and the Benders decomposition algorithm; we then detail our solution strategies for the subproblems. For details of Benders decomposition, the reader is referred to §1.2.

3.2.1 Benders Reformulation and Algorithm

Given the formulation (3.2)-(3.6), in the Benders reformulation of the problem, the hub location decisions are handled in the master problem and the rest is left to the subproblem. By fixing the values of the y-variables at $y^t \in Y$, we obtain the following linear *primal* *subproblem* (PSP):

$$
\begin{aligned}
\max \quad & \sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k}(r_k^m - \hat{C}_{ak})w_k^m x_{ak}^m && && (3.7)\\
\text{s.t.} \quad & \sum_{a\in A_k} x_{ak}^m \leq 1 && k\in K, m\in M && (3.8)\\
& \sum_{a\in A_k : i\in a} x_{ak}^m \leq y_i^t && i\in H, k\in K, m\in M && (3.9)\\
& \sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k : i\in a} w_k^m x_{ak}^m \leq \Gamma_i y_i^t && i\in H && (3.10)\\
& x_{ak}^m \geq 0 && k\in K, m\in M, a\in A_k. && (3.11)
\end{aligned}
$$

Associating the dual variables α_k^m, u_{ik}^m, and b_i to the constraints (3.8), (3.9), and (3.10), respectively, we have the following *dual subproblem* (DSP):

$$
\begin{aligned}
& \min \sum_{k\in K}\sum_{m\in M}\alpha_k^m + \sum_{i\in H} y_i^t(\Gamma_i b_i + \sum_{k\in K}\sum_{m\in M} u_{ik}^m) && && (3.12)\\
& \text{s.t. } \alpha_k^m + u_{ik}^m + u_{jk}^m + w_k^m(b_i + b_j) \geq (r_k^m - \hat{C}_{ijk})w_k^m && k\in K, m\in M, (i,j)\in A_k : i\neq j && (3.13)\\
& \alpha_k^m + u_{ik}^m + w_k^m b_i \geq (r_k^m - \hat{C}_{iik})w_k^m && k\in K, m\in M, i\in H && (3.14)\\
& \alpha_k^m, u_{ik}^m, b_i \geq 0 && k\in K, m\in M, i\in H && (3.15)
\end{aligned}
$$

Since any choice of $y \in Y$ results in a feasible primal subproblem, the primal and dual subproblems are always feasible and bounded, hence an optimal solution of DSP is one of the extreme points of DSP. Let P denote the polyhedron defined by (3.13)-(3.15), and let $Ex(P)$ be the set of extreme points of P. Note that P and $Ex(P)$ do not depend on y^t. Hence, for any arbitrary y, DSP can be restated as

$$
Q(y) = \min\left\{\sum_{k\in K}\sum_{m\in M}\alpha_k^m + \sum_{i\in H} y_i(\Gamma_i b_i + \sum_{k\in K}\sum_{m\in M} u_{ik}^m) : (\alpha, u, b)\in Ex(P)\right\}. \qquad (3.16)
$$

Let η be an under-estimator of $Q(y)$ representing the overall revenue obtained by satisfying the demand; the Benders *master problem* (MP) can then be formulated as:

$$\max \quad \eta - \sum_{i \in H} f_i y_i \tag{3.17}$$

$$\text{s.t.} \quad \eta \leq \sum_{k \in K} \sum_{m \in M} \alpha_k^m + \sum_{i \in H} y_i (\Gamma_i b_i + \sum_{k \in K} \sum_{m \in M} u_{ik}^m) \qquad (\alpha, u, b) \in Ex(P) \tag{3.18}$$

$$y \in Y. \tag{3.19}$$

Note that MP contains an exponential number of constraints of the form (3.18). Therefore, instead of $Ex(P)$, starting with initial set $\hat{P} \subset Ex(P)$, we iteratively solve a relaxed master problem with $Ex(P)$ replaced with $\hat{P}$ and keep adding new extreme points to $\hat{P}$ by solving dual subproblems, until the optimal solution to MP is found. An overview of the basic BD algorithm is given in Algorithm 4. In this algorithm, UB, LB, t and z_{MP}^t stand for the current upper and lower bounds, the iteration counter, and the optimal solutions obtained from the master problem at iteration t, respectively. ζ is a parameter of the algorithm controlling the convergence threshold (in our computations we set $\zeta = 10^{-6}$).

Algorithm 4 Benders Decomposition for deterministic HLP

```
$UB \leftarrow +\infty, LB \leftarrow -\infty, t \leftarrow 1, \hat{P} \leftarrow \emptyset$
while $LB - UB < \zeta$ do
    SOLVE MP to obtain $y^t$ and $z_{MP}^t$
    $UB \leftarrow z_{MP}^t$
    SOLVE DSP($y^t$) to obtain $(\alpha, u, b)^t$ and $Q(y^t)$
    $LB \leftarrow \max\{LB, Q(y^t) - \sum_{i \in H} f_i y_i^t\}$
    $\hat{P} \leftarrow \hat{P} \cup \{(\alpha, u, b)^t\}$
    $t \leftarrow t + 1$
end while
```

The computational efficiency of the Benders decomposition algorithm generally depends on the number of iterations required to obtain an optimal solution and the computational effort needed to solve MP as well as DSP at each iteration. In the following sections, we first explain how the subproblem can be solved efficiently and then describe how the size of the problem can be reduced via variable fixing.

3.2.2 A Two-Phase Method for Solving the Subproblem

Note that the dual subproblem contains $O(|N|^3|M|)$ variables and $O(|N|^4|M|)$ constraints. Solving such a huge problem is the most challenging part of the BD algorithm. Due to different definition of capacity usage of hubs in our problems, unlike Contreras et al. (2012), we can no longer cast the subproblem (3.12)-(3.15) as a transportation problem[2]. Moreover, the algorithm may suffer from slow convergence due to weakness of the generated cuts.

Pareto-optimal (PO) cuts (Magnanti and Wong 1981) have been extensively used in the literature for generating strong optimality cuts. Since DSP is a huge problem, solving it as whole is computationally prohibitive, specially for large instances of the problem. Note that for fixed values of the b-variables in the DSP, DSP can be solved as $|M||K|$ independent problems, one for each (k, m) (see §3.2.3 below). By fixing b at specific values, Contreras et al. (2012) generate good cuts by approximating the PO subproblem (Contreras et al. 2011a). Because of the structural differences of our problem, as we will demonstrate via computational experiments, the method proposed by Contreras et al. (2012) does not generate good enough cuts for our problem, or it comes with a high computational cost. Hence, we seek a method for generating good optimality cuts with less computational effort.

In this study, we propose a method for solving the subproblem in two sequential phases based on the set of values of the binary variables at which the subproblem is evaluated, and break each phase into smaller and simpler problems. In Phase I, we obtain the optimal value of the subproblem (i.e. $Q(y^t)$). In Phase II, we strengthen the cut while preserving the

[2] There exists a Lagrangian relaxation of the subproblem which converts it into a transportation problem. However, this transportation problem would be different than that of Contreras et al. (2012). Moreover, due to difficulties with convergence, solving this problem using a sub-gradient method would be computationally challenging.

optimality and feasibility of the solution. Note that this procedure does not require solving the subproblem twice; rather, we break the subproblem into two smaller problems and solve them sequentially.

At iteration t of the BD algorithm, we obtain an optimal solution y^t from MP. Let $H^t = \{i : y_i^t = 1\}$ and $\bar{H}^t = \{i : y_i^t = 0\}$ be the set of open and closed hubs, respectively. Note that any feasible value of b_i and u_{ik}^m would be optimal when $i \in \bar{H}^t$. Hence, we can solve the subproblem in two phases. In *Phase I*, we remove the variables b_i and u_{ik}^m and their corresponding constraints from DSP associated with $i \in \bar{H}^t$, and compute the values of the remaining variables by solving the following *Phase I* subproblem (DSP-I).

$$\min \quad \sum_{k \in K} \sum_{m \in M} \alpha_k^m + \sum_{i \in H^t} (\Gamma_i b_i + \sum_{k \in K} \sum_{m \in M} u_{ik}^m) \tag{3.20}$$

$$\text{s.t.} \quad \alpha_k^m + u_{ik}^m + u_{jk}^m + w_k^m (b_i + b_j) \geq (r_k^m - \hat{C}_{ijk}) w_k^m \qquad k \in K, m \in M, (i,j) \in A_k^t \tag{3.21}$$

$$\alpha_k^m + u_{ik}^m + w_k^m b_i \geq (r_k^m - \hat{C}_{iik}) w_k^m \qquad k \in K, m \in M, i \in H^t \tag{3.22}$$

$$\alpha_k^m, u_{ik}^m, b_i \geq 0 \qquad k \in K, m \in M, i \in H^t \tag{3.23}$$

where $A_k^t = \{(i,j) \in A_k \cap H^t \times H^t : i \neq j\}$ denotes the set of distinct potential open hub arcs associated with commodity k at iteration t. Note that α-variables are independent from i, accordingly, solving DSP-I results in obtaining the optimal values of all α-variables. Hence, in *Phase II*, we find feasible values of b_i and u_{ik}^m for $i \in \bar{H}^t, k \in K, m \in M$ with respect to constraints (3.13)-(3.15), aiming to generate optimality cuts as strong as possible.

Let $\bar{A}_k^t = \{(i,j) \in A_k \cap \bar{H}^t \times \bar{H}^t : i \neq j\}$ denote the set of distinct potential closed hub arcs of commodity $k \in K$ at iteration t, and let $H_i^t = \{j \in H^t : (i,j) \in A_k \textit{ or } (j,i) \in A_k\}$ be the set of open hubs that together with the closed hub $i \in \bar{H}^t$ form a potential hub arc for commodity $k \in K$. Updating DSP by fixing the value of the computed variables and removing the already satisfied constraints, we get the following *Phase II* feasibility problem:

$$u_{ik}^m + u_{jk}^m + w_k^m(b_i + b_j) \geq \rho_{ij}^{km} \qquad k \in K, m \in M, (i,j) \in \bar{A}_k^t \tag{3.24}$$

$$u_{ik}^m + w_k^m b_i \geq \rho_{ii}^{km} \qquad k \in K, m \in M, i \in \bar{H}^t \tag{3.25}$$

$$u_{ik}^m, b_i \geq 0 \qquad k \in K, m \in M, i \in \bar{H}^t \tag{3.26}$$

where $\rho_{ij}^{km} = (r_k^m - \hat{C}_{ijk})w_k^m - \alpha_k^m$ and

$$\rho_{ii}^{km} = \max\left\{\max_{j \in H_i^t}\{(r_k^m - \hat{C}_{ijk})w_k^m - u_{jk}^m - w_k^m b_j\}, (r_k^m - \hat{C}_{iik})w_k^m\right\} - \alpha_k^m. \tag{3.27}$$

Recall that the optimality cuts are in the form of

$$\eta \leq \sum_{k \in K}\sum_{m \in M} \alpha_k^m + \sum_{i \in H_t} y_i \kappa_i + \sum_{i \in \bar{H}_t} y_i \kappa_i, \tag{3.28}$$

where $\kappa_i = \Gamma_i b_i + \sum_{k \in K}\sum_{m \in M} u_{ik}^m$ is the coefficient of hub i in the produced cut. By the end of Phase I subproblem, the constant term (i.e. $\sum_{k \in K}\sum_{m \in M} \alpha_k^m$) and coefficient of currently open hubs (i.e. κ_i for $i \in H^t$) in (3.28) are computed. To produce a strong cut, we need to find a feasible solution to the Phase II problem such that the remaining coefficients (i.e. κ_i for currently closed hubs) in the produced optimality cut are as small as possible. This problem is a multi-objective optimization problem of the form

$$[\text{DSP-II}(y^t)] \qquad \min_{(3.24)-(3.26)} \kappa = (\kappa_i)_{i \in \bar{H}^t}. \tag{3.29}$$

We propose two algorithms for producing non-dominated solutions for Phase-II subproblem. In the first method, we scalarize this multi-objective problem into a single objective linear problem by assigning equal weights to each κ_i, which mimics the idea of producing dense cuts as suggested by Fischetti et al. (2010). We show that the special structure of the Phase II subproblem allows for solving this problem as a set of LP-relaxations of maximum weighted matching problems for a given value of b_i for $i \in \bar{H}^t$. In the second approach, we produce non-dominated solutions to DSP-II by solving this problem in a lexicographic manner. We show that this version of DSP-II can be solved as a sequence of LP-relaxations of knapsack problems. In the following sections, we present some theoretical insights and discuss how to solve the two phases efficiently.

3.2.3 Solving the Phase I Subproblem

We now turn our attention to solving the DSP-I subproblem (3.20)-(3.23). First, note that for b fixed, DSP-I (3.20)-(3.23) can be decomposed into smaller problems, one for each $k \in K$ and $m \in M$, denoted DSP-I(km):

$$\begin{aligned}
\min \quad & \alpha_k^m + \sum_{i \in H^t} u_{ik}^m && && (3.30)\\
\text{s.t.} \quad & \alpha_k^m + u_{ik}^m + u_{jk}^m \geq \beta_{ij}^{km} && (i,j) \in A_k^t && (3.31)\\
& \alpha_k^m + u_{ik}^m \geq \beta_{ii}^{km} && i \in H^t && (3.32)\\
& \alpha_k^m, u_{ik}^m \geq 0 && i \in H^t, && (3.33)
\end{aligned}$$

where $\beta_{ij}^{km} = (r_k^m - \hat{C}_{ijk} - b_i - b_j)w_k^m$ for $(i,j) \in A_k^t$, and $\beta_{ii}^{km} = (r_k^m - \hat{C}_{iik} - b_i)w_k^m$ for $i \in H^t$. We further simplify DSP-I by noting the following property of DSP-I.

Proposition 3.1. *There exists an optimal solution to DSP-I, in which $u_{ik}^m = 0$ for $k \in K$, $m \in M$ and $i \in H^t$.*

Proof. Let μ_{ij} and μ_{ii} be the dual variables associated with (3.31) and (3.32), respectively, for $(i,j) \in A_k^t$ and $i \in H^t$. Dual of DSP-I(km) can be formulated as:

$$\max \quad \sum_{a \in \hat{A}_k^t} \mu_a \beta_a^{km} \tag{3.34}$$

$$\text{s.t.} \quad \sum_{a \in \hat{A}_{ke}^1} \mu_a \le 1 \tag{3.35}$$

$$\sum_{a \in \hat{A}_k^t : i \in a} \mu_a \le 1 \qquad i \in H^t \tag{3.36}$$

$$\mu_a \ge 0 \qquad a \in \hat{A}_k^t \tag{3.37}$$

where $\hat{A}_k^t = A_k^t \cup \{(i,i) : i \in H^t\}$. Constraints (3.36) are implied by (3.35) for every $i \in H^t$, thus can be removed entirely from the dual problem. Accordingly, it is optimal to set the dual variables associated with (3.36) to zero, i.e. $u_{ik}^m = 0$ for $i \in H^t$. □

The outcome of Proposition 3.1 is that we can compute the optimal solution of DSP-I(km) and its dual using the corollaries stated below:

Corollary 3.1. *In the optimal solution of DSP-I(km),* $\alpha_k^m = \max\{0, \max_{a \in \hat{A}_k^t}\{\beta_a^{km}\}\}$.

Corollary 3.2. *Let* $a^* = \arg\max_{a \in \hat{A}_k^t}\{\beta_a^{km}\}$*, where ties are broken arbitrarily. In the optimal solution of Dual DSP-I(*km*),* $\mu_a = 0$ *for* $a \in \hat{A}_k^t \setminus \{a^*\}$*. If* $\beta_{a^*}^{km} > 0$*, then* $\mu_{a^*} = 1$*, otherwise* $\mu_{a^*} = 0$.

When the u-variables are set to 0 in the DSP-I subproblem, we refer to the resulting problem as the restricted DSP-I subproblem. Consequently, restricted DSP-I can be solved using a cutting-plane algorithm, where b is the master problem variable and α is the subproblem variable, and the dual subproblems are solved using Corollary 3.2. Equivalently, DSP-I can be solved using Dantzig-Wolfe decomposition in the dual space of DSP-I. We also note the following remark on difficulty of solving the restricted DSP-I as opposed to DSP.

Remark 3.2. *Restricted DSP-I subproblem contains* $O(|M||K||H^t|^2)$ *constraints and* $O(|M||K|+$

$|H^t|)$ *variables. Since only a few hubs are opened at each iteration (i.e.* $|H^t| \ll |H|$*), the restricted DSP-I is much easier to solve than DSP.*

3.2.4 Solving the Phase II Subproblem as Maximum Weighted Matching Problems

In our first approach for solving the DSP-II subproblem, we assign equal weights to each κ_i and turn the multi-objective problem (3.29) into the following single objective problem:

$$\min \left\{ \sum_{i \in \bar{H}^t} \kappa_i = \sum_{i \in \bar{H}^t} (\Gamma_i b_i + \sum_{k \in K} \sum_{m \in M} u_{ik}^m) : (3.24) - (3.26) \right\}. \tag{3.38}$$

Similar to DSP-I, for a given vector $(b_i)_{i \in \bar{H}^t}$, (3.38) can be restated for each $k \in K$ and $m \in M$ as:

$$\begin{aligned}
&[\text{DSP-II}(km)] \quad \min \quad \sum_{i \in H^t} u_{ik}^m && && (3.39)\\
&\text{s.t.} \quad u_{ik}^m + u_{jk}^m \geq \rho_{ij}^{km} - w_k^m (b_i + b_j) && (i,j) \in \bar{A}_k^t && (3.40)\\
&u_{ik}^m \geq \rho_{ii}^{km} - w_k^m b_i && i \in \bar{H}^t && (3.41)\\
&u_{ik}^m \geq 0 && i \in \bar{H}^t. && (3.42)
\end{aligned}$$

Assume that the values of the b-variables are known. We will discuss how suitable values of these variables can be computed later. Observe that (3.41) jointly with (3.42) serve as lower bounds on u_{ik}^m. Define $\tau_{ik}^m = u_{ik}^m - lb_{ik}^m$, where $lb_{ik}^m = \max\{\rho_{ii}^{km} - w_k^m b_i, 0\}$. DSP-II$(km)$ can be restated as:

$$[\text{MWC}(km)] \quad \min \quad \sum_{i \in \bar{H}^t} \tau_{ik}^m \tag{3.43}$$

$$\text{s.t.} \quad \tau_{ik}^m + \tau_{jk}^m \geq \beta_{ij}^{km} \qquad (i,j) \in \bar{A}_k^t \tag{3.44}$$

$$\tau_{ik}^m \geq 0 \qquad i \in \bar{H}^t. \tag{3.45}$$

where $\beta_{ij}^{km} = \rho_{ij}^{km} - w_k^m(b_i + b_j) - lb_{ik}^m - lb_{jk}^m$ for $(i,j) \in \bar{A}_k^t$. Not that by definition, $i \neq j$ for each $(i,j) \in \bar{A}_k^t$. Also, assume without loss of generality that β_{ij}^{km} values are non-negative, otherwise their corresponding constraints can be dropped. Problem (3.43)-(3.45) is commonly known as the *Minimum Weight Cover (MWC)* problem (see e.g. Galil 1986), which is the dual of the LP-relaxation of the *Maximum Weighted Matching (MWM)* problem in the edge-weighted undirected graph $G_{km} = (\bar{H}^t, \bar{A}_k^t)$, where edges are weighted according to β_{ij}^{km}. LP-relaxation of MWM can be formulated as

$$[\text{MWM}(km)] \quad \max \quad \sum_{a \in \bar{A}_k^t} \mu_a^{km} \beta_a^{km} \tag{3.46}$$

$$\text{s.t.} \quad \sum_{a \in \bar{A}_k^t : i \in a} \mu_a^{km} \leq 1 \qquad i \in \bar{H}^t \tag{3.47}$$

$$\mu_a^{km} \geq 0 \qquad a \in \bar{A}_k^t, \tag{3.48}$$

where μ_a^{km} is the dual variable associated with (3.44) and represents the extent to which edge $a \in \bar{A}_k^t$ is picked in the MWM. To solve this problem, we note that unlike the LP-relaxation of the MWM problem in general graphs, LP-relaxation of the MWM problem in bipartite graphs (denoted MWMB) is tight, in that there exists an integral solution that is optimal for LP-relaxation of MWMB. Hence, methods proposed for solving MWMB can also be used for solving its LP-relaxation. In Appendix B.1, we show that the MWM problem (3.46)-(3.48) can be transformed into an MWMB counterpart problem in an equivalent sparse bipartite graph.

Several algorithms have been proposed for solving MWMB. (For a review on recent methods, see, e.g., Burkard et al. 2009.) As a byproduct of solving the MWM problem (3.46)-(3.48), we eventually need to find the optimal value of (3.39)-(3.42). Hence, we adapt a primal-dual algorithm due to Galil (1986), which can be solved in $O(mn \log_{\lceil m/n+1 \rceil} n)$, where $n = |\bar{H}^t|$ and $m = |\bar{A}^t_k|$ are the number of nodes and edges, respectively. The details of the proposed algorithm are given in Appendix B.1.

The strength of the cut generated using this method depends highly on the value at which vector $(b_i)_{i \in \bar{H}^t}$ is fixed. One could simply set b_i to zero for $i \in \bar{H}^t$, or to the average of b_i in previous iterations of the BD algorithm. However, our primary experiments show that these choices of b will likely result in weak cuts. Moreover, note that larger values of b result in fewer positive β^{km}_a values, thus reducing the number of edges (i.e. $|\bar{A}^t_k|$) of the graph under which the MWM problem is being solved. Consequently, the computational time for solving the MWM problems is lower for larger values of b. In Appendix B.2, we explain how proper values of $(b_i)_{i \in \bar{H}^t}$ can be calculated efficiently through a relaxation of (3.38).

3.2.5 Solving the Phase II Subproblem as Knapsack Problems

At each iteration of the BD algorithm, we obtain a solution y defining a set of open/closed hubs. The open hubs are potentially the ones with desirable properties; e.g., with high capacity, low installation cost, and/or proximity to demand points. Hence, the most frequently opened hubs in the preceding iterations of BD are more likely to be opened again in the subsequent iterations. Therefore, if we define $O^t_i = \sum\limits_{h:h \leq t} y^h_i$ as the number of times that hub i has been opened prior to iteration t of BD, we can assume that a higher value of O^t_i implies a higher chance for hub i to be open at iteration $t+1$.

Thus, in our second approach, we solve the multi-objective problem DSP-II (3.29) in a lexicographic manner by prioritizing minimizing the coefficients of more frequently opened hubs over the hubs that are less likely to be open in the subsequent iterations. In other words, we sequentially minimize the coefficient of y_i for one closed hub $i \in \bar{H}^t$ at a time, in the order of importance of the hubs, rather than minimizing the summation of coefficients of all closed hubs simultaneously. Therefore, we first solve DSP-II (3.29) as if $i = \arg\max_{j \in \bar{H}^t}\{O_j^t\}$ is the only hub in $\bar{H}^t$:

$$\begin{aligned} [\text{DSP-II}(i)] \quad \min \quad & \Gamma_i b_i + \sum_{k \in K}\sum_{m \in M} u_{ik}^m && && (3.49) \\ \text{s.t.} \quad & u_{ik}^m + w_k^m b_i \geq \rho_{ii}^{km} && k \in K, m \in M && (3.50) \\ & u_{ik}^m, b_i \geq 0 && k \in K, m \in M. && (3.51) \end{aligned}$$

Note that since i is assumed to be the only hub in $\bar{H}^t$, constraints (3.24) do not appear in this model, but will be satisfied by the next closed hubs in the sequence by means of updating the respective ρ_{ii}^{km} values using (3.27).

Upon solving DSP-II(i), we obtain the values of all dual variables associated with hub i (i.e., u_{ik}^m and b_i). Consequently, we fix these values, add i to H^t and remove i from $\bar{H}^t$. We continue this procedure with the next closed hub $i = \arg\max_{j \in \bar{H}^t}\{O_j^t\}$ with ρ_{ii}^{km} values updated according to the new sets H^t and H_i^t using (3.27). We replicate this procedure until values of all dual variables are computed (i.e., until $\bar{H}^t$ becomes empty). An overview of this procedure is presented in Algorithm 5.

Algorithm 5 Solving DSP-II as a sequence of continuous knapsack problems

```
while $\bar{H}^t \neq \emptyset$ do
    $i \leftarrow \arg\max_{j \in \bar{H}^t}\{O_j^t\}$
    Compute $\rho_{ii}^{km}$ values using (3.27) for the selected $i$ and each $k \in K$ and $m \in M$ with respect to the updated $H^t$ and $H_i^t$.
    SOLVE DSP-II($i$) and obtain $b_i$ and $u_{ik}^m$ for $k \in K$ and $m \in M$.
    $\bar{H}^t \leftarrow \bar{H}^t \setminus \{i\}$
    $H^t \leftarrow H^t \cup \{i\}$
end while
```

Observe that DSP-II(i) is the dual of the LP-relaxation of a knapsack problem (KP) with knapsack capacity Γ_i, and items $(k,m) \in K \times M$ with weight w_k^m and profit ρ_{ii}^{km}, as formulated below:

$$
\begin{aligned}
[\text{KP}(i)] \quad & \max \quad \sum_{(k,m)\in K\times M} \rho_{ii}^{km}\mu_k^m & & (3.52)\\
& \text{s.t.} \quad \sum_{(k,m)\in K\times M} w_k^m \mu_k^m \leq \Gamma_i & & (3.53)\\
& 0 \leq \mu_k^m \leq 1 \qquad (k,m) \in K \times M, & & (3.54)
\end{aligned}
$$

where μ_k^m is the dual variable associated with constraint (3.50) and represents the extent to which item $(k,m) \in K \times M$ is picked. Note that the items with a non-positive profit (i.e., $\rho_{ii}^{km} \leq 0$) can be discarded from the problem. Dantzig (1957) showed that the optimal solution to this problem can be found by filling the knapsack in a non-increasing order of profit-to-weight ratio ρ_{ii}^{km}/w_k^m of the items, until we reach the first item that does not fit into the knapsack. This item is called the break item (denoted $(\bar{k},\bar{m})$), which is partially picked according to the residual capacity. Using the complementary-slackness conditions, it can be verified that the optimal value of b_i is the profit-to-weight ratio of the break item; i.e.,

$$b_i = \rho_{ii}^{\bar{k}\bar{m}}/w_{\bar{k}}^{\bar{m}}. \tag{3.55}$$

Moreover, using (3.50) and (3.51), the optimal value of u_{ik}^m can be calculated by setting $u_{ik}^m = \max\{\rho_{ii}^{km} - w_k^m b_i, 0\}$. By finding the break item as a weighted median, rather than by explicitly sorting the items, Balas and Zemel (1980) showed that the LP-relaxation of knapsack problem can be solved in $O(n)$ instead of $O(n \log n)$, where $n = |K||M|$ is the number of items. This implies that the optimal values of b and u can be found in the same time bound.

3.2.6 Variable Fixing

Variable fixing can improve the efficiency of the Benders decomposition algorithm by reducing the computational time of solving the master problems and subproblems by means of solution space reduction. The dominance properties presented in §3.1 reduce the size of the model significantly via preprocessing. We can further reduce the size by eliminating hubs that cannot be open in an optimal solution. Contreras et al. (2011a) propose two reduction tests that eliminate such hubs by using the information obtained during the inner iterations of the BD algorithm. In this study, we adapt and improve on these tests for eliminating hubs from H and the associated variables from the model.

The first reduction test is based on the primal information obtained by solving the LP-relaxation of MP. Let MP^t_{LP} denote the LP-relaxation of MP at iteration t, z^t_{LP} its optimal value, and $\bar{f}^t_i$ the reduced cost associated with variable y_i for $i \in H$. Let LB be a known lower bound on the optimal value of MP. Since $z^t_{LP} + \bar{f}^t_i$ provides a lower bound on the optimal value of MP, any hub $i \in H$ for which $z^t_{LP} + \bar{f}^t_i < LB$ cannot be open in any optimal solution. Hence, such hubs and their associated variables in the master and subproblem can be discarded in subsequent iterations.

The second reduction test is a logical test that attempts to prove that a set of hubs $\tilde{H} \subset H$ must be closed in an optimal solution. Let $\text{MP}^t(\tilde{H})$ denote the MP at iteration t with the additional constraint $\sum_{i \in \tilde{H}} y_i \geq 1$, and $z^t_{MP}(\tilde{H})$ its optimal value. Let LB be a lower bound on the optimal value of MP. Note that $\text{MP}^t(\tilde{H})$ provides a lower bound on the optimal value of MP at iteration t. Hence, if $z^t_{MP}(\tilde{H}) < LB$, then none of these hubs can be open in any optimal solution. Therefore, the hubs contained in $\tilde{H}$ and their associated variables can be safely eliminated in the subsequent iterations.

3.3 Computational Experiments

3.3.1 Experimental Setup

We use the well-known Australia Post (AP) dataset to test our models and algorithms. AP dataset contains postal service data of 200 cities in Australia and it is the most commonly used dataset in hub location literature (Ernst and Krishnamoorthy 1996). The distances (d_{ij}) and the postal flow between pairs of cities (w_k) are provided in OR Library, and a computer code is presented to generate smaller subsets of the data by grouping cities (Beasley 1990). We assume that the demand of commodities are segmented into 3 classes; i.e., $|M| = 3$, where $w_k^1 = 0.2w_k$, $w_k^2 = 0.3w_k$, and $w_k^3 = 0.5w_k$. Motivated from the postal delivery applications, where the price of sending a parcel depends on its size and the distance between the origin-destination, and also considering revenue elasticity of demand, the revenue per unit demand is taken to be dependent on the distance, class, and the amount of commodity to be shipped. Hence, for the revenue from commodity $k \in K$ of class $m \in M$, we generate random values as $r_k^m = \gamma^m \frac{c_k}{w_k^m}$, where $\gamma^1 \sim U[50, 60]$, $\gamma^2 \sim U[40, 50]$, and $\gamma^3 \sim U[30, 40]$. Collection, transfer, and distribution costs per unit are taken as $\chi = 2$, $\alpha = 0.75$, and $\delta = 3$ as defined in the AP dataset (Beasley 1990). We test instances with $|H| \in \{10, 20, 25, 40, 50, 75, 100, 200\}$.

There are two different sets for installation costs and capacities of hubs available on the AP dataset referred to as loose and tight. The opening cost of hubs in the instances with tight (T) installation costs is larger than those with loose (L) installation costs. In contrast, instances with tight (T) capacities have smaller available capacities compared to the instances with loose (L) capacities. Hence, there are four instances for a given node size corresponding to different combinations of installation costs and capacities. We denote each instance as $nf\Gamma$ where n is the instance size, f is the installation costs, and Γ is the capacity.

Computational experiments were carried out on a workstation that contains: Intel Core i7-

3930K 2.61GHz CPU, and 39 GB of RAM. The algorithms were coded in $C\#$ and the time limit was set to 15 hours. The master problems of all versions of the Benders decomposition algorithms as well as the master problems of the inner cutting-plane algorithm of Phase I subproblems were solved using the callable library of CPLEX 12.7.

3.3.2 Results

In this section, we first evaluate the performance and effectiveness of the algorithms proposed in §3.2 for the deterministic model. We implemented two different versions of Algorithm 4 referred to as BD1 and BD2, corresponding to different solution strategies of Phase II. In BD1, Phase II is solved as LP-relaxations of maximum weighted matching problem as described in §3.2.4, whereas in BD2, Phase II is solved as LP-relaxations of knapsack problems as described in §3.2.5. For comparison, we also implemented Pareto-optimal cuts (PO), the best known cuts from the literature, to solve our subproblems. We used $\tilde{y}$ with $\tilde{y}_i = 0.1$ as the core point as suggested by Contreras et al. (2012). We used the two reduction tests as described in §3.2.6 to eliminate candidate hubs within all of the three algorithms: PO, BD1, and BD2.

The detailed results of the comparison between these Benders algorithms using the AP instances are provided in Table 3.1. The first column represents the name and size of the instance. The next four columns labeled "Total time (sec)" present the computational time of instances (in seconds) obtained from solving the problems to optimality by using CPLEX, PO, BD1, and BD2, respectively. The next three columns labeled "Iterations" provide the required number of iterations for the convergence of the algorithms PO, BD1, and BD2, respectively. The columns labeled "% hubs elim." present the percentage of the total candidate hubs eliminated by algorithms PO, BD1, and BD2, respectively. The last two columns labeled "Optimal solution" indicate the maximum net profit and the locations of hub nodes, respectively, for the optimal solution found for each of the considered instances. Whenever an algorithm is not able to solve an instance within the time limit (15 hours of CPU time) to optimality, we write "Time" in the corresponding entry of the table. If an algorithm runs out of memory, we write "Mem".

Our goal of presenting the results of PO in Table 3.1 is to compare the strength of the cuts generated by BD1 and BD2 with this method. We use the decomposition scheme proposed by Contreras et al. (2012) for solving the Pareto-optimal subproblem. In this method, at iteration t of Algorithm 4, by fixing the value of $(b_i)_{i \in H^t}$ at the optimal values obtained from the original subproblem, and by setting $(b_i)_{i \in \bar{H}^t}$ to zero, they decompose the PO subproblem into $|K|$ (here $|K||M|$) independent problems. These problems can be solved using an LP solver, however, as Contreras et al. (2011a) argue, for computational tractability, they sacrifice the strength of the cuts by solving the resulting problems via an approximation technique. Since our goal of implementing PO method is to compare the strength of the resulting cuts with our proposed methods, after decomposing the PO subproblem into $|K||M|$ independent problems, rather than employing the approximation technique, we solve the dual of the resulting problems using the CPLEX LP solver. Note that the number of iterations cannot be reduced by the approximation algorithm, hence the number of iterations of PO as reported in Table 3.1 provides a lower bound on the number of iterations that we would obtain using the approximation technique proposed by Contreras et al. (2011a).

Table 3.1 shows that both algorithms BD1 and BD2 outperform CPLEX in terms of computational time and the number of instances solved to optimality. Additionally, the results of Table 1 clearly indicate that our algorithms (BD1 and BD2) outperform PO, with the only exception of instance 40LL.

Each of the Benders algorithms proposed in this chapter is able to solve all considered instances to optimality within an hour of CPU time, with the exception of instances 200LL

	Total time (sec)				Iterations			% hubs elim.			Optimal solution	
$\|H\|$	CPLEX	PO	BD1	BD2	PO	BD1	BD2	PO	BD1	BD2	Profit	Open hubs
10LL	1.02	1.37	0.13	0.01	7	4	4	50.0	60.0	60.0	20,417	5,6,9,10
10LT	0.89	0.94	0.03	0.01	5	4	4	60.0	80.0	80.0	3,336	5
10TL	1.00	0.81	0.03	0.07	9	6	6	50.0	60.0	60.0	13,488	5,9
10TT	1.16	0.26	0.03	0.03	4	3	3	80.0	90.0	90.0	2,682	5
20LL	9.64	35.64	1.70	2.11	51	11	11	70.0	65.0	80.0	100,443	7,9,10,19
20LT	11.38	45.26	0.97	1.34	60	7	7	65.0	65.0	65.0	57,139	5,10,12,14,19
20TL	10.34	21.52	0.84	0.99	37	9	9	75.0	80.0	85.0	49,559	5,7,10
20TT	6.17	4.51	0.22	0.29	11	3	3	70.0	95.0	95.0	10,135	10
25LL	15.42	68.03	7.91	6.31	48	18	18	76.0	72.0	72.0	125,390	7,14,17,23
25LT	12.82	157.51	3.27	2.35	93	7	7	68.0	72.0	72.0	88,022	6,9,10,12,14,25
25TL	14.21	53.80	2.12	2.09	40	9	9	76.0	72.0	72.0	76,933	6,9,14,23
25TT	12.30	100.76	2.02	1.92	71	10	10	80.0	76.0	76.0	35,121	6,10,14,25
40LL	79.23	261.31	52.57	42.47	48	64	58	70.0	82.5	85.0	76,995	12,22,26,29
40LT	83.18	623.02	39.58	28.93	66	42	38	82.5	82.5	82.5	66,860	12,14,26,29,30,38
40TL	41.25	71.96	9.53	5.26	13	14	11	85.0	82.5	82.5	62,960	14,19,29
40TT	39.21	209.68	6.80	5.49	24	11	11	87.5	82.5	90.0	49,938	14,19,25,38
50LL	142.18	466.78	64.34	39.92	49	42	38	76.0	88.0	86.0	73,054	15,28,33,35
50LT	138.98	666.23	33.69	26.67	24	19	17	76.0	86.0	82.0	68,904	6,26,32,46
50TL	115.73	188.11	11.01	8.06	14	11	13	88.0	88.0	88.0	54,009	3,26,45
50TT	105.01	303.43	7.42	5.23	20	7	7	88.0	88.0	92.0	45,506	17,26,48
75LL	Mem	8,979.00	525.81	387.66	111	105	88	85.3	89.3	86.7	142,551	14,23,35,37,56
75LT	Mem	8,327.27	137.00	87.58	73	23	15	90.7	92.0	90.7	110,194	14,25,32,35,38,59
75TL	Mem	3,918.76	69.85	71.12	26	13	13	92.0	92.0	96.0	89,562	14,35,37
75TT	Mem	4,986.99	42.92	33.84	36	12	11	93.3	90.7	90.7	81,697	25,32,38,59
100LL	Mem	32,872.67	904.00	665.94	58	53	54	94.0	94.0	94.0	1,777,224	29,55,64,73
100LT	Mem	Time	1,078.75	684.20	—	63	56	—	92.0	92.0	1,775,001	29,44,54,68,96
100TL	Mem	4,236.89	87.64	76.50	13	12	13	94.0	93.0	95.0	1,724,826	5,52,95
100TT	Mem	6,743.21	156.57	159.85	13	13	13	95.0	93.0	93.0	1,712,163	5,34,44,52,95
200LL	Mem	Time	11,435.00	6,430.31	—	40	39	—	96.0	96.0	1,102,073	43,95,159
200LT	Mem	Time	12,404.83	7,120.03	—	53	46	—	95.0	95.5	1,087,014	41,96,168,171
200TL	Mem	Time	953.94	879.52	—	12	12	—	97.5	97.5	1,068,431	54,95,186
200TT	Mem	Time	783.13	673.33	—	6	6	—	97.5	97.5	1,055,760	52,54,115,168,186

Table 3.1: Comparison of Benders reformulations and CPLEX with the AP dataset for the deterministic model.

and 200LT, which take approximately 3.3 hours for BD1 and 1.9 hours for BD2, respectively. The columns % hubs elim. show that a large percentage of candidate hubs can be eliminated by variable fixing. The columns Total time (sec) and Iterations indicate that the convergence of the Benders algorithm is improved by solving Phase II as LP-relaxations of knapsack problem, especially for the larger size instances. This implies that BD2 outperforms BD1; hence, we use BD2 for the rest of the computational experiments.

Table 3.1 also shows the locations of installed hubs in optimal solutions, where the optimal number of hubs to locate varies between one and six. It seems that, in these particular instances, the number of installed hubs does not depend on the size of the instance; it is rather more dependent on hub installation costs and capacities. For example, in the instances with tight installation costs and loose capacities, the problem tends to result in locating fewer hubs.

Instance	Demand class			Average
type	1	2	3	
LL	98.32%	94.58%	71.74%	88.21%
LT	98.32%	87.95%	55.83%	80.70%
TL	98.32%	93.45%	59.99%	83.92%
TT	93.00%	75.45%	39.64%	69.37%

Table 3.2: Percentage of total demand satisfied for each demand class.

In Table 3.2, we observe the percentages of satisfied demand from different market segments. The averages for each demand class are calculated over instances from Table 3.1 with the same type of installation costs and capacities. The last column provides the average percentages of total satisfied demand. Among the three demand classes, the first class is the one with the highest percentages of satisfied demand, as serving this class of demand yields the highest revenue. On the other hand, for the instances with the same configuration of hub installation costs and capacities, the third demand class, having the least revenues, has the least percentages of satisfied demand as expected. Moreover, instances with loose capacities (LL and TL) result in higher percentages on average compared to the instances with tight

Instance	Total time (sec)	Iterations	% hubs elim.	Profit	#open hubs
Set I					
50L	22.30	31	88.0	21,765	6
50T	15.49	19	90.0	16,943	4
100L	45.43	36	94.0	30,273	4
100T	50.18	23	95.0	30,215	4
150L	648.31	65	93.3	77,667	7
150T	523.05	25	94.7	74,952	6
200L	1,734.22	30	96.0	155,097	8
200T	3,160.84	33	95.5	144,874	7
250L	19,441.16	69	96.0	325,657	9
250T	4,861.76	37	97.2	181,027	5
300L	20,132.06	75	96.0	388,322	11
300T	14,885.61	52	97.7	188,944	6
Set II					
50L	64.29	35	88.0	85,327	6
50T	50.78	27	86.0	60,174	5
100L	224.41	48	92.0	201,451	7
100T	448.40	58	90.0	199,819	8
150L	3,785.39	121	91.3	458,915	12
150T	2,904.36	38	92.7	438,771	9
200L	1,442.94	25	95.0	236,247	8
200T	3,556.37	21	94.5	216,241	9
250L	10,414.33	62	95.2	437,819	11
250T	5,589.51	47	96.0	354,688	10
300L	17,367.18	68	96.0	1,532,224	10
300T	16,399.61	49	97.0	986,373	7
Set III					
50L	32.50	37	88.0	19,037	5
50T	19.93	32	90.0	16,353	5
100L	736.18	21	88.0	511,879	9
100T	530.51	16	91.0	39,672	7
150L	5,945.02	28	91.3	129,155	12
150T	5,327.03	27	92.7	964,137	9
200L	7,920.45	49	90.0	144,503	15
200T	6,738.52	45	93.5	103,987	11
250L	12,166.41	41	92.8	992,585	16
250T	11,057.31	32	95.2	710,562	11
300L	22,565.61	39	93.3	1,004,591	17
300T	18,349.47	34	95.0	903,773	14

Table 3.3: Computational results for the deterministic model using BD2 with Sets I, II, and III instances.

capacities (LT and TT).

To better understand the performance of the proposed algorithms from a computational point of view, we also present computational results with larger-size instances introduced by Contreras et al. (2011a), and later extended by Contreras et al. (2012). There are three different sets of instances, referred as Set I, Set II, and Set III, which are constructed by considering three different levels of magnitude for the amount of flow originating at a given node: low-level (LL) nodes, medium-level (ML) nodes, and high-level (HL) nodes. The total outgoing flow of LL, ML and HL nodes are obtained from the interval [1,10], [10,100], and [100,1000], respectively. Capacities of hubs are generated by using the formula provided in Ebery et al. (2000) in which parameter ρ is taken to be 0.5 and 1.5 for the loose (L) and tight (T) types of capacities, respectively. The other sets of parameters are as described at the beginning of §3.3. For Sets I, II, and III, we test instances with $|H| \in \{50, 100, 150, 200, 250, 300\}$. The detailed results are provided in Table 3.3 where the column titles have the same meanings as in the previous table.

All of the instances presented in Table 3.3 from Sets I, II, and III are solved to optimality. The most time-consuming instance in Sets I, II, and III took around 6, 5, and 6 hours, respectively, to solve to optimality. The averages of the computational times reported in Table 3.3 for the Sets I, II, and III are 1.5, 1.4, and 2.1 hours, respectively.

Lastly, we present additional runs from Sets I and II with $|H| \in \{350, 400, 500\}$ to analyze the limit of our algorithm. We have extended the CPU time limit to 24 hours for these instances. The results are presented in Table 3.4. Our algorithm is able to solve all of the instances to optimality within the time limit, except for the instance 500L in Set II. That particular instance resulted in an optimality gap of 2.52%. These results further confirm the efficiency and robustness of the BD2 algorithm when considering more challenging and larger-size instances.

Instance	Total time (sec)	Iterations	% hubs elim.	Profit	#open hubs
Set I					
350L	28,118.49	19	96.9	188,759	11
350T	21,881.74	15	97.7	147,501	7
400L	43,942.63	16	97.0	248,458	12
400T	35,763.21	14	97.5	186,239	9
500L	83,149.54	15	95.4	1,137,769	23
500T	54,731.43	11	97.0	821,439	14
Set II					
350L	31,025.56	31	95.7	682,514	15
350T	23,149.81	26	96.6	530,697	11
400L	40,209.23	29	95.3	852,627	19
400T	33,195.23	22	95.3	671,734	13
500L	(2.52% Gap)	17	71.8	965,113	25
500T	82,296.42	14	96.6	583,129	17

Table 3.4: Computational results of Sets I and II instances with $|H| \in \{350, 400, 500\}$.

3.4 Conclusions

In this chapter, we defined the profit maximizing hub location problem with capacity allocation by incorporating revenue management decisions, and embedding more realistic and challenging capacity constraints for hubs. We developed two Benders decomposition algorithms accelerated by a new cut strengthening technique for solving the Benders subproblems to solve the large-scale instances of the problem. We showed that the subproblems can be broken into smaller and simpler problem in two phases, We solved the first phase using a cutting-plane algorithm and formulated the second phase as a multi-objective optimization problem. We showed that non-dominated solutions to the second phase can be obtained by either a set of LP-relaxations of maximum weighted matching problems (BD1), or by a series of LP-relaxations of knapsack problems in a lexicographic manner (BD2). We further enhanced the algorithms by incorporating improved variable fixing techniques.

We performed extensive computational experiments on the well-known AP dataset, and also on larger-size instances from the literature, to analyze the performance of the proposed algorithms. In view of our computational results, both BD algorithms outperform the best known cuts (approximated Pareto-optimal cuts) in the literature, in terms of computational efficiency as well as strength of the cuts. The results further show that BD2 outperforms BD1 particularly on larger instances. BD2 succeeded to optimally solve instances with up to 500 nodes and 750,000 commodities of different demand classes.

Chapter 4

Benders Decomposition for Robust-Stochastic Hub Location Problems

In this chapter[1], we consider two sources of uncertainty in profit maximizing hub location problems: demand and revenue. Because of the availability of historical data, we assume that demand is described by a known probability distribution. On the other hand, since revenue might be affected by unpredictable external sources (e.g., economical conditions or competition) and historical data may fail to effectively describe such variations, it may not make sense to assume a known probability distribution for the revenue describing its behavior. Hence, we use robust optimization techniques to incorporate uncertain revenues into the problem by considering both interval representation and discrete scenarios. Modeling profit maximizing hub location problems using both robust and stochastic optimization techniques

[1] **Statement of collaboration:** The following is the summary of a joint work with my coauthors Gita Taherkhani and Sibel A. Alumur. This chapter contains materials from two papers (Taherkhani et al. 2020) and (Taherkhani et al. 2021). All authors were involved in writing the papers and my main contributions to these studies were in developing and implementing the models and algorithms.
Reprint permission acquired from Transportation Science with License IDs 1225865-1 and 1225865-2.

surely brings on extra computational challenges, yet we believe this is a much more realistic problem setting with respect to information availability.

We propose a robust-stochastic model by taking interval uncertainty into account for revenues using the max-min criterion with a budget of uncertainty. The max-min criterion maximizes profit under the worst case scenario. We then propose a min-max regret stochastic model by considering a finite set of scenarios that describe uncertainty associated with the revenues to model. With the min-max regret criterion, the decision maker decides based on the regret (or opportunity loss) from not selecting the best strategy. Both max-min profit and min-max regret criteria fit a conservative decision maker approach (Aissi et al. 2009). In this study, we model both approaches to empirically show the level of robustness and conservatism of each metric in addressing the uncertainty associated with revenues.

We develop exact algorithms based on Benders decomposition coupled with a sample average approximation (SAA) scheme to solve large-scale instances of the problem. We also propose novel acceleration techniques to enhance the convergence of the algorithms. We perform extensive computational analysis and investigate the effects of uncertainty under different settings on optimal hub networks and empirically evaluate the quality of the solutions obtained from different modeling approaches under various parameter settings.

The rest of the chapter is organized as follows. In §4.1, we present an SAA algorithm, coupled with Benders decomposition, for the problem with stochastic demand and deterministic revenue. In §4.2, we address simultaneous uncertainty in demand and revenue. We introduce two models, one based on interval uncertainty for revenue and the other based on discrete scenarios for revenue. We present two SAA algorithms, coupled with Benders decomposition, for the problem with stochastic demand and uncertain revenue. We perform extensive computational experiments in §4.3 to test our mathematical models and evaluate our algorithms. The chapter is concluded in §4.4 with concluding remarks. Finally, technical details and supplementary numerical results are provided in Appendix C.

4.1 Stochastic Demand with Known Revenue

We now model the problem with uncertain demand assuming that the uncertainty associated with demands is described by a known probability distribution. As noted before, demand is segmented into different classes and we consider a stochastic demand for each class, while revenues are assumed to be known. To model this problem, let $w_k^m(\xi)$ denote the random variables representing the future demand for commodity $k \in K$ of class $m \in M$ to be shipped from origin $o(k) \in N$ to destination $d(k) \in N$. The demands of different commodities are considered as independent random variables, whereas different demand classes of each commodity are assumed to be dependent and thus correlated. Let $\mathbb{E}_\xi$ denote the expectation with respect to ξ, and Ξ be the support of ξ. The *profit maximizing capacitated hub location problem with stochastic demand* can be modeled as:

$$
\begin{aligned}
\max \quad & \mathbb{E}_\xi \Big[\sum_{m \in M} \sum_{k \in K} \sum_{a \in A_k} (r_k^m - \hat{C}_{ak}) w_k^m(\xi) x_{ak}^m(\xi) \Big] - \sum_{i \in H} f_i y_i && && (4.1) \\
\text{s.t.} \quad & \sum_{a \in A_k} x_{ak}^m(\xi) \leq 1 && k \in K, m \in M, \xi \in \Xi && (4.2) \\
& \sum_{a \in A_k : i \in a} x_{ak}^m(\xi) \leq y_i && i \in H, k \in K, m \in M, \xi \in \Xi && (4.3) \\
& \sum_{m \in M} \sum_{k \in K} \sum_{a \in A_k : i \in a} w_k^m(\xi) x_{ak}^m(\xi) \leq \Gamma_i y_i && i \in H, \xi \in \Xi && (4.4) \\
& x_{ak}^m(\xi) \geq 0 && k \in K, m \in M, a \in A_k, \xi \in \Xi && (4.5) \\
& y \in Y. && && (4.6)
\end{aligned}
$$

The above model forms a two-stage stochastic program. The first stage problem corresponds to strategic hub location decisions. These long-term decisions will not be influenced by demand variations, accordingly, the y-variables become known in the first stage. However, the allocation decisions and the optimal routes of flows through the network, as well as the decision on how much of total capacity should be allocated to demand from different classes, do vary in response to the change of demand and, thus, are influenced by the stochastic demand. These tactical decisions are determined in the second stage depending on the particular realization of the random vector $\xi \in \Xi$. Accordingly, the variables x_{ak}^{m} become known in the second stage.

The objective function (4.1) contains a deterministic term which calculates the installation cost of the hubs, and the expectation of the second stage objective which calculates the expected value of revenue and transportation cost. To solve this problem, we integrate a sampling technique, named as *sample average approximation* (SAA) algorithm (the reader may refer to Shapiro and Homem-de Mello 1998, Kleywegt et al. 2002) with the BD algorithm detailed in the following sections.

4.1.1 Sample Average Approximation

SAA is a Monte Carlo simulation based approach to stochastic discrete optimization problems. The main idea of this method is to reduce the size of the problem by generating a random sample and approximating the expected value of the corresponding sample average function. The sample average optimization problem is then solved (using the BD algorithm in our case), and the procedure is repeated. The SAA scheme has previously been applied to stochastic supply chain design as well as hub location problems with a large number of scenarios (see, e.g., Santoso et al. 2005, Schütz et al. 2009, and Contreras et al. 2011b).

The main challenge in solving the stochastic problem (4.1)-(4.6) is the evaluation of the expected value of the objective function (Kleywegt et al. 2002). To deal with this problem, we use SAA scheme in which a random sample $\mathcal{N}$ of realizations of the random vector ξ is generated, and the second-stage expectation

$$\mathbb{E}_\xi\Big[\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k}(r_k^m - \hat{C}_{ak})w_k^m(\xi)x_{ak}^m(\xi)\Big]$$

is approximated by the sample average function

$$\frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k}(r_k^m - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn}, \tag{4.7}$$

where w_k^{mn} and x_{ak}^{mn} denote the amount of commodity $k \in K$ of class $m \in M$ to be shipped from origin $o(k) \in N$ to destination $d(k) \in N$ under sample $n \in \mathcal{N}$, and the fraction of commodity $k \in K$ of class $m \in M$ that is satisfied through a hub link $a \in A_k$ under scenario $n \in \mathcal{N}$, respectively. Accordingly, the approximated form of the stochastic problem by the SAA algorithm is modeled as:

$$\begin{aligned}
\max \quad & \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k}(r_k^m - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn} - \sum_{i\in H} f_i y_i && && (4.8)\\
\text{s.t.} \quad & \sum_{a\in A_k} x_{ak}^{mn} \leq 1 && k \in K, m \in M, n \in \mathcal{N} && (4.9)\\
& \sum_{a\in A_k : i\in a} x_{ak}^{mn} \leq y_i && i \in H, k \in K, m \in M, n \in \mathcal{N} && (4.10)\\
& \sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k : i\in a} w_k^m x_{ak}^{mn} \leq \Gamma_i y_i && i \in H, n \in \mathcal{N} && (4.11)\\
& x_{ak}^{mn} \geq 0 && k \in K, m \in M, a \in A_k, n \in \mathcal{N} && (4.12)\\
& y \in Y. && && (4.13)
\end{aligned}$$

Hereafter, we use the above approximated model (4.8)-(4.13) as the mathematical model for the profit maximizing capacitated hub location problem with stochastic demand. The optimal solution and the optimal value of the SAA problem (4.8)-(4.13) converges with probability one to their true counterpart (4.1)-(4.6) as the sample size increases (Kleywegt et al. 2002).

To choose $\mathcal{N}$ in practice, one should take into account the trade-off between the quality of the solution obtained from the SAA problem and the computational time required to solve it. Hence, it can be more efficient to solve the SAA problem (4.8)-(4.13) with independent samples repeatedly (say $|\mathcal{M}|$ replications) rather than increasing the sample size $\mathcal{N}$. The detailed SAA procedure is described in Appendix C.1.

4.1.2 Benders Decomposition for the SAA Problem

In this section, we present a BD algorithm coupled with SAA to solve the profit maximizing capacitated hub location problem with stochastic demand. By approximating the second-stage expectation via the sample average function as described in the previous section, the stochastic problem can be formulated as (4.8)-(4.13). To solve this problem, we apply the same procedure described in §3.2.1 and assume that the hub location decisions are handled in the master problem, while the rest is left to the subproblem. For a given sample $\mathcal{N}$ and fixed value of the y-variables at $y^t \in Y$, the *primal subproblem* (PSP($\mathcal{N}$)) reads as

$$[\text{PSP}(\mathcal{N})] \quad \max \quad \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k}(r_k^m - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn} \tag{4.14}$$

$$\text{s.t.} \quad \sum_{a\in A_k} x_{ak}^{mn} \leq 1 \qquad k\in K, m\in M, n\in\mathcal{N} \tag{4.15}$$

$$\sum_{a\in A_k: i\in a} x_{ak}^{mn} \leq y_i^t \qquad i\in H, k\in K, m\in M, n\in\mathcal{N} \tag{4.16}$$

$$\sum_{m\in M}\sum_{k\in K}\sum_{a\in A_k: i\in a} w_k^{mn}x_{ak}^{mn} \leq \Gamma_i y_i^t \qquad i\in H, n\in\mathcal{N} \tag{4.17}$$

$$x_{ak}^{mn} \geq 0 \qquad k\in K, m\in M, a\in A_k, n\in\mathcal{N}. \tag{4.18}$$

Observe that PSP($\mathcal{N}$) can be decomposed into $|\mathcal{N}|$ independent subproblems of the form PSP (3.7)-(3.11) for each $n \in \mathcal{N}$. Consequently, the dual subproblem associated with each n can be formulated as DSP (3.12)-(3.15) and solved using the techniques proposed in §3.2.2. In the following, we denote the DSP under scenario $n \in \mathcal{N}$ by DSP($\mathcal{N}, n$), in which w_k^m is replaced with w_k^{mn}, for each $k \in K$ and $m \in M$.

Let $P^n_{\mathcal{N}}$ for $n \in \mathcal{N}$ denote the polyhedron defined by feasible region of DSP$(\mathcal{N}, n)$, and let $Ex(P^n_{\mathcal{N}})$ be its extreme points. The master problem, denoted MP$(\mathcal{N})$, can then be stated as

$$\max \quad \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}} \eta^n - \sum_{i\in H} f_i y_i \tag{4.19}$$

$$\text{s.t.} \quad \eta^n \leq \sum_{k\in K}\sum_{m\in M} \alpha_k^{mn} + \sum_{i\in H} y_i(\Gamma_i b_i^n + \sum_{k\in K}\sum_{m\in M} u_{ik}^{mn}) \; n \in \mathcal{N}, (\alpha^n, u^n, b^n) \in Ex(P^n_{\mathcal{N}}) \tag{4.20}$$

$$y \in Y \tag{4.21}$$

As described in §3.2.1, we solve this problem by replacing $Ex(P^n_{\mathcal{N}})$ with $\hat{P}^n_{\mathcal{N}} \subset Ex(P^n_{\mathcal{N}})$, and solve a sequence of relaxed master problems and dual subproblems, until the optimal solution is found.

4.1.3 Acceleration Techniques for the SAA Algorithm

The variable fixing techniques presented in §3.2.6 can be applied to each sample of the stochastic model. We can further enhance the convergence of our SAA algorithm by exploiting the repetitive structure of the SAA algorithm. Note that we must solve $|\mathcal{M}|$ replications of problem (4.8)-(4.13). Assume that at some stage of the SAA algorithm, problem (4.8)-(4.13) has been solved for an arbitrary sample $\hat{\mathcal{N}}$. Consequently, upon solving MP$(\hat{\mathcal{N}})$, we obtain a set of dual solutions $(\hat{\alpha}^{\hat{n}}, \hat{u}^{\hat{n}}, \hat{b}^{\hat{n}})$ contained in $\hat{P}^{\hat{n}}_{\hat{\mathcal{N}}}$ for each $\hat{n} \in \hat{\mathcal{N}}$. Now, assume that we want to solve (4.8)-(4.13) for a different sample $\mathcal{N}$. Solving MP$(\mathcal{N})$ with initially setting $\hat{P}^n_{\mathcal{N}} = \emptyset$ would disregard the fact that the optimal solution of MP$(\hat{\mathcal{N}})$ is potentially a near-optimal solution to MP$(\mathcal{N})$. We can exploit this property and retrieve feasible solutions $(\alpha^n, u^n, b^n) \in \hat{P}^n_{\mathcal{N}}$ for scenario n of sample $\mathcal{N}$ from the solutions contained in $\hat{P}^{\hat{n}}_{\hat{\mathcal{N}}}$ for $\hat{n} \in \hat{\mathcal{N}}$.

Given a feasible solution for DSP$(\hat{\mathcal{N}}, \hat{n})$, the following proposition provides a feasible solution for DSP$(\mathcal{N}, n)$.

Proposition 4.1. *Let* $(\hat{\alpha}^{\hat{n}}, \hat{u}^{\hat{n}}, \hat{b}^{\hat{n}}) \in \hat{P}^{\hat{n}}_{\hat{\mathcal{N}}}$ *be a feasible solution for DSP*$(\hat{\mathcal{N}}, \hat{n})$*, and* $w^{m\hat{n}}_k$ *be the demand for commodity* $k \in K$ *of class* $m \in M$ *under scenario* $\hat{n} \in \hat{\mathcal{N}}$. (α^n, u^n, b^n) *defined by* (4.22)-(4.24) *is feasible for DSP*$(\mathcal{N}, n)$*:*

$$b^n_i = \hat{b}^{\hat{n}}_i \qquad i \in H \tag{4.22}$$

$$\alpha^{mn}_k = \frac{w^{mn}_k}{w^{m\hat{n}}_k} \hat{\alpha}^{m\hat{n}}_k \qquad k \in K, m \in M \tag{4.23}$$

$$u^{mn}_{ik} = \frac{w^{mn}_k}{w^{m\hat{n}}_k} \hat{u}^{m\hat{n}}_{ik} \qquad k \in K, m \in M, i \in H \tag{4.24}$$

Proof. From (4.23) and (4.24), we obtain $\hat{\alpha}^{m\hat{n}}_k = \frac{w^{m\hat{n}}_k}{w^{mn}_k} \alpha^{mn}_k$ and $\hat{u}^{m\hat{n}}_{ik} = \frac{w^{m\hat{n}}_k}{w^{mn}_k} u^{mn}_{ik}$, respectively. Feasibility of (α^n, u^n, b^n) for DSP$(\mathcal{N}, n)$ can easily be verified by replacing $\hat{b}^{\hat{n}}_i$, $\hat{\alpha}^{m\hat{n}}_k$, and $\hat{u}^{m\hat{n}}_{ik}$ respectively with b^n_i, $\frac{w^{m\hat{n}}_k}{w^{mn}_k} \alpha^{mn}_k$, and $\frac{w^{m\hat{n}}_k}{w^{mn}_k} u^{mn}_{ik}$, in constraints (3.13)-(3.15). □

Corollary 4.1. *The solution obtained by* (4.22)-(4.24) *provides a valid cut for MP*$(\mathcal{N})$*.*

Note that for a given scenario $n \in \mathcal{N}$, each scenario $\hat{n} \in \hat{\mathcal{N}}$ can provide a valid cut for MP$(\mathcal{N})$. To avoid overburdening the MP with too many cuts, for each scenario $n \in N$ we select a single scenario $\hat{n}^* \in \hat{\mathcal{N}}$ and convert the solutions contained in $\hat{P}^{\hat{n}^*}_{\hat{\mathcal{N}}}$ to feasible solutions for $\hat{P}^n_{\mathcal{N}}$ using Proposition 4.1. A heuristic way for choosing such a scenario is to choose the one with the least demand deviation from the demand under scenario $n \in \mathcal{N}$, i.e.

$$\hat{n}^*(n) = \arg\min_{\hat{n} \in \hat{\mathcal{N}}} \left\{ \sum_{(k,m) \in K \times M} |w^{m\hat{n}}_k - w^{mn}_k| \right\}. \tag{4.25}$$

It should, however, be noted that if in the process of solving MP$(\hat{\mathcal{N}})$, we eliminate a number of hubs via variable fixing, we will not calculate the dual variables associated with the eliminated hubs, hence the incomplete solution obtained by (4.22)-(4.24) may not provide a valid cut for MP$(\mathcal{N})$. To tackle this problem, we sacrifice the first sample of the SAA algorithm without employing the variable fixing rules to ensure that the resulting cuts can be used for the subsequent samples of SAA. Once these solutions are obtained, we add the respective cuts to the master problem of the subsequent samples and continue with performing variable fixing as introduced in §3.2.6.

4.2 Stochastic Demand with Uncertain Revenue

We now model the robust-stochastic problems when there is uncertainty associated with revenues under stochastic demand. We present two mathematical formulations for the robust-stochastic version of the problem, where we first model a max-min profit criterion and then a min-max regret criterion.

4.2.1 Case I: Max-min Profit Criterion

We use interval uncertainty for revenues in which each parameter r_k^m for $k \in K, m \in M$ takes values in $[\bar{r}_k^m - \hat{r}_k^m, \bar{r}_k^m]$, where $\bar{r}_k^m$ is the nominal value of revenue and $\hat{r}_k^m \geq 0$ represents the deviation from the nominal value. Let $\gamma_r \in [0, |K| \times |M|]$ be an integer value denoting the uncertainty budget as defined by Bertsimas and Sim (2003), which controls the level of conservatism in the objective and the maximum number of revenue parameters r_k^m whose value is allowed to differ from its nominal value $\bar{r}_k^m$. Since our problem is a profit maximization problem, worse scenarios occur when the realized profits are lower than the nominal profit. Hence, for every given solution the adversary will choose a maximum number γ_r of the revenue coefficients in such a way as to obtained the worst profit (Bertsimas and Sim 2003). Therefore, we are interested in finding a solution that optimizes against all such realizations. The *robust-stochastic model with max-min profit criterion* for the profit maximizing hub location problem with capacity allocation is then modeled as:

$$[\text{RS-I}] \quad \max_{(x,y)\in\mho} \mathbb{E}_\xi \Big[\sum_{k\in K}\sum_{m\in M}\sum_{a\in A_k} (\bar{r}_k^m - \hat{C}_{ak}) w_k^m(\xi) x_{ak}^m(\xi) - \nu_\xi(x) \Big] - \sum_{i\in H} f_i y_i \tag{4.26}$$

where $\mho = \{(x,y) : (4.2) - (4.6) \text{ are satisfied}\}$, and $\nu_\xi(x)$ is defined as follows:

$$\nu_\xi(x) = \max_{U_r \subseteq K\times M : |U_r| \leq \gamma_r} \sum_{(k,m)\in U_r} \sum_{a\in A_k} \hat{r}_k^m w_k^m(\xi) x_{ak}^m(\xi). \tag{4.27}$$

In the above equation, U_r represents the subset of commodities whose revenue values are subject to variation. The goal of $\nu_\xi(x)$ is to determine the worst case deviation from the total revenue over all possible revenue realizations for a given solution x. Note that in extreme cases when $\gamma_r = 0$ or $\gamma_r = |K| \times |M|$ (alternatively, when $U_r = \emptyset$ or $U_r = K \times M$, respectively), the problem can be reduced to the stochastic model and it has trivial solutions such that for all commodities (k,m), in the former case, $r_k^m = \bar{r}_k^m$, whereas in the latter case, $r_k^m = \bar{r}_k^m - \hat{r}_k^m$. These extreme cases represent the least and highest levels of conservatism, respectively. In general, a higher value of γ_r leads to a more conservative solution in the expense of a possibly lower profit.

We can reformulate $\nu_\xi(x)$ by introducing a binary variable z_k^m which determines whether or not class $m \in M$ of commodity $k \in K$ is subject to uncertainty; i.e., $z_k^m = 1$ if $(k,m) \in U_r$, and 0 otherwise.

$$\nu_\xi(x) = \max \quad \sum_{k\in K}\sum_{m\in M} \left(\hat{r}_k^m w_k^m(\xi) \sum_{a\in A_k} x_{ak}^m(\xi) \right) z_k^m \tag{4.28}$$

$$\text{s.t.} \quad \sum_{k\in K}\sum_{m\in M} z_k^m \leq \gamma_r \tag{4.29}$$

$$z_k^m \in \{0,1\} \qquad k \in K, m \in M. \tag{4.30}$$

Since γ_r is integer, $\nu_\xi(x)$ is simply a sorting problem. Hence, constraint (4.30) can be replaced with its linear relaxation counterpart without losing integrality. Let $\mu(\xi)$ and $\lambda_k^m(\xi)$ be the dual variables associated with constraints (4.29) and the linear relaxation of (4.30), respectively. The dual of problem (4.28)-(4.30) can be obtained as:

$$\begin{aligned}
\nu_\xi(x) = \min \quad & \gamma_r \mu(\xi) + \sum_{k \in K} \sum_{m \in M} \lambda_k^m(\xi) && && (4.31) \\
\text{s.t.} \quad & \mu(\xi) + \lambda_k^m(\xi) \geq \hat{r}_k^m w_k^m(\xi) \sum_{a \in A_k} x_{ak}^m(\xi) && k \in K, m \in M && (4.32) \\
& \lambda_k^m(\xi), \mu(\xi) \geq 0 && k \in K, m \in M. && (4.33)
\end{aligned}$$

With this formulation of $\nu_\xi(x)$, mathematical program (4.26) can be reformulated as the following MILP:

$$\begin{aligned}
\max_{(x,y) \in \mho} \; & \mathbb{E}_\xi \Big[\sum_{k \in K} \sum_{m \in M} \sum_{a \in A_k} (\bar{r}_k^m - \hat{C}_{ak}) w_k^m(\xi) x_{ak}^m(\xi) - \gamma_r \mu(\xi) - \sum_{k \in K} \sum_{m \in M} \lambda_k^m(\xi) \Big] - \sum_{i \in H} f_i y_i \\
& \text{s.t. } (4.32), (4.33) \qquad (4.34)
\end{aligned}$$

4.2.2 Case II: Min-max Regret Criterion

If there exists a set of scenarios describing uncertainty associated with the revenues, one may also use a min-max regret type objective function to model the problem (Correia and Saldanha-da Gama 2015). Let S_r define the set of scenarios for uncertain revenues and r_k^{ms} denote the amount of revenue obtained from satisfying a unit commodity $k \in K$ of class $m \in M$ under scenario $s \in S_r$.

For a given demand realization $\xi \in \Xi$, the maximum profit that can be achieved under revenue scenario $s \in S_r$, denoted by $Z^s(\xi)$, can be calculated by

$$
\begin{aligned}
Z^s(\xi) = \max \quad & \sum_{k \in K} \sum_{m \in M} \sum_{a \in A_k} (r_k^{ms} - \hat{C}_{ak}) w_k^m(\xi) x_{ak}^m(\xi) - \sum_{i \in H} f_i y_i && && (4.35)\\
\text{s.t.} \quad & \sum_{a \in A_k} x_{ak}^m(\xi) \le 1 && k \in K, m \in M && (4.36)\\
& \sum_{a \in A_k : i \in a} x_{ak}^m(\xi) \le y_i && i \in H, k \in K, m \in M && (4.37)\\
& \sum_{k \in K} \sum_{m \in M} \sum_{a \in A_k : i \in a} w_k^m(\xi) x_{ak}^m(\xi) \le \Gamma_i y_i && i \in H && (4.38)\\
& x_{ak}^m(\xi) \ge 0 && k \in K, m \in M, a \in A_k && (4.39)\\
& y \in Y. && && (4.40)
\end{aligned}
$$

For a given demand realization $\xi \in \Xi$, the regret of a solution $(x(\xi), y)$ under revenue scenario $s \in S_r$ is defined as the difference between the optimal profit that can be achieved under that scenario (i.e. $Z^s(\xi)$) and the total profit associated with $(x(\xi), y)$. With this definition, the *min-max regret stochastic model*, denoted RS-II, can be formulated as follows:

$$
\min_{(x,y) \in \mho} \mathbb{E}_\xi \Big[\max_{s \in S_r} \Big\{ Z^s(\xi) - (\sum_{k \in K} \sum_{m \in M} \sum_{a \in A_k} (r_k^{ms} - \hat{C}_{ak}) w_k^m(\xi) x_{ak}^m(\xi) - \sum_{i \in H} f_i y_i) \Big\} \Big]. \tag{4.41}
$$

The inner maximization calculates the maximum regret among all revenue scenarios. Replacing the inner maximization with a continuous variable $V(\xi)$, the above formulation can be linearized as follows:

$$
\min_{(x,y) \in \mho} \mathbb{E}_\xi [V(\xi)] \tag{4.42}
$$

$$
\text{s.t. } V(\xi) \ge Z^s(\xi) - (\sum_{k \in K} \sum_{m \in M} \sum_{a \in A_k} (r_k^{ms} - \hat{C}_{ak}) w_k^m(\xi) x_{ak}^m(\xi) - \sum_{i \in H} f_i y_i) \ \xi \in \Xi, s \in S_r. \tag{4.43}
$$

We now define $\bar{V}(\xi) := -(V(\xi) - \sum_{i \in H} f_i y_i)$ and replace $V(\xi)$ with $-(\bar{V}(\xi) - \sum_{i \in H} f_i y_i)$ and reformulate (4.42)-(4.43) in a maximization form as:

$$\max_{(x,y)\in\mho} \quad \mathbb{E}_{\xi}[\bar{V}(\xi)] - \sum_{i\in H} f_i y_i \tag{4.44}$$

$$\text{s.t.} \quad \bar{V}(\xi) - \sum_{k\in K}\sum_{m\in M}\sum_{a\in A_k}(r_k^{ms} - \hat{C}_{ak})w_k^m(\xi)x_{ak}^m(\xi) \leq -Z^s(\xi) \;\; \xi \in \Xi, s \in S_r. \tag{4.45}$$

We now like to compare the min-max regret stochastic model (RS-II) with the robust-stochastic model with max-min profit criterion (RS-I). Let's first assume that the set of revenue scenarios considered in the min-max regret model (i.e. S_r) complies with the requirements of the uncertainty sets considered in the robust-stochastic model with max-min profit criterion (i.e. U_r). In other words, let S_r consists of all revenue scenarios involving at most γ_r commodities with an uncertain revenue. The robust-stochastic model with max-min profit criterion for each solution (x, y) selects from S_r the scenario that minimizes the total revenue, and maximizes the expectation of this minimal revenue over all possible solutions (x, y). The min-max regret stochastic model, on the other hand, selects from S_r the scenario that maximizes the regret, and minimizes the expectation of this maximal regret over all possible solutions (x, y). Interestingly, as shown in Theorem 4.1 below, the robust-stochastic version with max-min profit criterion is actually a special case of the min-max regret stochastic model in which $Z^s(\xi) = \hat{Z}$ for some arbitrary value $\hat{Z}$ (e.g. 0) for each revenue scenario $s \in S_r$ and demand realization $\xi \in \Xi$.

Theorem 4.1. *Let S_r be the set of revenue scenarios where at most γ_r commodities are subject to revenue uncertainty. Then, the robust-stochastic model with min-max regret criterion* (4.41)*, in which regrets are calculated with respect to a fixed reference point $\hat{Z}$, is equivalent to the robust-stochastic model with max-min profit criterion* (4.26)*.*

Proof. Please see Appendix C.2 for the proof. □

As a consequence of Theorem 4.1, the robust-stochastic model with max-min profit criterion is computationally less challenging as there is no need to compute $Z^s(\xi)$ for each scenario. We empirically analyze the outcome and the level of robustness with both of the models through our computational experiments.

To approximate the expected values in the robust-stochastic models (4.41) and (4.26) we employ the same SAA scheme described in §4.1.1. We solve the SAA counterpart of the problems (4.34) and (4.44)-(4.45) using a Benders decomposition (BD) algorithm. The algorithms are very similar to the ones proposed for the model with stochastic demand and known revenue. In Appendices C.5 and C.6, we present exact algorithms based on BD coupled with SAA to solve the robust-stochastic models with max-min profit and min-max regret criteria, respectively.

4.3 Computational Experiments

4.3.1 Computational Results for the Stochastic Model

In this section, we first focus on the practical convergence of the SAA scheme using the stochastic model, we then test the performance of our methods on the instances involving up to 75 nodes. All computational experiments with the stochastic model are performed using BD2 coupled with SAA.

4.3.1.1 Sample generation

We generate independent samples for demands of commodities using a normal distribution parameterized as follows: Let $\bar{w}_k^m$ be the demand of commodity k of class m in the deterministic case. Moreover, let $\bar{w}_k = \sum_{m \in M} \bar{w}_k^m$ be the total demand of commodity k, and $\rho_m^k = \frac{\bar{w}_k^m}{\bar{w}_k}$ be the proportion of demand of segment m of commodity k. We assume that the total demand of k (i.e. w_k) is drawn from a normal distribution in which the mean demand is set to $\bar{w}_k$ and the standard deviation is equal to $\sigma_k = \nu \bar{w}_k$, where ν is the coefficient of variation. Consequently, once the total demand of commodity k is realized, the correlated demand of class m is computed as $w_k^m = \rho_m^k w_k$.

4.3.1.2 Results

In Appendix C.3 we analyze the practical convergence of the SAA procedure to choose number of replications $|\mathcal{M}|$ and the sample size per replication $|\mathcal{N}|$. As a result, we choose $|\mathcal{N}| = 50$ and $|\mathcal{M}| = 60$ to balance the trade-off between computation time and solution quality. In the following, we first evaluate the performance of the acceleration techniques proposed for SAA. The results are provided in Table C.1 in Appendix C.4. In our computational experiments, the algorithm runs up to five times faster, and more than two times faster on the average, with the implementation of the acceleration techniques. Hence, all computational experiments with the stochastic model are carried out using the acceleration techniques.

We now analyze and evaluate the performance of the SAA algorithm on larger-size instances with up to 75 nodes from the AP dataset. For each instance, we consider two values 0.5 and 1 as the coefficient of variation to represent the amount of uncertainty in the stochastic demand. The computational results are summarized in Table 4.1. The first two columns provide the number of nodes and the coefficient of variation. The next three columns labeled "Optimal solution" present the net profit ($\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}}$), the best hub locations, and the run time of instances (in seconds) obtained from solving the SAA algorithm, respectively. The next column labeled "% Gap" provides the percent optimality gap relative to the best solution obtained by the SAA algorithm. The last two columns labeled "CI for SAA % gap at" give the 95% and 99% confidence interval for the optimality gap of the best solution obtained by the SAA algorithm, respectively.

Instance			Optimal solution		% Gap	CI for SAA % gap at	
$\|H\|$	ν	Profit	Open hubs	Time (sec)	SAA	95%	99%
10LL	0.5	21,380	5,6,9,10	50.82	0.08	(-6.16, 11.73)	(-8.74, 14.31)
	1	22,529	5,6,9,10	41.76	-0.04	(-4.91, 13.47)	(-7.47, 16.03)
10LT	0.5	3,344	5	15.39	0.06	(-9.96, 13.67)	(-13.45, 17.16)
	1	3,358	5,6	18.33	0.18	(-7.24, 15.50)	(-9.29, 17.55)
10TL	0.5	13,890	4,5,9	33.89	0.14	(-14.02, 7.87)	(-18.39, 12.24)
	1	14,586	4,5,9	53.48	-0.04	(-11.05, 8.37)	(-12.19, 9.51)
10TT	0.5	2,690	5	14.97	0.03	(-9.96, 13.67)	(-13.56, 17.27)
	1	2,683	5	13.11	0.14	(-6.42, 16.57)	(-10.60, 20.75)
20LL	0.5	105,282	7,9,10,19	1,553.42	0.04	(-15.15, 14.31)	(-16.73, 15.89)
	1	113,599	7,9,10,14,19	2,713.17	0.03	(-9.92, 15.72)	(-14.67, 20.47)
20LT	0.5	59,727	5,9,10,12,14,19	1,442.36	0.05	(-4.45, 8.72)	(-9.21, 13.48)
	1	63,731	5,9,10,12,14,19	1,907.56	0.04	(-3.72, 10.74)	(-7.05, 14.07)
20TL	0.5	53,035	5,7,10	1,089.42	0.08	(-11.67, 10.97)	(-16.07, 15.37)
	1	57,511	5,7,10	1,173.79	0.06	(-8.60, 15.09)	(-11.86, 18.35)
20TT	0.5	13,672	10	101.11	0.02	(-10.62, 10.07)	(-11.90, 11.35)
	1	13,984	10	76.78	-0.03	(-8.08, 12.76)	(-10.94, 15.62)
25LL	0.5	133,240	7,14,17,23	1,418.24	-0.03	(-10.32, 13.46)	(-13.03, 16.17)
	1	140,164	7,14,17,23	2,014.32	-0.02	(-6.47, 14.83)	(-11.33, 19.69)
25LT	0.5	92,042	6,10,12,14,25	3,218.51	-0.01	(-6.23, 10.47)	(-11.13, 15.37)
	1	98,570	9,10,12,14,19,25	3,832.22	0.01	(-9.87, 11.51)	(-14.58, 16.22)
25TL	0.5	81,755	6,9,14,23	1,127.83	-0.02	(-8.48, 10.26)	(-9.95, 11.73)
	1	87,625	6,9,14,23	1,613.32	-0.02	(-5.98, 12.38)	(-10.64, 17.04)
25TT	0.5	36,956	6,10,14,25	1,420.91	0.00	(-9.16, 8.70)	(-11.60, 11.14)
	1	39,992	6,9,10,14,25	1,711.33	-0.01	(-7.18, 10.91)	(-8.92, 12.65)
40LL	0.5	80,696	12,22,26,29	15,219.35	-0.03	(-18.92, 17.71)	(-22.80, 21.59)
	1	86,456	9,22,26,29,38	17,612.49	0.02	(-15.39, 19.48)	(-16.88, 20.97)
40LT	0.5	71,192	12,14,26,29,30,38	20,369.86	-0.04	(-19.32, 22.57)	(-21.37, 24.62)
	1	76,989	5,14,19,26,29,30,38	22,928.64	0.04	(-16.86, 23.36)	(-18.55, 25.05)
40TL	0.5	65,621	14,19,29	7,549.28	-0.03	(-14.74, 19.37)	(-18.99, 23.62)
	1	71,406	14,19,29	8,084.24	0.05	(-12.37, 21.22)	(-14.30, 23.15)
40TT	0.5	52,843	14,19,25,38	9,672.36	-0.04	(-17.53, 19.75)	(-19.06, 21.28)
	1	57,349	5,19,25,30	10,836.71	0.05	(-15.95, 23.42)	(-20.59, 28.06)
50LL	0.5	77,216	15,28,33,35	13,565.38	0.03	(-19.46, 25.37)	(-27.00, 32.91)
	1	83,180	5,15,28,33,35	15,738.46	-0.03	(-17.11, 26.23)	(-25.07, 34.19)
50LT	0.5	73,467	6,26,32,46	11,874.35	0.05	(-15.83, 21.64)	(-22.38, 28.19)
	1	79,923	6,19,26,30,46	14,969.23	-0.06	(-13.26, 23.83)	(-16.90, 27.47)
50TL	0.5	58,384	3,26,45	5,595.46	0.05	(-16.30, 16.99)	(-20.26, 20.95)
	1	60,756	3,14,29,45	7,304.05	-0.09	(-12.61, 18.28)	(-17.92, 23.59)
50TT	0.5	48,376	6,26,48	6,393.29	0.12	(-14.67, 16.36)	(-20.39, 22.08)
	1	53,261	6,26,48	7,141.92	-0.07	(-12.71, 18.49)	(-17.16, 22.94)
75LL	0.5	145,792	14,23,35,37,56	43,955.24	0.10	(-17.54, 12.63)	(-23.37, 18.46)
	1	198,962	5,14,19,26,29,30,38	46,085.51	0.04	(-12.46, 18.49)	(-16.55, 22.58)
75LT	0.5	113,510	14,25,32,35,38,59	36,666.45	-0.08	(-14.29, 18.81)	(-18.50, 23.02)
	1	122,007	14,26,32,35,46,59	39,157.63	0.04	(-11.39, 19.41)	(-16.60, 24.62)
75TL	0.5	92,609	14,35,37	24,531.79	0.16	(-21.74, 19.83)	(-29.04, 27.13)
	1	96,829	14,35,37	25,742.90	0.07	(-18.25, 22.49)	(-24.14, 28.38)
75TT	0.5	81,697	25,32,38,59	32,559.79	0.14	(-23.67, 16.74)	(-31.43, 24.50)
	1	84,157	25,26,32,38,59	34,637.15	-0.10	(-17.34, 18.51)	(-22.33, 23.50)

Table 4.1: Computational results for the stochastic model with 48 instances of the AP dataset.

The results provided in Table 4.1 indicate that the estimated optimality gaps obtained by the SAA algorithm are always below 0.2%, and that the corresponding confidence intervals for the optimality gaps are quite narrow for both 95% and 99%. These confirm the efficiency of the SAA algorithm proposed for the problem with stochastic demand, and also imply that the solutions produced by our algorithm are good enough to be used in practical applications.

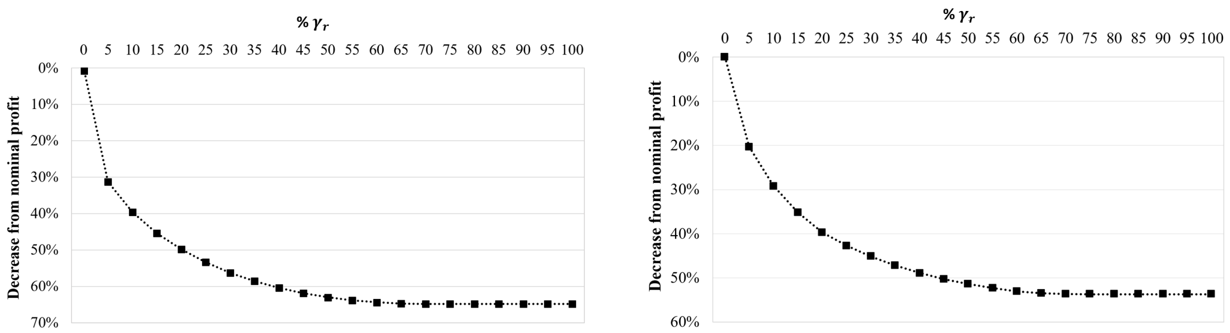
We next observe the effects of variability in uncertain demands on the solutions reported in Table 4.1. When the ν value increases, that is, when the variability in the uncertain demand increases, the net profit values and the computation time required for the SAA algorithm also increase. Note that the best found hub locations do not change significantly under these variations. We can identify a few instances in which hub locations change by demand variation, and in most of the instances, the locations of the hubs obtained with the deterministic and stochastic models are identical (Tables 3.1 and 4.1). It seems that, in these particular instances, the long-term location decisions are dependent more on the configuration of hub installation costs and capacities than the demand.

4.3.2 Computational Results for the Robust-Stochastic Model with Max-min Profit Criterion

For the analysis of the robust-stochastic model with max-min profit criterion, we take $\hat{r}_k^m \sim U[0, \varphi \bar{r}_k^m]$ to generate intervals of uncertainty, where φ is the maximum possible deviation from the nominal value of revenue. We first evaluate the effect of the uncertainty budget (γ_r) on total profit. We select two instances of the AP dataset on 20 and 25 nodes with $\varphi = 0.5$, and test the model using $\gamma_r \in \{0, 5\%, 10\%, \ldots, 100\%\}$. For simplicity, we use percentage to represent the budget of uncertainty which corresponds to the percentage of the revenue parameters under uncertainty.

Figures 4.1(a) and 4.1(b) plot the percentage of decrease from the nominal profit for different values of γ_r for the AP20TL and AP25LT instances, respectively. Let Z_{γ_r} denote the optimal profit obtained from the robust-stochastic model with max-min profit criterion when budget of uncertainty is γ_r and Z_0 denote the objective function value with the nominal profit that can be obtained from the stochastic model. The percentage of decrease from the nominal profit can then be calculated as $\frac{Z_0 - Z_{\gamma_r}}{Z_0}$ for any γ_r.

It is clear that a higher value of γ_r leads to a more conservative solution with a lower Z_{γ_r}. Moreover, as shown in Bertsimas and Sim (2003), Z_{γ_r} is a convex (in our case concave) function of γ_r. Consequently, as noted from the Figures 4.1(a) and 4.1(b), for smaller values of γ_r, percentage of decrease from the nominal profit drops faster compared to higher values of the budget of uncertainty. In particular, when we select our budget of uncertainty with $\gamma_r \geq 55\%$, we observe from both of the figures that, there is not much deviation in the optimal profits. This observation indicates that small and moderate values of γ_r provide better insights for evaluating the effects of the uncertainty budget on the solutions. For this reason, we use $\gamma_r \in \{15\%, 25\%, 50\%\}$ during the rest of our analysis.



(a) AP20TL

(b) AP25LT

Figure 4.1: Effect of the uncertainty budget on the max-min profit.

We now analyze the results obtained from the max-min profit model for larger size instances with up to 75 nodes from the AP dataset. For each instance, we consider two values to represent the amount of deviation from the nominal value of revenues; $\varphi \in \{0.5, 1\}$. The computational results are summarized in Table 4.2. The first two columns provide the size

Instance		$\gamma_r = 15\%$				$\gamma_r = 25\%$				$\gamma_r = 50\%$			
$\|N\|$	φ	Profit	Avg. iter.	Time (sec)	Open hubs	Profit	Avg. iter.	Time (sec)	Open hubs	Profit	Avg. iter.	Time (sec)	Open hubs
10LL	0.5	11,051	2.01	16	5,9	10,629	2.01	16	5,9	10,129	2.01	19	5,9
	1	10,057	2.01	21	5,9	9,536	2.01	19	5,9	9,336	2.01	78	5,9
10LT	0.5	2,799	2.01	14	5	2,522	2.01	13	5	2,222	2.01	13	5
	1	2,370	2.01	18	5	1,973	2.01	14	5	1,773	2.01	12	5
10TL	0.5	10,805	2.01	15	5,9	10,384	2.01	15	5,9	9,784	2.01	19	5,9
	1	9,720	2.01	21	5,9	9,098	2.01	17	5,9	9,088	2.01	77	5,9
10TT	0.5	2,409	2.01	13	5	2,349	2.01	13	5	2,049	2.01	13	5
	1	2,378	2.01	17	5	2,185	2.01	14	5	1,685	2.01	13	5
20LL	0.5	71,687	13.26	2,342	7,9,10,19	67,929	13.05	1,948	7,9,10,19	55,471	13.36	2,402	7,9,10,19
	1	49,394	9.71	1,526	7,10,19	43,784	9.5	2,257	7,10,19	26,496	7.83	2,130	10,19
20LT	0.5	33,992	8.03	2,265	5,10,12,14	31,514	7.82	1,099	5,10,12,14	24,129	6.10	1,383	5,10,12,14
	1	20,639	6.09	1,802	5,10,12,14	16,736	5.81	2,022	10,12,14	12,166	4.06	285	10,14
20TL	0.5	28,963	3.02	347	7,10	26,602	2.74	154	7,10	19,612	3.07	143	7,10
	1	16,662	3.01	311	7,10	12,667	3.17	194	7,10	9,168	3.08	105	10
20TT	0.5	5,416	2.01	78	10	4,911	2.01	71	10	4,565	2.01	57	10
	1	1,558	2.01	139	10	527	2.01	79	10	436	2.01	60	10
25LL	0.5	95,864	18.53	6,567	7,14,17,23	91,273	18.32	1,783	7,14,17,23	76,665	18.73	5,300	7,14,17,23
	1	67,639	12.63	2,028	7,14,17	60,669	12.42	3,848	7,14,17	38,708	17.63	2,035	9,17
25LT	0.5	59,640	8.02	2,413	6,9,10,12,14,25	55,493	7.91	1,555	6,9,10,12,25	44,722	9.64	2,607	6,9,10,12,25
	1	41,798	6.60	1,886	6,9,12,14,25	36,685	6.39	2,186	9,12,14,25	23,167	6.60	1,762	12,14,25
25TL	0.5	49,639	8.07	1,966	6,9,14	46,153	7.86	1,586	6,9,14	35,093	7.20	1,638	6,9,14
	1	33,417	7.20	1,118	6,9,14	27,273	6.99	2,078	6,9,14	16,152	6.06	860	14
25TT	0.5	17,306	8.03	1,933	6,10,14	15,481	7.82	1,067	10,14	12,515	8.03	1,219	10,14
	1	10,282	7.02	1,329	10,14	8,632	6.81	1,632	14	8,205	2.01	321	14
40LL	0.5	61,336	26.03	21,631	12,22,26,29	58,575	25.82	16,324	12,22,26,29	53,575	24.01	19,324	12,22,26,29
	1	42,520	25.54	20,749	17,26,35	38,203	25.33	19,017	17,26,35	31,203	23.18	18,017	17,26,35
40LT	0.5	48,425	22.13	21,852	10,14,26,30,38	46,080	21.92	18,448	10,14,26,30,38	42,080	17.42	18,448	10,14,26,38
	1	29,070	19.09	20,318	10,17,26, 38	27,736	18.88	21,547	10,26, 38	22,736	15.75	16,547	10,26, 38
40TL	0.5	42,569	16.83	8,116	14,19,29	40,477	16.12	6,903	14,19,29	35,185	10.09	71,032	14,29
	1	25,900	10.96	6,401	14,29	22,730	10.75	9,165	14,29	14,752	8.16	3,367	14,29
40TT	0.5	32,427	10.03	10,331	14,19,25,38	30,583	9.82	7,523	14,19,25,38	26,691	7.55	3,027	14,19,38
	1	20,168	8.43	8,252	14,19,38	17,523	8.22	10,048	14,19,38	11,809	4.24	948	14,38
50LL	0.5	59,282	22.71	14,393	15,28,33,35	57,541	22.5	12,143	15,28,33,35	54,541	16.94	12,143	15,28,33,35
	1	39,066	20.08	16,141	15,28,33,35	36,174	17.27	16,732	15,28,35	31,174	14.18	10,732	15,28,35
50LT	0.5	55,565	19.23	15,057	26,32,46	54,186	16.02	10,436	26,32,46	52,186	13.07	13,436	26,32,46
	1	39,119	17.08	13,326	26,32,46	35,992	9.87	14,681	26,46	30,992	8.74	8,681	26,46
50TL	0.5	34,324	9.13	9,645	26,45	32,431	8.92	5,912	26,45	27,449	8.23	6,113	26,45
	1	17,729	7.73	7,912	24	15,340	4.52	9,448	24	9,882	3.23	2,436	24
50TT	0.5	29,183	4.06	7,299	26,48	27,727	3.85	4,479	26,48	24,434	3.01	4,125	26,48
	1	16,113	3.81	6,729	26,48	13,639	3.6	8,759	26	9,800	3.02	2,403	26
75LL	0.5	93,115	47.37	68,749	14,26,35,38,56	81,343	47.16	65,231	14,26,35,38,56	55,318	41.82	63,194	14,26,38,56
	1	64,054	45.29	64,184	14,26,35,38,56	53,172	41.23	57,418	14,26,38,56	32,719	23.08	56,319	14,38,56
75LT	0.5	74,926	43.16	59,317	25,32,35,38,59	57,343	42.95	53,568	25,35,38,59	41,663	24.76	43,717	25,38,59
	1	40,963	45.83	61,732	25,32,35,38,59	31,568	40.26	55,619	25,35,38,59	24,428	19.80	21,368	25,38
75TL	0.5	52,219	11.76	47,273	14,35,37	39,618	11.21	45,236	14,35,37	25,763	9.96	20,813	14,37
	1	27,165	10.74	45,613	14,35,37	21,308	5.91	34,192	14,37	19,233	2.21	12,341	37
75TT	0.5	42,708	11.63	45,763	26,32,38,59	30,784	11.42	30,118	26,38	20,619	9.59	21,679	26,38
	1	28,816	8.89	31,573	26,38,59	17,193	8.28	29,918	26,38	14,672	2.20	11,972	38

Table 4.2: Computational results for the robust-stochastic model with the max-min profit criterion with 48 instances of the AP dataset.

and name of the instances and the amount of deviation, respectively. The columns labeled "Profit", "Avg. iter.", and "Time (sec)" indicate the optimal expected profit, the average number of iterations required for the convergence of the BD algorithm at each replication of SAA, and the computation time of the instances (in seconds) obtained from solving the model, respectively. The columns labeled "Open hubs" show the locations of the hub nodes. Table 4.2 is split into three parts to represent the results for $\gamma_r = 15\%$, $\gamma_r = 25\%$, and $\gamma_r = 50\%$, respectively.

All of the instances presented in Table 4.2 are solved to optimality. We observe that the computation times and average number of iterations do not vary significantly by varying γ_r values. This can be attributed to the fact that the number of dual variables associated with the intervals of uncertainty is independent from the value of γ_r. This characteristic enables the algorithm to solve instances with up to 16,875 commodities containing stochastic demand and uncertain revenue. The averages of the computational times reported in Table 4.2 for the $\gamma_r = 15\%$, 25%, and 50% instances are 3.8, 3.4, and 2.8 hours, respectively. These results clearly confirm the efficiency of the proposed algorithm for the robust-stochastic model with max-min profit criterion.

The profits obtained from the max-min profit model represent the lowest profit that can be expected, as long as the revenues comply with the model of uncertainty. This profit provides a valuable information, in particular to a conservative decision maker, since the profit associated with this solution will never fall below the obtained value.

Next, we analyze the effects of variability in uncertain revenues on the optimal solutions presented in Table 4.2. When the level of uncertainty (i.e., φ and γ_r) increases, the net profit value and the number of open hubs in the optimal solutions decrease. It can also be observed that the set of open hubs with a high level of uncertainty (e.g., $\gamma_r = 50\%$ and $\varphi = 1$) is a subset of the open hubs, when the level of uncertainty is low (e.g., $\gamma_r = 15\%$ and $\varphi = 0.5$). For example, in the optimal solution of 20LL with $\gamma_r = 50\%$ and $\varphi = 1$, hubs are located at nodes 10 and 19. While, by decreasing γ_r to 15% and φ to 0.5, hubs are located at nodes 7, 9, 10, and 19.

Instance type	φ	$\gamma_r = 15\%$				$\gamma_r = 25\%$				$\gamma_r = 50\%$			
		Demand Class (%)			Avg.	Demand Class (%)			Avg.	Demand Class (%)			Avg.
		m=1	m=2	m=3		m=1	m=2	m=3		m=1	m=2	m=3	
LL	0.5	97.56	84.19	69.38	83.71	87.50	74.75	58.42	73.56	74.89	62.81	44.81	60.84
	1	87.80	75.94	50.83	71.52	78.42	67.10	41.85	62.45	66.58	55.96	31.07	51.20
LT	0.5	91.13	78.44	57.69	75.75	84.29	69.84	48.88	67.67	72.92	57.68	37.97	56.19
	1	83.40	68.46	39.14	63.67	73.65	59.34	30.87	54.62	62.90	49.21	19.89	44.00
TL	0.5	93.69	82.34	61.56	79.20	86.66	71.72	51.08	69.82	75.66	58.29	38.05	57.33
	1	85.69	71.10	40.62	65.80	77.27	66.41	32.07	58.58	65.54	56.78	20.81	47.71
TT	0.5	83.62	72.19	42.38	66.06	76.43	65.68	32.75	58.29	65.53	54.48	21.89	47.30
	1	73.73	63.00	26.98	54.57	64.45	53.98	18.14	45.52	53.02	41.70	8.62	34.45

Table 4.3: Percentage of demand satisfied for each demand class with the max-min profit model.

Table 4.3 presents the percentages of satisfied demand from different market segments. For a given (φ, γ_r) pair, the averages for each demand class are calculated over instances from Table 4.2 with the same type of installation costs and capacities. The average percentages of total satisfied demand are provided in the last columns corresponding to each γ_r value. When φ or γ_r value increases, the percentage of satisfied demand for all three market segments decreases as expected. For a given (φ, γ_r) pair, in the instances with the same configuration of hub installation costs and capacities, the first class is the one with the highest percentages of satisfied demand, while the third demand class has the least. This is because serving the first and third classes result the highest and lowest revenues, respectively. On average, instances with loose capacities (LL and TL) yield higher percentages compared to the instances with tight capacities (LT and TT).

4.3.3 Computational Results for the Robust-Stochastic Model with Min-max Regret Criterion

We now analyze the results obtained with the min-max regret stochastic model. We use instances with up to 75 nodes from the AP dataset. For each instance, we perform two sets of experiments each involving five different scenarios with uncertain revenues (i.e., $|S_r| = 5$). In the first set, revenue scenarios are randomly generated from the interval $[0.75\bar{r}_k^m, \bar{r}_k^m]$, while in the second set, revenue scenarios are drawn from the interval $[0.5\bar{r}_k^m, \bar{r}_k^m]$, where $\bar{r}_k^m$ is the nominal revenue of commodity k of class m.

The computational results are summarized in Table 4.4, which is split into two parts to represent the results for $r_k^{ms} \in [0.75\bar{r}_k^m, \bar{r}_k^m]$ and $r_k^{ms} \in [0.5\bar{r}_k^m, \bar{r}_k^m]$, respectively. In this table, the column "Regret" indicates the optimal regret of the problem and "Avg. profit" represents the average anticipated profits, which are computed by taking the average profits over 50 demand and 5 revenue scenarios in 60 replications. The rest of the columns report the same as in Table 4.2.

All of the instances in Table 4.4 are solved to optimality. The CPU times and average number of iterations required for solving the instances to optimality indicate the efficiency and robustness of the algorithm and also the acceleration techniques proposed for the min-max regret stochastic model. As can be observed from Table 4.4, different revenue intervals have no significant impact on the performance of the algorithm. In particular, the averages of the computational time for $r_k^{ms} \in [0.75\bar{r}_k^m, \bar{r}_k^m]$ and $r_k^{ms} \in [0.5\bar{r}_k^m, \bar{r}_k^m]$ are 3.8 and 3.7 hours, respectively.

The regret values reported in Table 4.4 indicate the maximum amount of profit that can be lost under this data uncertainty. This implies that if the decision maker employs the obtained solution, the anticipated loss in profit is expected to be less than this value. Moreover, the average profits provide an insight on the expected profit itself.

We now analyze the effect of the lower bound of the interval from which revenue scenarios are generated on the optimal solutions. For a given instance, when the lower bound decreases from 0.75 to 0.5, that is, when the range of fluctuations in revenue values increases, the regret of the solution increases, while the average profit and the optimal number of open hubs decrease. This is because with a wider range of fluctuations in the data, we would expect lower revenues, thus the installation cost of fewer hubs can be justified by the expected revenue. It is worthwhile to also note that the set of open hubs, when $r_k^{ms} \in [0.5\bar{r}_k^m, \bar{r}_k^m]$, turned out to be a subset of the open hubs, when $r_k^{ms} \in [0.75\bar{r}_k^m, \bar{r}_k^m]$ in all the instances in Table 4.4.

Instance	$r_k^{ms} \in [0.75\bar{r}_k^m, \bar{r}_k^m]$					$r_k^{ms} \in [0.5\bar{r}_k^m, \bar{r}_k^m]$				
$\lvert N \rvert$	Regret	Avg. profit	Avg. iter.	Time (sec)	Open hubs	Regret	Avg. profit	Avg. iter.	Time (sec)	Open hubs
10LL	139	13,642	2.00	39	5,9	678	8,379	2.00	40	5,9
10LT	565	4,700	2.00	31	5	1,296	1,692	2.00	39	5
10TL	137	11,023	2.00	32	5,9	975	7,942	2.00	31	5,9
10TT	561	3,560	2.00	26	5	1,368	1,799	2.00	31	5
20LL	550	73,309	5.70	2,237	7,9,10,19	2,323	40,839	7.20	1,820	7,10,19
20LT	902	35,684	3.04	1,946	5,10,12,14	1,994	23,234	6.00	1,241	5,10,14
20TL	1,027	29,181	2.03	104	7,10	2,137	21,532	3.00	144	7,10
20TT	1,074	7,231	2.00	91	10	1,799	5,712	2.00	102	10
25LL	145	94,318	2.10	4,563	7,14,17,23	781	70,435	3.84	4,548	7,14,17,23
25LT	383	59,317	2.00	5,394	6,9,10,12,14,25	1,203	41,059	5.03	5,637	6,9,12,14,25
25TL	760	48,941	2.90	2,572	6,9,14	1,094	31,520	4.00	2,776	6,9,14
25TT	979	17,593	4.00	1,774	10,14	1,872	11,451	3.00	457	14
40LL	219	65,561	6.07	22,437	12,22,26,29	935	49,338	6.79	18,717	12,26,29
40LT	189	56,402	4.70	20,214	12,14,26,29,30,38	869	37,804	4.03	15,749	14,26,29,38
40TL	127	48,637	2.00	4,426	14,19,29	929	34,389	2.21	2,616	14,29
40TT	397	37,614	4.01	9,180	14,19,25,38	1,170	25,789	2.16	4,736	14,19,38
50LL	126	64,073	4.09	14,718	15,28,33,35	489	35,848	5.12	15,134	15,28,33,35
50LT	148	60,578	2.00	7,625	6,26,32,46	577	35,812	4.07	7,526	6,26,32,46
50TL	233	37,516	3.00	4,712	26,45	780	15,456	4.40	4,495	26,45
50TT	363	31,974	2.00	5,021	26,48	1,083	13,384	2.00	5,549	26,48
75LL	368	100,106	5.86	72,163	14,23,35,38,56	954	59,933	6.14	74,318	14,23,35,38,56
75LT	464	80,243	4.91	67,335	14,25,32,38,59	1,063	37,002	5.08	67,660	14,25,32,38,59
75TL	534	55,236	4.79	43,307	14,35,37	1,187	24,537	4.36	47,208	14,35,37
75TT	599	46,406	4.15	37,639	26,32,38	1,608	25,810	4.09	41,314	26,32,38

Table 4.4: Computational results for the min-max regret stochastic model with 24 instances of the AP dataset.

Table 4.5 presents the percentages of satisfied demand from different market segments for $r_k^{ms} \in [0.75\bar{r}_k^m, \bar{r}_k^m]$ and $[0.5\bar{r}_k^m, \bar{r}_k^m]$. When the lower bound of the interval decreases, the

Instance type	$r_k^{ms} \in [0.75\bar{r}_k^m, \bar{r}_k^m]$				$r_k^{ms} \in [0.5\bar{r}_k^m, \bar{r}_k^m]$			
	Demand Class (%)			Avg.	Demand Class (%)			Avg.
	m=1	m=2	m=3		m=1	m=2	m=3	
LL	97.19	83.34	48.67	76.40	93.21	74.66	37.56	68.48
LT	89.90	70.80	36.68	65.79	84.02	58.25	26.97	56.41
TL	92.21	71.38	38.35	67.31	88.07	62.51	28.05	59.55
TT	76.90	47.51	17.96	47.46	66.09	32.70	12.56	37.12

Table 4.5: Percentage of demand satisfied for each demand class with the min-max regret model.

percentage of satisfied demand for all three market segments also decreases. Similar to our observations with Table 4.3, in the instances with the same configuration, the first and third classes result in the highest and lowest percentages of satisfied demand, respectively. For a given revenue interval, the percentage of satisfied demand in the instances with loose capacities, on average, is higher than that of the instances with tight capacities.

4.3.4 Comparison of Stochastic and Robust-Stochastic Solutions

In this section, we compare the solutions obtained from the stochastic and the two robust-stochastic models to analyze the effect of uncertain revenues. For the stochastic model, we use the nominal revenues (i.e., $\bar{r}_k^m$). For the robust-stochastic models, the set of revenue scenarios considered in the min-max regret model (i.e., S_r) should comply with the requirements of the uncertainty sets considered in the max-min profit model (i.e., φ and γ_r). Recall that in the max-min version, we use φ to determine the variability in uncertain revenue, such that $\hat{r}_k^m \sim U[0, \varphi\bar{r}_k^m]$ and the interval of uncertainty for revenue is $[\bar{r}_k^m - \hat{r}_k^m, \bar{r}_k^m]$. To make the settings under which the two models are executed comparable, for the min-max regret stochastic model, we generate ten revenue scenarios (i.e., $|S_r| = 10$) using the same intervals as in the max-min version (i.e., $r_k^{ms} \in [\bar{r}_k^m - \hat{r}_k^m, \bar{r}_k^m]$ for each $s \in S_r$) and implicitly satisfy the budget of uncertainty constraint (4.29) by ensuring that $\sum\limits_k \sum\limits_m \frac{\bar{r}_k^m - r_k^{ms}}{\hat{r}_k^m} \leq \gamma_r$, for each scenario $s \in S_r$.

Note that each model optimizes a different metric, hence, we cannot compare the quality of the solutions based on the individual objective function values. However, we can compare the hub networks obtained from each of the models. We suggest evaluating the quality of the solutions under two metrics: the profit that can be expected from each solution, and the frequency at which each solution attains the highest profit among other solutions.

Let $(\tilde{x}, \tilde{y})$ denote the optimal solution obtained from any of the stochastic or robust-stochastic models. For revenue $r_k^m \in [\bar{r}_k^m - \hat{r}_k^m, \bar{r}_k^m]$, the total profit associated with a solution $(\tilde{x}, \tilde{y})$ is calculated as:

$$\begin{aligned} Z_r(\tilde{x}, \tilde{y}) &= \mathbb{E}_\xi[\sum_{k \in K} \sum_{m \in M} \sum_{a \in A_k} (r_k^m - \hat{C}_{ak}) w_k^m(\xi) \tilde{x}_{ak}^m(\xi)] - \sum_{i \in H} \tilde{y}_i f_i \\ &= \sum_{k \in K} \sum_{m \in M} r_k^m \tilde{X}_k^m - \tilde{C} \end{aligned} \tag{4.46}$$

where

$$\tilde{X}_k^m = \mathbb{E}_\xi[\sum_{a \in A_k} w_k^m(\xi) \tilde{x}_{ak}^m(\xi)] \text{ and } \tilde{C} = \mathbb{E}_\xi[\sum_{k \in K} \sum_{m \in M} \sum_{a \in A_k} \hat{C}_{ak} w_k^m(\xi) \tilde{x}_{ak}^m(\xi)] + \sum_{i \in H} \tilde{y}_i f_i.$$

Recall that r_k^m is a random variable (with unknown distribution). Therefore, by central limit theorem, $Z_r(\tilde{x}, \tilde{y})$ is a normally distributed random variable with expected value

$$\mathbb{E}_r[Z_r(\tilde{x}, \tilde{y})] = \sum_{k \in K} \sum_{m \in M} \mathbb{E}[r_m^k] \tilde{X}_k^m - \tilde{C}, \tag{4.47}$$

and variance

$$\mathbb{V}_r[Z_r(\tilde{x}, \tilde{y})] = \sum_{k \in K} \sum_{m \in M} \mathbb{V}[r_m^k] (\tilde{X}_k^m)^2. \tag{4.48}$$

The first metric can be computed using (4.47). In our experiments, we adapt the SAA scheme and use $|\mathcal{N}| = 50$ demand scenarios. Moreover, we assume that $r_k^m \sim U[\bar{r}_k^m - \hat{r}_k^m, \bar{r}_k^m]$.

Therefore, $\mathbb{E}_r[Z(\tilde{x}, \tilde{y})]$ and $\mathbb{V}_r[Z(\tilde{x}, \tilde{y})]$ can easily be computed by setting $\mathbb{E}[r_m^k] = \bar{r}_k^m - 0.5\hat{r}_k^m$ and $\mathbb{V}[r_m^k] = \frac{1}{12}(\hat{r}_k^m)^2$.

The second metric estimates the probability that a solution dominates other two solutions in terms of the profit that can be obtained under different revenue realizations. Let $(\tilde{x}_S, \tilde{y}_S)$, $(\tilde{x}_{R1}, \tilde{y}_{R1})$, and $(\tilde{x}_{R2}, \tilde{y}_{R2})$ be the solutions obtained by the stochastic model with nominal revenue, robust-stochastic model with max-min profit criterion and robust-stochastic model with min-max regret criterion, respectively. The probability that solution obtained by the stochastic model yields the highest profit among the three models is computed by

$$\mathbb{P}[Z_r(\tilde{x}_S, \tilde{y}_S) > Z_r(\tilde{x}_{R1}, \tilde{y}_{R1}), Z_r(\tilde{x}_S, \tilde{y}_S) > Z_r(\tilde{x}_{R2}, \tilde{y}_{R2})]. \tag{4.49}$$

The probability that other two solutions yield the highest profits are defined similarly. Note that although random variables $Z_r(.)$ are normally distributed with known parameters, these random variables are not independent and so computing the probability defined by (4.49) is not straightforward. To approximate these probabilities, we generate a large sample of revenue scenarios (here of size 1,000) and estimate the percentages by counting the realizations under which each model outperforms the other two.

We performed the above analysis using four instances from the AP dataset: 20LL, 25LT, 40TL, and 50LL. For each instance, we considered five levels of variability for the revenue intervals, $\varphi \in \{0.2, 0.4, 0.6, 0.8, 1\}$, and two values for the budget of uncertainty with $\gamma_r \in \{25\%, 50\%\}$. Computational results are reported in Table 4.6. Columns under the heading of "Expected profit" represent the expected profit obtained from the stochastic, robust-stochastic model with max-min profit criterion, and min-max regret stochastic model, respectively. Columns under the heading "Dominance probability (%)", on the other hand, provide the percentage that each model yields the highest profit among the three models under 1,000 replica. The bold entries in Table 4.6 highlight the highest expected profit for each instance.

The solutions obtained from both of the robust-stochastic models substantially outperform the solutions obtained from the stochastic model based on the nominal revenues, in general. On average, the robust-stochastic models yield significantly higher expected profits compared to the stochastic model. More specifically, in our experiments, there is a 97% chance that at least one of the robust-stochastic models dominates the stochastic model with nominal revenues. The robustness of the solutions obtained from the robust-stochastic models increases as the uncertainty in the revenues (i.e. φ) increases. These observations underline the robustness of the solutions obtained from the robust-stochastic models over the stochastic model and justify the need for incorporating both sources of uncertainty in decision making.

When the two robust-stochastic models are compared with each other, the min-max regret model turned out to be more likely to yield the highest profit. In particular, the min-max regret model attained the highest profit in 71% of the instances whereas the max-min profit model attained the highest profit only in 22%. It can also be observed from the results reported in Table 4.6 that increasing the budget of uncertainty from $\gamma_r = 25\%$ to 50% results in a higher expected profit for the robust-stochastic models, on average. This suggests a positive effect of increasing the level of conservatism on the quality of the solutions obtained from the robust-stochastic models. Increasing the level of conservatism plays a role in favour, in particular, of the min-max regret model, as it not only increases the expected profit, but also increases the chance that this model yields the highest profit among the three models.

Instance		$\gamma_r = 25\%$						$\gamma_r = 50\%$					
		Expected profit			Dominance probability (%)			Expected profit			Dominance probability (%)		
$\lvert N\rvert$	φ	Nominal revenue	Max-min profit	Min-max regret	Nominal revenue	Max-min profit	Min-max regret	Nominal revenue	Max-min profit	Min-max regret	Nominal revenue	Max-min profit	Min-max regret
20LL	0.2	42,746	**42,791**	42,748	22.7	54.5	22.8	42,746	42,899	**43,063**	0.1	8.0	91.9
	0.4	32,896	34,671	**36,210**	0.0	0.0	100.0	32,896	35,232	**36,396**	0.0	0.0	100.0
	0.6	23,047	26,225	**29,678**	0.0	0.0	100.0	23,047	27,950	**29,879**	0.0	0.0	100.0
	0.8	13,197	18,720	**23,180**	0.0	0.0	100.0	13,197	18,985	**23,602**	0.0	0.0	100.0
	1	3,348	14,087	**16,769**	0.0	2.4	97.6	3,348	16,298	**17,595**	0.0	12.9	87.1
25LT	0.2	80,587	80,373	**80,609**	23.3	0.0	76.7	80,587	80,425	**80,655**	4.8	0.0	95.2
	0.4	69,063	67,812	**69,225**	9.1	0.0	90.9	69,063	65,583	**69,324**	2.4	0.0	97.6
	0.6	57,539	53,817	**57,988**	1.9	0.0	98.1	57,539	52,355	**58,038**	17.9	0.0	82.1
	0.8	46,015	42,143	**46,948**	0.4	0.0	99.6	46,015	40,420	**49,117**	0.0	0.0	100.0
	1	34,491	34,108	**36,131**	0.2	3.5	96.3	34,491	34,634	**41,383**	0.0	0.0	100.0
40TL	0.2	52,727	52,659	52,830	0.0	0.0	100.0	52,727	52,666	**52,839**	0.0	0.0	100.0
	0.4	45,753	45,488	45,935	0.0	0.0	100.0	45,753	45,363	**45,950**	0.0	0.0	100.0
	0.6	38,779	**40,131**	39,142	0.0	100.0	0.0	38,779	**39,864**	39,541	0.0	95.6	4.4
	0.8	31,805	31,531	**32,664**	0.0	0.0	100.0	31,805	31,905	**36,304**	0.0	0.0	100.0
	1	24,831	23,562	**26,216**	0.0	0.0	100.0	24,831	24,353	**30,462**	0.0	0.0	100.0
50LL	0.2	54,568	**54,692**	54,681	0.0	83.6	16.4	54,568	54,684	**54,705**	0.0	0.6	99.4
	0.4	47,232	47,354	**47,449**	0.0	3.3	96.7	47,232	47,485	**47,519**	0.0	27.1	72.9
	0.6	39,897	**40,279**	40,181	0.0	77.4	22.6	39,897	**42,151**	40,762	0.0	100.0	0.0
	0.8	32,562	**37,606**	33,286	0.0	100.0	0.0	32,562	**38,646**	36,247	0.0	100.0	0.0
	1	25,226	**28,177**	26,323	0.0	100.0	0.0	25,226	**31,280**	29,937	0.0	99.9	0.1
Avg.		39815	40811	**41910**	2.9	26.2	70.9	39815	41159	**43166**	1.3	22.2	76.5

Table 4.6: Profit comparison with stochastic and robust-stochastic models.

4.4 Conclusions

In this chapter, to capture more realistic cases, we extended the profit-maximizing hub location models to cases where demand or revenue are uncertain. We addressed demand uncertainty by developing a two-stage stochastic program. We further extended the stochastic model to robust-stochastic models to address the uncertainty in the revenues. To incorporate uncertain revenues into the problem, we employed robust optimization techniques and considered two particular cases including interval representation with a max-min profit criterion and discrete scenarios using a min-max regret objective. We proposed mixed integer programming formulations for each of these cases and showed that the robust-stochastic version with max-min profit criterion can be viewed as a special case of the min-max regret stochastic model.

For the stochastic and robust-stochastic problems, we presented a solution method that integrates the proposed Benders decomposition algorithms with the SAA scheme to obtain solutions to problems with a large number of scenarios. We additionally developed novel acceleration techniques to enhance the performance of the algorithms enabling them to solve large-scale intractable instances of the stochastic and robust-stochastic problems. Integrating BD2 with the accelerated SAA algorithm, instances involving up to 75 nodes and 16,875 commodities were solved to optimality. These results clearly confirm the efficiency and robustness of our algorithms.

We additionally compared the quality of the solutions obtained from the stochastic and robust-stochastic models. The results provide several important insights in the design of optimal hub networks to maximize profit. For example, the expected profit obtained from both of the robust-stochastic models is significantly higher than that of the stochastic model. These observations verify that the robust-stochastic models indeed provide robust solutions and justify the need for embedding both sources of uncertainty.

Chapter 5

Contributions and Concluding Remarks

Benders decomposition is a prevalent large-scale optimization technique for tackling challenging problems that lie at the heart of operations research and supply chain management. This thesis contributed to the literature in this area in three main directions. First, we presented general techniques for accelerating Benders decomposition algorithm. Second, we illustrated applicability of Benders decomposition in a new profit maximizing hub location problem. Third, we introduced a novel modelling approach for taking uncertainties in designing hub networks and introduced techniques for efficiently solving these problems. Contributions of this thesis in each direction are summarized as follows.

In Chapter 2, we proposed a general method for selecting Benders cuts to enhance convergence of the BD algorithm. Our approach explicitly takes cut depth into account through Euclidean distance from the master solution to the candidate cuts. We then extended this measure to general ℓ_p-norms, and unveiled their properties from a primal perspective. We also presented reformulation techniques for deriving these cuts by solving the separation problems as linear or quadratic programs. We further established a duality between separation and projection, and through this duality, we introduced the Guided Projections Algorithm for producing ℓ_p-deepest cuts.

Through a generalization of notion of distance, we illustrated the connection of our method to some well-known cut selection strategies. Specifically, we established the connection to MIS cuts, and provided three novel ways of choosing the normalization coefficients in the MIS subproblem, that connect our distance functions to the Magnanti-Wong procedure for producing Pareto-optimal cuts, as well as the Conforti-Wolsey procedure for producing facet-defining cuts. As a proof of concept, we performed computational experiments on CFLP instances, and we showed the benefits of deepest cuts and other distance-based cuts, particularly when generated using GPA, in decreasing the number of cuts as well as the runtime of the BD algorithm.

In Chapter 3, we considered revenue management decisions within hub location problems and proposed MILP formulations of the problem. We presented a fast implementation of Benders decomposition equipped with efficient and tailored routines for producing effective optimality cuts leveraging the combinatorial structures of the hub location problem. In our approach, referred to as the two-phase cut generation, we introduced a framework for decomposing the subproblem into two sequential problems, where the first phase guarantees optimality of the solution, and the second phase strengthens the cut while maintaining optimality. We introduced two routines for the seconds phase by treating this problem as a multiple objective optimization problem. The first routine is based on solving the subproblem as a set of continuous maximum weighted matching problems, while the second routine solves the subproblem as a series of continuous knapsack problems. We further enhanced the BD algorithm by incorporating improved variable fixing techniques.

Through computational experiments, we showed that both cut selection routines result in generating effective cuts, outperforming the state-of-the-art cuts from the literature (Pareto-optimal cuts; Magnanti and Wong 1981) in terms of effectiveness in closing the optimality gap as well as computational efforts required for producing them, offering orders of magnitude speedups. Our two-phase cut generation routine as well as the improved variable fixing proposed can be used for solving several classes of transportation and network optimization problems. We evaluated the proposed Benders decomposition algorithms through extensive computational experiments, and illustrated the capability of our method in solving instances with up to 500 nodes, exceeding the largest instances solved to date (300 nodes; Contreras et al. 2012), while considering an even more difficult problem setting with generic capacity constraints, multiple demand segments, and a profit maximizing objective function.

In Chapter 4, we extended the profit maximizing hub location problems by considering several types of uncertainty in the parameters. We first proposed a two-stage stochastic program for considering uncertain demand. We further extended the model by incorporating uncertain revenue through robust-stochastic formulations. We employed two robust optimization techniques based on interval representation with a max-min robustness criterion as well as discrete scenarios using a min-max regret objective. To the best of our knowledge, this is the first study that compares two robust approaches under a stochastic setting. The proposed modeling techniques can be used to formulate uncertainty in other types of optimization problems. We proposed MILP formulations for theses three cases and developed sample average approximation scheme to obtain solutions to problems with large number of scenarios. We further employed Benders decomposition algorithms tailored to the structures of each problem for solving large-scale instances. Leveraging the repetitive structures of these problems, we proposed novel acceleration techniques to improve the convergence of the algorithms. Our acceleration techniques can be employed in a more general context.

We performed extensive computational analysis and illustrated the benefits of these techniques in reducing the computation times by a factor of more than five, enabling our algorithm to solve large-scale intractable instances of this problem. We showed that our methods were able to solve instances with up to 75 nodes and 16,875 commodities of multiple demand classes, surpassing are the largest instances solved for any type of stochastic hub location problems. Our results further provided several important insights in the design of hub networks. We showed that the expected profit obtained from both of the robust-stochastic models are significantly higher than that of the stochastic model. We also observed that the uncertainty associated with the revenues resulted in building denser hub networks with a higher number of allocation connections. These observations justify the need for embedding both sources of uncertainty for designing robust hub networks.

Bibliography

Adulyasak Y, Cordeau JF, Jans R (2015) Benders decomposition for production routing under demand uncertainty. *Operations Research* 63(4):851–867.

Aissi H, Bazgan C, Vanderpooten D (2009) Min–max and min–max regret versions of combinatorial optimization problems: A survey. *European Journal of Operational Research* 197(2):427–438.

Alibeyg A, Contreras I, Fernández E (2016) Hub network design problems with profits. *Transportation Research Part E: Logistics and Transportation Review* 96:40–59.

Alibeyg A, Contreras I, Fernández E (2018) Exact solution of hub network design problems with profits. *European Journal of Operational Research* 266(1):57–71.

Alshamsi A, Diabat A (2018) Large-scale reverse supply chain network design: An accelerated Benders decomposition algorithm. *Computers & Industrial Engineering* 124:545–559.

Alumur S, Kara BY (2008) Network hub location problems: The state of the art. *European Journal of Operational Research* 190(1):1–21.

Balas E, Ceria S, Cornuéjols G (1993) A lift-and-project cutting plane algorithm for mixed 0–1 programs. *Mathematical Programming* 58(1-3):295–324.

Balas E, Zemel E (1980) An algorithm for large zero-one knapsack problems. *Operations Research* 28(5):1130–1154.

Bayram V, Yaman H (2017) Shelter location and evacuation route assignment under uncertainty: A Benders decomposition approach. *Transportation Science* 52(2):416–436.

Beasley J (2021) ORLIB. `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/capinfo.html`.

Beasley JE (1990) OR Library: Hub location. http://people.brunel.ac.uk/mastjjb/jeb/ orlib/phubinfo.html.

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

Benders JF (1962) Partitioning procedures for solving mixed-variables programming problems. *Numerische mathematik* 4(1):238–252.

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

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

Bodur M, Dash S, Günlük O, Luedtke J (2016) Strengthened Benders cuts for stochastic integer programs with continuous recourse. *INFORMS Journal on Computing* 29(1):77–91.

Bodur M, Luedtke JR (2016) Mixed-integer rounding enhanced Benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty. *Management Science* 63(7):2073–2091.

Boland N, Krishnamoorthy M, Ernst AT, Ebery J (2004) Preprocessing and cutting for multiple allocation hub location problems. *European Journal of Operational Research* 155(3):638–653.

Bonami P, Salvagnin D, Tramontani A (2020) Implementing automatic Benders decomposition in a modern MIP solver. *International Conference on Integer Programming and Combinatorial Optimization*, 78–90 (Springer).

Burkard RE, Dell'Amico M, Martello S (2009) *Assignment problems*, volume 125 (SIAM).

Cadoux F (2010) Computing deep facet-defining disjunctive cuts for mixed-integer programming. *Mathematical Programming* 122(2):197–223.

Camargo RS, Miranda Jr G, Luna H (2008) Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & Operations Research* 35(4):1047–1064.

Campbell JF (1994) Integer programming formulations of discrete hub location problems. *European Journal of Operational Research* 72(2):387–405.

Campbell JF, Ernst AT, Krishnamoorthy M (2002) Hub location problems. Drezner Z, Hamacher HW, eds., *Facility Location: Applications and Theory*, 373–407 (Springer).

Charnes A, Cooper WW (1962) Programming with linear fractional functionals. *Naval Research Logistics Quarterly* 9(3-4):181–186.

Cho SH, Jang H, Lee T, Turner J (2014) Simultaneous location of trauma centers and helicopters for emergency medical service planning. *Operations Research* 62(4):751–771.

Codato G, Fischetti M (2006) Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research* 54(4):756–766.

Conforti M, Wolsey LA (2019) "facet" separation with one linear program. *Mathematical Programming* 178(1):361–380.

Contreras I (2015) Hub location problems. *Location Science*, 311–344 (Springer).

Contreras I, Cordeau JF, Laporte G (2011a) Benders decomposition for large-scale uncapacitated hub location. *Operations Research* 59(6):1477–1490.

Contreras I, Cordeau JF, Laporte G (2011b) Stochastic uncapacitated hub location. *European Journal of Operational Research* 212(3):518–528.

Contreras I, Cordeau JF, Laporte G (2012) Exact solution of large-scale hub location problems with multiple capacity levels. *Transportation Science* 46(4):439–459.

Cornuéjols G, Sridharan R, Thizy JM (1991) A comparison of heuristics and relaxations for the capacitated plant location problem. *European Journal of Operational Research* 50(3):280–297.

Correia I, Saldanha-da Gama F (2015) Facility location under uncertainty. *Location Science*, 177–203 (Springer).

Dantzig GB (1957) Discrete-variable extremum problems. *Operations Research* 5(2):266–288.

de Sá EM, de Camargo RS, de Miranda G (2013) An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research* 226(2):185–202.

Drezner Z, Hamacher HW (2001) *Facility location: applications and theory* (Springer Science & Business Media).

Ebery J, Krishnamoorthy M, Ernst A, Boland N (2000) The capacitated multiple allocation hub location problem: Formulations and algorithms. *European Journal of Operational Research* 120(3):614–631.

Ernst AT, Krishnamoorthy M (1996) Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science* 4(3):139–154.

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

Fischetti M, Ljubić I, Sinnl M (2017) Redesigning Benders decomposition for large-scale facility location. *Management Science* 63(7):2146–2162.

Fischetti M, Salvagnin D, Zanette A (2010) A note on the selection of Benders cuts. *Mathematical Programming* 124(1-2):175–182.

Fontaine P, Minner S (2018) Benders decomposition for the hazmat transport network design problem. *European Journal of Operational Research* 267(3):996–1002.

Fortz B, Poss M (2009) An improved benders decomposition applied to a multi-layer network design problem. *Operations Research Letters* 37(5):359–364.

Galil Z (1986) Efficient algorithms for finding maximum matching in graphs. *ACM Computing Surveys (CSUR)* 18(1):23–38.

Geoffrion AM (1972) Generalized Benders decomposition. *Journal of Optimization Theory and Applications* 10(4):237–260.

Hooker JN, Ottosson G (2003) Logic-based Benders decomposition. *Mathematical Programming* 96(1):33–60.

Keyvanshokooh E, Ryan SM, Kabir E (2016) Hybrid robust and stochastic optimization for closed-loop supply chain network design using accelerated Benders decomposition. *European Journal of Operational Research* 249(1):76–92.

Khassiba A, Bastin F, Cafieri S, Gendron B, Mongeau M (2020) Two-stage stochastic mixed-integer programming with chance constraints for extended aircraft arrival management. *Transportation Science* 54(4):897–919.

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

Lin CC, Lee SC (2018) Hub network design problem with profit optimization for time-definite LTL freight transportation. *Transportation Research Part E: Logistics and Transportation Review* 114:104–120.

Magnanti TL, Wong RT (1981) Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research* 29(3):464–484.

Maheo A, Kilby P, Van Hentenryck P (2017) Benders decomposition for the design of a hub and shuttle public transit system. *Transportation Science* 53(1):77–88.

Meraklı M, Yaman H (2016) Robust intermodal hub location under polyhedral demand uncertainty. *Transportation Research Part B: Methodological* 86:66–85.

Meraklı M, Yaman H (2017) A capacitated hub location problem under hose demand uncertainty. *Computers & Operations Research* 88:58–70.

Mercier A (2008) A theoretical comparison of feasibility cuts for the integrated aircraft-routing and crew-pairing problem. *Transportation Science* 42(1):87–104.

Naderi B, Roshanaei V, Begen MA, Aleman DM, Urbach DR (2021) Increased surgical capacity without additional resources: Generalized operating room planning and scheduling. *Production and Operations Management* .

Papadakos N (2008) Practical enhancements to the Magnanti–Wong method. *Operations Research Letters* 36(4):444–449.

Papadakos N (2009) Integrated airline scheduling. *Computers & Operations Research* 36(1):176–195.

Pearce RH, Forbes M (2018) Disaggregated Benders decomposition and branch-and-cut for solving the budget-constrained dynamic uncapacitated facility location and network design problem. *European Journal of Operational Research* 270(1):78–88.

Rahimi A, Gönen M (2021) Efficient multitask multiple kernel learning with application to cancer research. *IEEE Transactions on Cybernetics* .

Rahmaniani R, Crainic TG, Gendreau M, Rei W (2017) The benders decomposition algorithm: A literature review. *European Journal of Operational Research* 259(3):801–817.

Rahmaniani R, Crainic TG, Gendreau M, Rei W (2018) Accelerating the Benders decomposition method: Application to stochastic network design problems. *SIAM Journal on Optimization* 28(1):875–903.

Rodriguez-Martin I, Salazar-Gonzalez JJ (2008) Solving a capacitated hub location problem. *European Journal of Operational Research* 184(2):468–479.

Saharidis GK, Ierapetritou MG (2010) Improving Benders decomposition using maximum feasible subsystem (MFS) cut generation strategy. *Computers & Chemical Engineering* 34(8):1237–1245.

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.

Schütz P, Tomasgard A, Ahmed S (2009) Supply chain design under uncertainty using sample average approximation and dual decomposition. *European Journal of Operational Research* 199(2):409–419.

Shapiro A, Homem-de Mello T (1998) A simulation-based approach to two-stage stochastic programming with recourse. *Mathematical Programming* 81(3):301–325.

Sherali HD, Lunday BJ (2013) On generating maximal nondominated Benders cuts. *Annals of Operations Research* 210(1):57–72.

Sim T, Lowe TJ, Thomas BW (2009) The stochastic p-hub center problem with service-level constraints. *Computers & Operations Research* 36(12):3166–3177.

Taherkhani G, Alumur SA, Hosseini M (2020) Benders decomposition for the profit maximizing capacitated hub location problem with multiple demand classes. *Transportation Science* 54(6):1446–1470.

Taherkhani G, Alumur SA, Hosseini M (2021) Robust stochastic models for profit-maximizing hub location problems. *Transportation Science* 55(6):1322–1350.

Tibshirani R (1996) Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1):267–288.

Yang TH (2009) Stochastic air freight hub location and flight routes planning. *Applied Mathematical Modelling* 33(12):4424–4430.

Appendix A

Supplementary Materials for Chapter 2

A.1 Proofs

In this appendix, we provide the proof of the propositions and theorems given in Chapter 2. For convenience, we formally restate the propositions and theorems as well.

Proposition 2.1. Solution (y, η) satisfies constraints (2.5) if and only if $(y, \eta) \in \mathcal{E}$.

Proof. We first show that any $(y, \eta) \in \mathcal{E}$ satisfies all constraints (2.5). It suffices to show that constraints (2.5) are implied by the classical Benders cuts. For an arbitrary certificate (π, π_0), we may consider two cases:

Case 1: For $\pi_0 > 0$, define $u = \frac{\pi}{\pi_0}$. Then, $u \geq 0$ and $u^\top A \leq c^\top$, implying $u \in \mathcal{U}$. Consequently, by Minkowski's representation theorem, u is a convex combination of extreme points of $\mathcal{U}$ and a weighted combination of extreme rays of $\mathcal{U}$, which implies that constraint $0 \geq \pi^\top(b - By) + \pi_0(f^\top y - \eta)$ is implied by the classical Benders feasibility and optimality cuts.

Case 2: For $\pi_0 = 0$, we have $\pi^\top A \leq 0$, which means π is in the recession cone of $\mathcal{U}$. Consequently, constraint $0 \geq \pi^\top(b - By) + \pi_0(f^\top y - \eta) = \pi^\top(b - By)$ is implied by the classical Benders feasibility cuts.

Next, we show that any (y, η) satisfying constraints (2.5) belongs to $\mathcal{E}$. Observe that any extreme point u of $\mathcal{U}$ can be represented as a certificate $(\pi, \pi_0) = (u, 1)$. Similarly, any extreme ray v of $\mathcal{U}$ can be represented as a certificate $(\pi, \pi_0) = (v, 0)$. Consequently, constraint set (2.5) contains all classical Benders optimality and feasibility cuts. □

Proposition 2.2. Given $q \geq 1$ and $\hat{z} \in \mathbb{R}^{n+1}$, the minimum ℓ_q-distance from the point $\hat{z}$ to the points on the hyperplane $\alpha^\top z + \beta = 0$ is

$$\min_{z:\alpha^\top z+\beta=0} \|z - \hat{z}\|_q = \frac{|\alpha^\top \hat{z} + \beta|}{\|\alpha\|_p},$$

where ℓ_p is the dual norm of ℓ_q (i.e., $\frac{1}{p} + \frac{1}{q} = 1$).

Proof. For generality, we prove the proposition for general norms using the definition of dual norms; proof for ℓ_p norms follows directly. By definition of dual norms, we have

$$\|\alpha\|_* = \max_x \left\{ \frac{|\alpha^\top x|}{\|x\|} \right\}.$$

Replacing $x = z - \hat{z}$, we get

$$\|\alpha\|_* = \max_z \left\{ \frac{|\alpha^\top (z - \hat{z})|}{\|z - \hat{z}\|} \right\}. \tag{A.1}$$

For $z \in \mathbb{R}^{n+1} \setminus \{\hat{z}\}$, define $\tilde{z}(z)$ to be the intersection of hyperplane $\alpha^\top z + \beta = 0$ and the line that crosses points $(z, \hat{z})$. Note that the intersection point for any optimal z exists, since the line crossing $(z, \hat{z})$ cannot be parallel to the hyperplane $\alpha^\top z + \beta = 0$ for optimal z. This is because a parallel line crossing $(z, \hat{z})$ and hyperplane $\alpha^\top z + \beta = 0$ would imply that $\alpha^\top (z - \hat{z}) = 0$, which cannot be optimal, since $\|\alpha\|_* > 0$. Now, since $\hat{z}$ does not belong to the hyperplane $\alpha^\top z + \beta = 0$, there exists $\theta(z) \neq 0$ such that $z - \hat{z} = \theta(z) \times (\tilde{z}(z) - \hat{z})$. We can therefore rewrite (A.1) as

$$\|\alpha\|_* = \max_z \left\{ \frac{|\theta(z)||\alpha^\top (\tilde{z}(z) - \hat{z})|}{\|\theta(z) \times (\tilde{z}(z) - \hat{z})\|} \right\} = \max_z \left\{ \frac{|\alpha^\top (\tilde{z}(z) - \hat{z})|}{\|(\tilde{z}(z) - \hat{z})\|} \right\}, \tag{A.2}$$

where the last equality holds since norms are homogeneous. Consequently, without loss of generality we may restrict z to the points on the hyperplane $\alpha^\top z + \beta = 0$, that is

$$\|\alpha\|_* = \max_{z : \alpha^\top z + \beta = 0} \left\{ \frac{|\alpha^\top (z - \hat{z})|}{\|z - \hat{z}\|} \right\} = \max_{z : \alpha^\top z + \beta = 0} \left\{ \frac{|\alpha^\top \hat{z} + \beta|}{\|z - \hat{z}\|} \right\}, \tag{A.3}$$

where we have used $\beta = -\alpha^\top z$. But $|\alpha^\top \hat{z} + \beta|$ is constant, therefore we may rewrite (A.3) as

$$\|\alpha\|_* = |\alpha^\top \hat{z} + \beta| \max_{z : \alpha^\top z + \beta = 0} \left\{ \frac{1}{\|z - \hat{z}\|} \right\} = \frac{|\alpha^\top \hat{z} + \beta|}{\min\limits_{z : \alpha^\top z + \beta = 0} \|z - \hat{z}\|}, \tag{A.4}$$

which completes the proof for general norm. The proof for ℓ_q follows by replacing $\|\cdot\| = \|\cdot\|_q$ and $\|\cdot\|_* = \|\cdot\|_p$. □

Theorem 2.1. Separation problem (2.9) is equivalent to the following Lagrangian dual problem.

$$
\begin{aligned}
[\textit{Primal SSP}] \quad \min \quad & \|(y-\hat{y}, \eta - \hat{\eta})\|_q \\
\text{s.t.} \quad & \eta \geq c^\top x + f^\top y \\
& Ax \geq b - By \\
& x \geq 0,
\end{aligned}
\tag{A.5}
$$

in which (y, x, η) are the variables and ℓ_q is the dual norm of ℓ_p.

Proof. SSP (2.9) can be equivalently stated as (see Proposition 2.9):

$$
\begin{aligned}
\max_{(\pi,\pi_0)\in\Pi} \quad & \pi^\top(b - B\hat{y}) + \pi_0(f^\top \hat{y} - \hat{\eta}) \\
\text{s.t.} \quad & \|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_p \leq 1.
\end{aligned}
\tag{A.6}
$$

In the following, we prove the statement for $p < \infty$, since for $p = \infty$ the dual can be directly derived using LP duality by reformulating (A.6) as an LP (see Section 2.3.1). For $p < \infty$, $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_p \leq 1$ is equivalent to $\pi_0^p + \sum_{j=1}^n |\pi_0 f_j - \pi^\top B_{.j}|^p \leq 1$, where $B_{.j}$ is the j'th column of matrix B. Hence, introducing non-negative variables τ_j, $j = 1, \ldots, n$, we may restate (A.6) as

$$
\begin{aligned}
\max \quad & \pi^\top(b - B\hat{y}) + \pi_0(f^\top \hat{y} - \hat{\eta}) && \text{(A.7)} \\
\text{s.t.} \quad & \pi^\top A - \pi_0 c^\top \leq 0 && \text{(A.8)} \\
& \pi_0^p + \sum_{j=1}^n \tau_j^p \leq 1 && \text{(A.9)} \\
& \pi_0 f^\top - \pi^\top B \leq \tau^\top && \text{(A.10)} \\
& \pi^\top B - \pi_0 f^\top \leq \tau^\top && \text{(A.11)} \\
& \pi \geq 0, \tau \geq 0, \pi_0 \geq 0. && \text{(A.12)}
\end{aligned}
$$

Assigning Lagrange multipliers x, δ_0, y^- and y^+ respectively to constraints (A.8)-(A.11), we can derive the following Lagrangian dual problem

$$\begin{aligned} \min \quad & \|(y^+ + y^-, \delta_0)\|_q && \text{(A.13)} \\ \text{s.t.} \quad & c^\top x \le \hat{\eta} + \delta_0 - f^\top(\hat{y} + y^+ - y^-) && \text{(A.14)} \\ & Ax \ge b - B(\hat{y} + y^+ - y^-) && \text{(A.15)} \\ & x \ge 0, \delta_0 \ge 0, y^+ \ge 0, y^- \ge 0. && \text{(A.16)} \end{aligned}$$

The dual problem (A.5) is derived by replacing $y = \hat{y} + y^+ - y^-$ and noting that minimizing $y_j^+ + y_j^-$ is equivalent to minimizing $|y_j - \hat{y}_j|$. Finally, as we will show in Proposition 2.9, constraint (A.9) is binding at optimality; thus, at optimality $\delta_0 \ge 0$ and we may suppress this constraint. □

Proposition 2.3. Let $(\tilde{y}, \tilde{\eta}) \in \mathcal{E}$ be an ℓ_q-projection of $(\hat{y}, \hat{\eta})$ onto $\mathcal{E}$. Then, any ℓ_p-deepest cut separating $(\hat{y}, \hat{\eta})$ from $\mathcal{E}$ supports $\mathcal{E}$ at $(\tilde{y}, \tilde{\eta})$.

Proof. Let $(\hat{\pi}, \hat{\pi}_0)$ be the solution associated with the ℓ_p-deepest cut. By Theorem 2.1 we have

$$\|(\tilde{y} - \hat{y}, \tilde{\eta} - \hat{\eta})\|_q = \frac{\hat{\pi}^\top(b - B\hat{y}) + \hat{\pi}_0(f^\top \hat{y} - \hat{\eta})}{\|(\hat{\pi}^\top B - \hat{\pi}_0 f^\top, \hat{\pi}_0)\|_p}. \tag{A.17}$$

On the other hand, $(\tilde{y}, \tilde{\eta}) \in \mathcal{E}$ implies $\hat{\pi}^\top(b - B\tilde{y}) + \hat{\pi}_0(f^\top \tilde{y} - \tilde{\eta}) \le 0$. To the contrary, assume that $(\tilde{y}, \tilde{\eta})$ is not on the hyperplane. Then, $\hat{\pi}^\top(b - B\tilde{y}) + \hat{\pi}_0(f^\top \tilde{y} - \tilde{\eta})$ must be negative, implying

$$0 < -\frac{\hat{\pi}^\top(b - B\tilde{y}) + \hat{\pi}_0(f^\top \tilde{y} - \tilde{\eta})}{\|(\hat{\pi}^\top B - \hat{\pi}_0 f^\top, \hat{\pi}_0)\|_p}. \tag{A.18}$$

Adding (A.17) and (A.18) we get

$$\|(\tilde{y}-\hat{y},\tilde{\eta}-\hat{\eta})\|_q < \frac{(\hat{\pi}^\top B - \hat{\pi}_0 f^\top)(\tilde{y}-\hat{y}) + \hat{\pi}_0(\tilde{\eta}-\hat{\eta})}{\|(\hat{\pi}^\top B - \hat{\pi}_0 f^\top, \hat{\pi}_0)\|_p}.$$

But this contradicts with Hölder's inequality since ℓ_p and ℓ_q are dual norms. □

Proposition 2.4. For sufficiently small $\hat{\eta}$, the ℓ_1-deepest cut separating $(\hat{y},\hat{\eta})$ from $\mathcal{E}$ is the flat cut $\eta \geq Q^*$, where $Q^* = \min_y Q(y)$ is the optimal value of Q for unrestricted y.

Proof. Since the dual norm of ℓ_1 is ℓ_∞, the objective function in Primal SSP (2.10) is to minimize the component with largest absolute value in $(y-\hat{y},\eta-\hat{\eta})$, which, for sufficiently small $\hat{\eta}$, is $|\eta-\hat{\eta}| = \eta-\hat{\eta}$. Thus, we can restate Primal SSP as the following LP

$$\begin{aligned} -\hat{\eta} + \min \quad & \eta \\ \text{s.t.} \quad & \eta \geq c^\top x + f^\top y \\ & Ax \geq b - By \\ & x \geq 0. \end{aligned} \tag{A.19}$$

Let $(\tilde{\eta},\tilde{y},\tilde{x})$ be the optimal solution of (A.19). Observe that $\tilde{\eta} = Q(\tilde{y}) = \min_y Q(y)$, that is $(\tilde{\eta},\tilde{y})$ is an optimal corner point of $\mathcal{E}$. Further, let π_0 and π be the dual multipliers. The dual LP reads as

$$\begin{aligned} -\hat{\eta} + \max \quad & \pi^\top b \\ \text{s.t.} \quad & \pi^\top A \leq \pi_0 c \\ & \pi^\top B = \pi_0 f \\ & \pi_0 = 1 \\ & \pi \geq 0. \end{aligned} \tag{A.20}$$

Let $(\hat{\pi}, \hat{\pi}_0)$ be the optimal solution to (A.20). The ℓ_1-deepest cut is $\hat{\pi}^\top(b - By) + \hat{\pi}_0(f^\top y - \eta) = \hat{\pi}^\top b - \eta \leq 0$. By strong duality, $\hat{\pi}^\top b = \tilde{\eta} = Q^*$, hence the deepest cut is the flat cut $\eta \geq Q^*$. □

Proposition 2.5. Let $Q^* = \min_y Q(y)$ be the optimal unrestricted value of Q. Provided that $\hat{\eta} < Q^*$ and $p > 1$, the ℓ_p-deepest cut that separates $(\hat{y}, \hat{\eta})$ is an optimality cut for any arbitrary $\hat{y}$ (i.e., even if $\hat{y} \notin \mathrm{dom}(Q)$).

Proof. Since $\hat{\eta} < Q^*$, we can separate $(\hat{y}, \hat{\eta})$ from $\mathcal{E}$ using the flat cut $\bar{\mathcal{H}} = \{(y, \eta) : \eta \geq Q^*\}$. Let $(\hat{\pi}, \hat{\pi}_0) \in \Pi$ be the dual solution associated with the deepest cut, and assume to the contrary that the deepest cut is vertical, that is $\hat{\pi}_0 = 0$.

Let $(\tilde{y}^H, \tilde{\eta}^H)$ and $(\tilde{y}^V, \tilde{\eta}^V)$ be the ℓ_q-projections of $(\hat{y}, \hat{\eta})$ onto $\bar{\mathcal{H}}$ and $\mathcal{H}(\hat{\pi}, \hat{\pi}_0)$, respectively. Observe that $(\tilde{y}^H, \tilde{\eta}^H) = (\hat{y}, Q^*)$ and $\tilde{\eta}^V = \hat{\eta}$, and that the ℓ_q-projection of $(\hat{y}, \hat{\eta})$ onto $\bar{\mathcal{H}} \cap \mathcal{H}(\hat{\pi}, \hat{\pi}_0)$ is $(\tilde{y}^V, \tilde{\eta}^H)$. Let $\bar{d}$ be the ℓ_q-distance of $(\hat{y}, \hat{\eta})$ from $\bar{\mathcal{H}} \cap \mathcal{H}(\hat{\pi}, \hat{\pi}_0)$. Note that

$$\begin{aligned}
\bar{d} =& \|(\hat{y}, \hat{\eta}) - (\tilde{y}^V, \tilde{\eta}^H)\|_q = \left(\|\hat{y} - \tilde{y}^V\|_q^q + \|\hat{\eta} - \tilde{\eta}^H\|_q^q\right)^{\frac{1}{q}} \\
=& \left(\|\hat{y} - \tilde{y}^V\|_q^q + \|\hat{\eta} - \tilde{\eta}^V\|_q^q + \|\hat{y} - \tilde{y}^H\|_q^q + \|\hat{\eta} - \tilde{\eta}^H\|_q^q\right)^{\frac{1}{q}} \\
=& \left(\|(\hat{y}, \hat{\eta}) - (\tilde{y}^V, \tilde{\eta}^V)\|_q^q + \|(\hat{y}, \hat{\eta}) - (\tilde{y}^H, \tilde{\eta}^H)\|_q^q\right)^{\frac{1}{q}} \\
=& \left((d^*)^q + (Q^* - \hat{\eta})^q\right)^{\frac{1}{q}},
\end{aligned}$$

where $d^* = \|(\hat{y}, \hat{\eta}) - (\tilde{y}^V, \tilde{\eta}^V)\|_q$. This implies that $\bar{d} > d^*$ since $q < \infty$ and $Q^* > \hat{\eta}$. However, both $\bar{\mathcal{H}}$ and $\mathcal{H}(\hat{\pi}, \hat{\pi}_0)$ support $\mathcal{E}$, therefore ℓ_q-distance of $(\hat{y}, \hat{\eta})$ from $\mathcal{E}$ (i.e., d^*) must be at least equal to $\bar{d}$, that is $d^* \geq \bar{d}$, which is a contradiction. □

Proposition 2.6. Let $(\tilde{y}, \tilde{\eta})$ be the unique ℓ_q-projection of $(\hat{y}, \hat{\eta})$ onto $\mathcal{E}$. If $\hat{\eta} < \tilde{\eta}$, then the ℓ_p-deepest cuts separating $(\hat{y}, \hat{\eta})$ are optimality cuts for any arbitrary $\hat{y}$ (i.e., even if $\hat{y} \notin \mathrm{dom}(Q)$).

Proof. Let $(\hat{\pi}, \hat{\pi}_0) \in \Pi$ be the dual solution associated with the deepest cut. Assume to the contrary that the deepest cut is a feasibility cut, that is $\hat{\pi}_0 = 0$. Since the projection is unique, the ℓ_q-projection of $(\hat{y}, \hat{\eta})$ onto the vertical cut $\mathcal{H}(\hat{\pi}, \hat{\pi}_0)$ must be $(\tilde{y}, \hat{\eta})$, which contradicts with the assumption that $\tilde{\eta} > \hat{\eta}$. □

Proposition 2.7. Epigraph distance function d^* certifies $d^*(\hat{y}, \hat{\eta}) > 0$ iff $(\hat{y}, \hat{\eta})$ is in the exterior of $\mathcal{E}$, $d^*(\hat{y}, \hat{\eta}) = 0$ iff $(\hat{y}, \hat{\eta})$ is on the boundary of $\mathcal{E}$, and $d^*(\hat{y}, \hat{\eta}) < 0$ iff $(\hat{y}, \hat{\eta})$ is in the interior of $\mathcal{E}$.

Proof. The proof follows from definition of d^* and noting that $\mathcal{E}$ is the intersection of half-spaces $\mathcal{H}(\pi, \pi_0)$ for all $(\pi, \pi_0) \in \Pi$. We can show each case one by one.

- $d^*(\hat{y}, \hat{\eta}) > 0$ iff there exists $(\hat{\pi}, \hat{\pi}_0) \in \Pi$ such that $d(\hat{y}, \hat{\eta}|\hat{\pi}, \hat{\pi}_0) > 0$. This implies that $d^*(\hat{y}, \hat{\eta}) > 0$ iff $(\hat{y}, \hat{\eta}) \notin \mathcal{H}(\hat{\pi}, \hat{\pi}_0)$ for some $(\hat{\pi}, \hat{\pi}_0) \in \Pi$, thus $(\hat{y}, \hat{\eta}) \notin \mathcal{E}$ (i.e., $(\hat{y}, \hat{\eta})$ is in the exterior of $\mathcal{E}$).
- $d^*(\hat{y}, \hat{\eta}) < 0$ iff $d(\hat{y}, \hat{\eta}|\hat{\pi}, \hat{\pi}_0) < 0$ for all $(\hat{\pi}, \hat{\pi}_0) \in \Pi$. This implies that $d^*(\hat{y}, \hat{\eta}) < 0$ iff $(\hat{y}, \hat{\eta})$ is in the interior of $\mathcal{H}(\hat{\pi}, \hat{\pi}_0)$ for all $(\hat{\pi}, \hat{\pi}_0) \in \Pi$, thus $(\hat{y}, \hat{\eta})$ is in the interior of $\mathcal{E}$.
- $d^*(\hat{y}, \hat{\eta}) = 0$ iff $d(\hat{y}, \hat{\eta}|\hat{\pi}, \hat{\pi}_0) \leq 0$ for all $(\hat{\pi}, \hat{\pi}_0) \in \Pi$, with at least one $(\hat{\pi}, \hat{\pi}_0) \in \Pi$ such that $d(\hat{y}, \hat{\eta}|\hat{\pi}, \hat{\pi}_0) = 0$. This implies that $d^*(\hat{y}, \hat{\eta}) = 0$ iff $(\hat{y}, \hat{\eta}) \in \mathcal{H}(\hat{\pi}, \hat{\pi}_0)$ for all $(\hat{\pi}, \hat{\pi}_0) \in \Pi$ and there exists $(\hat{\pi}, \hat{\pi}_0) \in \Pi$ such that $(\hat{y}, \hat{\eta})$ is on the boundary of $\mathcal{H}(\hat{\pi}, \hat{\pi}_0)$, thus $(\hat{y}, \hat{\eta})$ is on the boundary of $\mathcal{E}$.

□

Theorem 2.2. Benders distance functions are monotonic.

Proof. Let $\alpha_i \in [0,1]$ for $i = 1,2$ and assume that $\alpha_2 > \alpha_1$. Define $(\bar{y}^{(i)}, \bar{\eta}^{(i)}) = (1 - \alpha_i)(y^0, \eta^0) + \alpha_i(\hat{y}, \hat{\eta})$ for $i = 1,2$. Since $0 \leq \alpha_1 < \alpha_2 \leq 1$, we may state $(\bar{y}^{(1)}, \bar{\eta}^{(1)})$ as a convex combination of $(\bar{y}^{(2)}, \bar{\eta}^{(2)})$ and (y^0, η^0) of the following form

$$(\bar{y}^{(1)}, \bar{\eta}^{(1)}) = (1 - \frac{\alpha_1}{\alpha_2})(y^0, \eta^0) + \frac{\alpha_1}{\alpha_2}(\bar{y}^{(2)}, \bar{\eta}^{(2)}).$$

Convexity of d^* implies that

$$\begin{aligned} d^*(\alpha_1) = d^*(\bar{y}^{(1)}, \bar{\eta}^{(1)}) &\leq (1 - \frac{\alpha_1}{\alpha_2})d^*(y^0, \eta^0) + \frac{\alpha_1}{\alpha_2}d^*(\bar{y}^{(2)}, \bar{\eta}^{(2)}) = \frac{\alpha_1}{\alpha_2}d^*(\bar{y}^{(2)}, \bar{\eta}^{(2)}) \\ &\leq d^*(\bar{y}^{(2)}, \bar{\eta}^{(2)}) = d^*(\alpha_2), \end{aligned}$$

where the we have used $d^*(y^0, \eta^0) = 0$ because $(y^0, \eta^0) \in \partial\mathcal{E}$. Hence d is monotonic. □

Proposition 2.8. Distance function $d_{\ell p}$ is strongly monotonic for any $p \geq 1$.

Proof. The proof is exactly the same as Theorem 2.2, except

$$\begin{aligned} d^*_{\ell p}(\alpha_1) = d^*_{\ell p}(\bar{y}^{(1)}, \bar{\eta}^{(1)}) &\leq (1 - \frac{\alpha_1}{\alpha_2})d^*_{\ell p}(y^0, \eta^0) + \frac{\alpha_1}{\alpha_2}d^*_{\ell p}(\bar{y}^{(2)}, \bar{\eta}^{(2)}) = \frac{\alpha_1}{\alpha_2}d^*(\bar{y}^{(2)}, \bar{\eta}^{(2)}) \\ &< d^*_{\ell p}(\bar{y}^{(2)}, \bar{\eta}^{(2)}) = d^*_{\ell p}(\alpha_2), \end{aligned}$$

where the last strict inequality holds because $\frac{\alpha_1}{\alpha_2} < 1$ and $0 < d^*_{\ell p}(\bar{y}^{(2)}, \bar{\eta}^{(2)}) < \infty$ since $(\bar{y}^{(2)}, \bar{\eta}^{(2)}) \notin \mathcal{E}$ and $d^*_{\ell p}(\bar{y}^{(2)}, \bar{\eta}^{(2)})$ measures the ℓ_q distance of $(\bar{y}^{(2)}, \bar{\eta}^{(2)})$ to $\mathcal{E}$ and is thus finite. □

PROPOSITION 2.9. Let $d_g(\hat{y},\hat{\eta}|\pi,\pi_0) = \frac{\pi^\top(b-B\hat{y})+\pi_0(f^\top\hat{y}-\hat{\eta})}{g(\pi,\pi_0)}$ be a normalized distance function. Then, the separation problem (2.12) is equivalent to the normalized separation problem (2.13). That is

$$d_g^*(\hat{y},\hat{\eta}) = \max_{(\pi,\pi_0)\in\Pi_g} \pi^\top(b-B\hat{y}) + \pi_0(f^\top\hat{y}-\hat{\eta}).$$

Additionally, $g(\pi,\pi_0)\leq 1$ is binding at optimality.

Proof. The separation problem (2.12) can be equivalently expressed as

$$\max_{q>0}\left\{\max_{(\pi,\pi_0)\in\Pi:g(\pi,\pi_0)=q} \frac{\pi^\top(b-B\hat{y})+\pi_0(f^\top\hat{y}-\hat{\eta})}{q}\right\}. \tag{A.21}$$

Define $(\tilde{\pi},\tilde{\pi}_0)=\frac{1}{q}(\pi,\pi_0)$. Since Π is a cone it follows that $(\tilde{\pi},\tilde{\pi}_0)\in\Pi$. Additionally, since g is homogeneous, we have $g(\tilde{\pi},\tilde{\pi}_0)=\frac{1}{q}g(\pi,\pi_0)=1$. Therefore, the inner maximization in (A.21) can be restated as

$$\max_{(\tilde{\pi},\tilde{\pi}_0)\in\Pi:g(\tilde{\pi},\tilde{\pi}_0)=1} \tilde{\pi}^\top(b-B\hat{y})+\tilde{\pi}_0(f^\top\hat{y}-\hat{\eta}), \tag{A.22}$$

which is constant with respect to q. Therefore, (A.21) itself is equivalent to (A.22). We next show that (A.22) is equivalent to (2.13), that is $g(\pi,\pi_0)=1$ can be replaced with $g(\pi,\pi_0)\leq 1$. Let $(\hat{\pi},\hat{\pi}_0)\in\Pi_g$ be an arbitrary solution to (2.13) with $\hat{\alpha}=g(\hat{\pi},\hat{\pi}_0)<1$. Note that $(\bar{\pi},\bar{\pi}_0)=\frac{1}{\hat{\alpha}}(\hat{\pi},\hat{\pi}_0)\in\Pi_g$ with $g(\bar{\pi},\bar{\pi}_0)=1$. Additionally, we have

$$\bar{\pi}^\top(b-B\hat{y})+\bar{\pi}_0(f^\top\hat{y}-\hat{\eta}) = \frac{\hat{\pi}^\top(b-B\hat{y})+\hat{\pi}_0(f^\top\hat{y}-\hat{\eta})}{\hat{\alpha}} \geq \hat{\pi}^\top(b-B\hat{y})+\hat{\pi}_0(f^\top\hat{y}-\hat{\eta}),$$

which is strict if $\hat{\pi}^\top(b-B\hat{y})+\hat{\pi}_0(f^\top\hat{y}-\hat{\eta})>0$. Thus, at optimality $g(\pi,\pi_0)\leq 1$ is binding. □

Proposition 2.10. Distance function d_g with any positive homogeneous normalization function g is monotonic.

Proof. Since g is positive homogeneous, using Proposition 2.9, for any $(\bar{y}, \bar{\eta})$ we may state $d^*(\bar{y}, \bar{\eta})$ as

$$d_g^*(\bar{y}, \bar{\eta}) = \max_{(\pi, \pi_0) \in \Pi_g} \ \pi^\top (b - B\bar{y}) + \pi_0 (f^\top \bar{y} - \bar{\eta}),$$

where $\Pi_g = \{(\pi, \pi_0) \in \Pi : g(\pi, \pi_0) \leq 1\}$. This implies that d_g^* is convex, since it is the maximum of a number of linear functions. Therefore, by Theorem 2.2, d_g is monotonic. $\square$

Theorem 2.3. Let d_g be a Benders normalized distance function with g a convex piece-wise linear function. Then BD Algorithm 2 converges to an optimal solution or asserts infeasibility of MP in a finite number of iterations.

Proof. First, we show that the BD algorithm does not stagnate in a degenerate loop. Let $\hat{\Pi}_t$ be the set of dual solutions obtained before iteration t of the BD algorithm. Let $\text{MP}^{(t)}$ be the current approximation of MP with $(y^{(t)}, \eta^{(t)})$ its optimal solution and let $(\bar{\pi}, \bar{\pi}_0)$ be the dual solution obtained from BSP (2.12) for separating $(y^{(t)}, \eta^{(t)})$. If $\bar{\pi}^\top (b - By^{(t)}) + \bar{\pi}_0 (f^\top y^{(t)} - \eta^{(t)}) = 0$, then $(y^{(t)}, \eta^{(t)})$ is optimal for MP since $\eta^{(t)}$ is a lower bound on the optimal value of MP. Hence, assume that $\bar{\pi}^\top (b - By^{(t)}) + \bar{\pi}_0 (f^\top y^{(t)} - \eta^{(t)}) > 0$. Since $(y^{(t)}, \eta^{(t)})$ is feasible for $\text{MP}^{(t)}$, it follows that $\hat{\pi}^\top (b - By^{(t)}) + \hat{\pi}_0 (f^\top y^{(t)} - \eta^{(t)}) \leq 0$ for each $(\hat{\pi}, \hat{\pi}_0) \in \hat{\Pi}_t$; hence, $(\bar{\pi}, \bar{\pi}_0)$ cannot be a conical (i.e., scaling or a convex) combination of the solutions contained in $\hat{\Pi}_t$, meaning that, at each iteration, the BSP will produce a cut that is not implied by the cuts hitherto obtained.

Finally, since g is positive homogeneous and Π is a cone, by Proposition 2.9 we can restate the separation subproblem (2.12) as

$$\max_{(\pi,\pi_0)\in\Pi_g} \pi^\top(b - By^{(t)}) + \pi_0(f^\top y^{(t)} - \eta^{(t)}), \tag{A.23}$$

where $\Pi_g = \{(\pi,\pi_0) \in \Pi : g(\pi,\pi_0) \le 1\}$. Since g is a convex piece-wise linear function, Π_g is a polyhedron. Let Π_g^v and Π_g^r be the set of extreme points and rays of Π_g, respectively. Note that $\Pi_g^v \subset \Pi$ and $\Pi_g^r \subset \Pi$, and that they do not depend on $(y^{(t)}, \eta^{(t)})$. If (A.23) is bounded, then its optimal solution is attained at one of the points in Π_g^v, otherwise an extreme ray belonging to Π_g^r causes unsoundness. Either way, the produced extreme point/ray of Π_g serves as the certificate. Therefore, the number of iterations is bounded by $|\Pi_g^v| + |\Pi_g^r|$. □

Proposition 2.11. The following relationship between classical cuts, ℓ_p-deepest cuts and relaxed ℓ_1 pseudonorm cuts holds

$$d^*_{\mathrm{CB}}(\hat{y},\hat{\eta}) = Q(\hat{y}) - \hat{\eta} \ge d^*_{\ell\infty}(\hat{y},\hat{\eta}) \ge \cdots \ge d^*_{\ell p}(\hat{y},\hat{\eta}) \ge \cdots \ge d^*_{\ell 1}(\hat{y},\hat{\eta}) \ge d^*_{R\ell 1}(\hat{y},\hat{\eta}).$$

Proof. Recall that the $d_{\ell p}$, $d_{R\ell 1}$, and d_{CB} distance functions are defined as

$$\begin{aligned}
d_{\ell p}(\hat{y},\hat{\eta}|\pi,\pi_0) =& \frac{\pi^\top(b - B\hat{y}) + \pi_0(f^\top\hat{y} - \hat{\eta})}{\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_p}\\
d_{R\ell 1}(\hat{y},\hat{\eta}|\pi,\pi_0) =& \frac{\pi^\top(b - B\hat{y}) + \pi_0(f^\top\hat{y} - \hat{\eta})}{\sum_{i=1}^m \pi_i \sum_{j=1}^n |B_{ij}| + (1 + \sum_{j=1}^n |f_j|)\pi_0}\\
d_{\mathrm{CB}}(\hat{y},\hat{\eta}|\pi,\pi_0) =& \frac{\pi^\top(b - B\hat{y}) + \pi_0(f^\top\hat{y} - \hat{\eta})}{\pi_0}
\end{aligned}$$

The proof follows by noting the following facts:

(i) $\|\cdot\|_p \leq \|\cdot\|_{p'}$ for any $1 \leq p' < p$. This implies that

$$d_{\ell\infty}(\hat{y}, \hat{\eta}|\pi, \pi_0) \geq \cdots \geq d_{\ell p}(\hat{y}, \hat{\eta}|\pi, \pi_0) \geq \cdots \geq d_{\ell 1}(\hat{y}, \hat{\eta}|\pi, \pi_0) \qquad \forall (\pi, \pi_0) \in \Pi.$$

(ii) $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_p \geq \pi_0$ for any $p \geq 1$. This implies that

$$d_{\text{CB}}(\hat{y}, \hat{\eta}|\pi, \pi_0) \geq d_{\ell\infty}(\hat{y}, \hat{\eta}|\pi, \pi_0) \qquad \forall (\pi, \pi_0) \in \Pi.$$

(iii) $\|(\pi_0 f^\top - \pi^\top B, \pi_0)\|_1 \leq \sum_{i=1}^m \pi_i \sum_{j=1}^n |B_{ij}| + (1 + \sum_{j=1}^n |f_j|)\pi_0$. This implies that

$$d_{\ell 1}(\hat{y}, \hat{\eta}|\pi, \pi_0) \geq d_{R\ell 1}(\hat{y}, \hat{\eta}|\pi, \pi_0) \qquad \forall (\pi, \pi_0) \in \Pi.$$

Since the relationships follow for any $(\pi, \pi_0) \in \Pi$, they must hold when each distance function is at maximum. □

A.2 Implementation Details

We conducted our computational study on a Dell desktop equipped with Intel(R) Xeon(R) CPU E5-2680 v3 at 2.50GHz with 8 Cores and 32 GB of memory running a 64-bit Windows 10 operating system. We coded our algorithms in `C#` and solved the linear/quadratic problems using the `ILOG Concert` library and `CPLEX 12.10` solver. In the following, we provide general implementation details for BD, which we believe are of technical value beyond the application of this study.

A.2.1 Modern Implementation of BD Algorithm

While the sequential implementations of BD Algorithms 1 or 2 may appear intuitive, they come with several limitations. Of note, building a new branch-and-bound tree from scratch at each iteration incurs a large amount of overhead, especially when several iterations of the BD algorithm are needed. Moreover, at each iteration, only the optimal integer solution to the master problem is provided to the separation problems. This in turn may ignore several integer feasible solutions that are encountered during the branch-and-bound search, which are potentially optimal for the original problem but sub-optimal for the current iteration. This also ignores fractional solutions, which may prove useful for producing effective Benders cuts.

An alternative is to implement BD algorithms in the modern fashion, known as Branch-and-Benders-Cut (BBC), where Benders cuts are added to the cut pool of branch-and-cut on the fly (see e.g., Fortz and Poss 2009, for one of the early implementations of BBC). BBC allows for solving the integer master problem in a single run, thus potentially saving computation time by avoiding solving multiple integer master problems. Furthermore, this framework permits separating both integer and fractional master solutions. The former is implemented by treating BD cuts as lazy constraints, while the latter is implemented by treating BD cuts as valid inequalities for the master problem, which are invoked by `CPLEX` using the `LazyConstraint` callback when the current-node solution happens to be integer and `UserCut` callback at each branch decision node, respectively.

A.2.2 Scaling of η in the Cuts and Normalization Functions

A pitfall in implementing BD is that scales of master problem variables η and y are often unbalanced, meaning that the coefficient of η in (optimality) cuts is often too small or too large compared to the coefficients of the y variables. This imbalance poses two numerical issues:

(i) The cuts become numerically unstable and the solver may not handle them correctly, which in turn may result in having to cut off (almost) the same point over and over.

(ii) Note that, in the cut $\pi^\top b \leq (\pi^\top B - \pi_0 f^\top)y + \pi_0 \eta$, the coefficient of η is π_0, while the coefficient of y is $\pi^\top B - \pi_0 f^\top$. Therefore, an imbalanced cut implies an imbalanced normalization function $g(\pi, \pi_0) = \|\pi^\top B - \pi_0 f^\top, \pi_0\|_p$ in the ℓ_p-distance function (2.8), which may pose numerical issues in the separation problem (2.9).

To resolve this issue, we replace η in the master problem (2.1) with $\eta = \beta\gamma$ for some suitably-chosen scaling factor $\beta > 0$. Consequently, γ becomes the decision variable in place of η, and we minimize γ in the objective function of (2.1). Note that, for a given $(\pi, \pi_0) \in \Pi$, the cut becomes

$$\pi^\top b \leq (\pi^\top B - \pi_0 f^\top)y + \beta\pi_0\gamma.$$

Consequently, the ℓ_p-norm normalization function in (2.8) becomes

$$g(\pi, \pi_0) = \|\pi^\top B - \pi_0 f^\top, \beta\pi_0\|_p,$$

and the relaxed ℓ_1-normalization function is obtained accordingly. Note that if we replace $\eta = \beta\gamma$, the scaling must be reflected in the primal space as well, and the projection problems must be done in the space of y and γ. For instance, the projection problem (2.19) at iteration h of GPA becomes

$$(\tilde{y}^{(h)}, \tilde{\gamma}^{(h)}) \leftarrow \operatorname*{argmin}_{(y,\gamma)\in\mathcal{C}^{(h)}} \quad \|(y - \hat{y}, \gamma - \hat{\gamma})\|_q.$$

From the projection perspective, it is not difficult to see that if β is too large, y becomes the dominant component in the projection (particularly for small values of q, e.g., $q = 1$), causing deepest cuts to converge to classical Benders cuts. To choose a suitable value for

β, we first solve DSP (2.17) using a core point $\bar{y}$ to obtain the dual solution $(\hat{u}, 1)$ and the optimality cut $\eta + (\hat{u}^\top B - f^\top)y \geq \hat{u}^\top b$. We then set β as

$$\beta = \frac{1}{n}\|\hat{u}^\top B - f^\top\|_1,$$

which is the average absolute coefficient value of the y variables in the cut. Note that we choose β only once in the course of BD Algorithm 2, and use the same β for stabilizing all cuts.

A.2.3 Reoptimizing the Separation Subproblems

Another important aspect in implementing the BD algorithm is being able to reoptimize the separation problems and retrieving the cuts quickly when a solver is used for solving the separation subproblems. Note that only the objective function in the separation problem (2.13) changes from one iteration of the BD algorithm 2 to another. For linear separation subproblems, one can use the primal simplex algorithm by setting parameter `Cplex.Param.RootAlgorithm` to `Cplex.Algorithm.Primal` to leverage the reoptimization capabilities of this method.

Additionally, at iteration t of BD algorithm, rearranging the objective function in (2.13) as

$$\pi^\top b - \sum_{j=1}^{n}(\pi^\top B_{.j} - \pi_0 f_j)y_j^{(t)} - \pi_0\eta^{(t)},$$

note that one needs to update the coefficient of $\pi^\top B_{.j} - \pi_0 f_j$ (i.e., $y_j^{(t)}$) only when $y_j^{(t)} \neq y_j^{(t-1)}$. Therefore, we may additionally define n auxiliary variables $\tau_j = \pi^\top B_{.j} - \pi_0 f_j$ to avoid changing the coefficients of all dual variables in the separation subproblems. For instance, the separation problem (2.14) for producing ℓ_p-deepest cuts becomes

$$\max\left\{\pi^\top b - \tau\hat{y} - \pi_0\hat{\eta} : \|(\tau,\pi_0)\|_p \le 1,\ \tau = \pi^\top B - \pi_0 f^\top,\ (\pi,\pi_0)\in\Pi\right\}.$$

The τ variables also simplify the expression for the normalization constraint (e.g., in ℓ_p-norm or in CW). Moreover, after the subproblem is solved, one can save $\mathcal{O}(mn)$ arithmetic operations in computing the cut coefficients by easily retrieving the value of the τ variables from the solver without having to recalculate the coefficients based on the π variables.

Appendix B

Supplementary Materials for Chapter 3

B.1 Bipartite Transformation of the MWM Problem

Here, we show how MWM(km) (3.46)-(3.48) can be transformed into an equivalent MWM problem in a bipartite graph. Define bipartite graph $G^B_{km} = (V_1, V_2, B)$, where $V_1 = \{v^1_i\}_{i \in \bar{H}^t}$, $V_2 = \{v^2_i\}_{i \in \bar{H}^t}$, and $B = \{(v^1_i, v^2_j) \in V_1 \times V_2 : (i,j) \in \bar{A}^t_k\}$. Let μ^B be a maximum weighted matching in G^B_{km}, and let

$$\mu^{km}_{ij} = \frac{1}{2}\left(\mu^B_{(v^1_i, v^2_j)} + \mu^B_{(v^1_j, v^2_i)}\right) \qquad \forall (i,j) \in \bar{A}^t_k. \tag{B.1}$$

Proposition B.1. *$\mu^{km} = (\mu^{km}_{ij})_{(i,j)\in \bar{A}^t_k}$ defined in* (B.1) *is optimal to MWM(km)* (3.46)-(3.48).

Proof. μ^B being a matching in G^B_{km} requires

$$\sum_{j\in\bar{H}^t:(i,j)\in\bar{A}^t_k} \mu^B_{(v^1_i,v^2_j)} \leq 1 \qquad \forall v^1_i \in V_1 \text{ (or equivalently } i\in\bar{H}^t) \tag{B.2}$$

$$\sum_{j\in\bar{H}^t:(i,j)\in\bar{A}^t_k} \mu^B_{(v^1_j,v^2_i)} \leq 1 \qquad \forall v^2_i \in V_2 \text{ (or equivalently } i\in\bar{H}^t). \tag{B.3}$$

Summing (B.2) and (B.3) and dividing both sides of the resulting inequality by 2 yields

$$\sum_{j\in\bar{H}^t:(i,j)\in\bar{A}^t_k} \frac{1}{2}\left(\mu^B_{(v^1_i,v^2_j)} + \mu^B_{(v^1_j,v^2_i)}\right) = \sum_{j\in\bar{H}^t:(i,j)\in\bar{A}^t_k} \mu^{km}_{ij} \leq 1 \qquad \forall i\in\bar{H}^t,$$

which implies (3.47), thus feasibility of μ^{km} defined in (B.1) for MWM(km). Now we show that μ^{km} is also optimal for MWM(km). By contradiction assume that μ^{km} is not optimal for MWM(km), hence there must exist a solution $\hat{\mu}^{km}$ with a strictly higher total weight. We show that this contradicts with optimality of μ^B. Define $\hat{\mu}^B$ as

$$\hat{\mu}^B_{(v^1_i,v^2_j)} = \hat{\mu}^B_{(v^1_j,v^2_i)} = \hat{\mu}^{km}_{ij} \qquad \forall (i,j)\in\bar{A}^t_k.$$

$\hat{\mu}^B$ clearly satisfies (B.2) and (B.3), hence is a feasible matching for the bipartite graph G^B_{km}. Moreover, note that

$$\sum_{(i,j)\in\bar{A}^t_k}\left(\hat{\mu}^B_{(v^1_i,v^2_j)} + \hat{\mu}^B_{(v^1_j,v^2_i)}\right) = 2\sum_{(i,j)\in\bar{A}^t_k}\hat{\mu}^{km}_{ij} > 2\sum_{(i,j)\in\bar{A}^t_k}\mu^{km}_{ij} = \sum_{(i,j)\in\bar{A}^t_k}\left(\mu^B_{(v^1_i,v^2_j)} + \mu^B_{(v^1_j,v^2_i)}\right),$$

which implies that $\hat{\mu}^B$ is a better matching than μ^B, contradicting the optimality of μ^B. □

The optimal solution to MWC(km) (3.43)-(3.45) can be obtained in a similar manner. Let τ^B be a minimum weight cover in G^B_{km}, and let

$$\tau^m_{ik} = \frac{1}{2}\left(\tau^B_{v^1_i} + \tau^B_{v^2_i}\right) \qquad \forall i\in\bar{H}^t. \tag{B.4}$$

Proposition B.2. *$\tau_k^m = (\tau_{ik}^m)_{i \in \bar{H}^t}$ defined in* (B.4) *is optimal to MWC(km)* (3.43)-(3.45).

Proof. τ^B being a cover in G_{km}^B implies that

$$\tau_{v_i^1}^B + \tau_{v_j^2}^B \geq \beta_{ij}^{km} \qquad \forall (i,j) \in \bar{A}_k^t \tag{B.5}$$

$$\tau_{v_j^1}^B + \tau_{v_i^2}^B \geq \beta_{ji}^{km} \qquad \forall (j,i) \in \bar{A}_k^t. \tag{B.6}$$

Since β^{km} is symmetric, summing (B.5) and (B.6) and dividing the resulting inequality by 2 implies (3.44); hence, τ_k^m defined in (B.4) is feasible for MWC(km). Now, by contradiction assume that τ_k^m is not optimal, hence there exists a solution $\hat{\tau}_k^m$ with a strictly lower cover weight. Define $\hat{\tau}^B$ as

$$\hat{\tau}_{v_i^1}^B = \hat{\tau}_{v_i^2}^B = \hat{\tau}_{ik}^m \qquad \forall i \in \bar{H}^t. \tag{B.7}$$

$\hat{\tau}^B$ satisfies (B.5) and (B.6), and has a cover weight lower than τ^B, contradicting the optimality of τ^B. □

Propositions B.1 and B.2 indicate that to solve MWC(km) and MWM(km), it is enough to find a MWM in bipartite graph G_{km}^B.

B.2 Computing the b-variables in the Single Objective DSP-II

We show how proper values of $(b_i)_{i \in \bar{H}^t}$ can be found via a relaxation of DSP-II (3.38). Observe that by relaxing constraints (3.24), relaxed DSP-II can be decomposed into $|\bar{H}^t|$

independent smaller subproblems, one for each $i \in \bar{H}^t$:

$$\begin{aligned}
\min \quad & \Gamma_i b_i + \sum_{k \in K} \sum_{m \in M} u_{ik}^m && && \text{(B.8)} \\
\text{s.t.} \quad & u_{ik}^m + w_k^m b_i \geq \rho_{ii}^{km} && (k, m) \in K \times M && \text{(B.9)} \\
& u_{ik}^m, b_i \geq 0 && (k, m) \in K \times M && \text{(B.10)}
\end{aligned}$$

Similar to problem (3.49)-(3.51), this problem is the dual of the LP-relaxation of a knapsack problem with knapsack capacity Γ_i, and items $(k, m) \in K \times M$ with weight w_k^m and profit ρ_{ii}^{km} for each item, and can be solved in the same fashion. As explained in §3.2.5, b_i is the profit-to-weight ratio of the break item $(\bar{k}, \bar{m})$

$$b_i = \rho_{ii}^{\bar{k}\bar{m}} / w_{\bar{k}}^{\bar{m}}. \tag{B.11}$$

While problem (B.8)-(B.10) has a formulation similar to problem (3.49)-(3.51), they are inherently different: problem (3.49)-(3.51) is a restriction of DSP-II, whereas problem (B.8)-(B.10) is a relaxation of DSP-II, hence the solution obtained by (B.8)-(B.10) may not be feasible to DSP-II. Therefore, once the value of $(b_i)_{i \in \bar{H}^t}$ is calculated, to obtain a complete feasible solution we need to calculate the value of the u-variables using the MWM algorithm.

It is worth mentioning that the major computational effort required for solving this problem is due to calculating the ρ_{ii}^{km} values, while the time required for solving the knapsack problem itself is almost negligible even for the largest instances. On the other hand, note that we need to calculate ρ_{ii}^{km} as part of preparing DSP-II (3.38), independently of the value the b-variables. This means using this hybrid MWM-Knapsack method, we can find promising values for the b-variables for the MWM problems (hence stronger cuts) without incurring additional computational cost.

Appendix C

Supplementary Materials for Chapter 4

C.1 Details of the SAA Algorithm

In this appendix, we describe the SAA procedure for approximating the model with stochastic demand and known revenue. The model with stochastic demand and uncertain revenue is approximated in a similar manner.

1. Generate $\mathcal{M}$ independent samples each of size $|\mathcal{N}|$; i.e., $\mathcal{N}_j = \{n_j^1, \dots, n_j^{|\mathcal{N}|}\}$, for $j \in \mathcal{M}$ and solve the corresponding SAA problem (4.8)-(4.13) for each sample $\mathcal{N}_j$. Let $\mathcal{V}^{\mathcal{N}_j}$ and $\hat{y}^{\mathcal{N}_j}$, $j \in \mathcal{M}$, be the corresponding optimal objective value and an optimal solution, respectively.

2. Calculate the average of all optimal values from the SAA problems and their variance:

$$\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}} = \frac{1}{|\mathcal{M}|} \sum_{j \in \mathcal{M}} \mathcal{V}^{\mathcal{N}_j}, \qquad \sigma^2_{\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}}} = \frac{1}{(|\mathcal{M}|-1)|\mathcal{M}|} \sum_{j \in \mathcal{M}} (\mathcal{V}^{\mathcal{N}_j} - \bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}})^2$$

The expected value of $\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}}$ provides an upper statistical bound for the optimal value of the original problem, and $\sigma^2_{\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}}}$ is an estimate of the variance of this estimator.

3. Pick a feasible solution $\hat{y} \in Y$ for problem (4.1)-(4.6), for example, use one of the previously computed solutions $\hat{y}^{\mathcal{N}_j}$. Estimate the objective function value of the original problem by using this solution as follows:

$$\mathcal{V}_{\mathcal{N}'}(\hat{y}) = \frac{1}{|\mathcal{N}'|}\Big[\sum_{n\in\mathcal{N}'}\sum_{k\in K}\sum_{a\in A_k}\sum_{m\in M}(r_k^m - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn}\Big] - \sum_{i\in H} f_i\hat{y}_i$$

where $\mathcal{N}'$ is a sample generated independently of the samples used in the SAA problems. Note that since the first-stage variables are fixed, one can take much larger number of scenarios for $|\mathcal{N}'|$ than the sample size $|\mathcal{N}|$ used to solve the SAA problems. The estimator $\mathcal{V}_{\mathcal{N}'}(\hat{y})$ serves as a lower bound on the optimal objective function value. We can estimate the variance of $\mathcal{V}_{\mathcal{N}'}(\hat{y})$ as follows:

$$\sigma^2_{\mathcal{N}'}(\hat{y}) = \tfrac{1}{(|\mathcal{N}'|-1)|\mathcal{N}'|}\sum_{n\in\mathcal{N}'}\left([\sum_{k\in K}\sum_{a\in A_k}\sum_{m\in M}(r_k^m - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn}] - \sum_{i\in H} f_i\hat{y}_i - \mathcal{V}_{\mathcal{N}'}(\hat{y})\right)^2$$

4. Calculate the estimators for the optimality gap and its variance. Employing the estimators computed in steps 2 and 3, we get:

$$\text{gap}_{\mathcal{N},\mathcal{M},\mathcal{N}'}(\hat{y}) = \bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}} - \mathcal{V}_{\mathcal{N}'}(\hat{y}), \qquad \sigma^2_{\text{gap}} = \sigma^2_{\bar{\mathcal{V}}_{\mathcal{M}}^{\mathcal{N}}} + \sigma^2_{\mathcal{N}'}(\hat{y})$$

We can then use these estimators to construct a confidence interval for the optimality gap.

C.2 Proof of Theorem 4.1

Theorem 4.1. Let S_r be the set of revenue scenarios where at most γ_r commodities are subject to revenue uncertainty. Then, the min-max regret stochastic model (4.41), in which regrets are calculated with respect to a fixed reference point $\hat{Z}$, is equivalent to the robust-stochastic model with max-min profit criterion (4.26).

Proof. Let $Z^s(\xi) = \hat{Z}$ for each revenue scenario $s \in S_r$ and demand realization $\xi \in \Xi$. Then, the min-max regret stochastic model (4.41) reads as:

$$\min_{(x,y)\in\mho} \mathbb{E}_\xi\Big[\max_{s\in S_r}\Big\{\hat{Z} - (\sum_{k\in K}\sum_{m\in M}\sum_{a\in A_k}(r_k^{ms} - \hat{C}_{ak})w_k^m(\xi)x_{ak}^m(\xi) - \sum_{i\in H} f_i y_i)\Big\}\Big]. \tag{C.1}$$

We now prove that (4.26) is equivalent to (C.1). Since $\hat{Z}$ is constant, it can be taken out from the inner maximization, the expectation, and the minimization, respectively. Hence, (C.1) can be reformulated as:

$$\hat{Z} - \max_{(x,y)\in\mho} \mathbb{E}_\xi\Big[\min_{s\in S_r}\Big\{\sum_{k\in K}\sum_{m\in M}\sum_{a\in A_k}(r_k^{ms} - \hat{C}_{ak})w_k^m(\xi)x_{ak}^m(\xi) - \sum_{i\in H} f_i y_i\Big\}\Big]. \tag{C.2}$$

For a given $(x, y) \in \mho$ and for each $\xi \in \Xi$, the inner minimization in (C.2) calculates the worst possible profit associated with (x, y) over all revenue scenarios. Given that each revenue scenario $s \in S_r$ involves at most γ_r commodities with uncertain revenue, we can map each revenue scenario s to a subset of commodities $U_r(s)$ including the commodities with uncertain revenue. That is, $r_k^{ms} = \bar{r}_k^m - \hat{r}_k^m$ if $(k, m) \in U_r(s)$, and $r_k^{ms} = \bar{r}_k^m$, otherwise. Therefore, the inner minimization in (C.2) can be rewritten as:

$$\begin{aligned}
&\min_{s\in S_r}\Big\{\sum_{k\in K}\sum_{m\in M}\sum_{a\in A_k}(\bar{r}_k^m - \hat{C}_{ak})w_k^m(\xi)x_{ak}^m(\xi) - \sum_{i\in H} f_i y_i - \sum_{(k,m)\in U_r(s)}\sum_{a\in A_k}(\hat{r}_k^m - \hat{C}_{ak})w_k^m(\xi)x_{ak}^m(\xi)\Big\}\\
&= \sum_{k\in K}\sum_{m\in M}\sum_{a\in A_k}(\bar{r}_k^m - \hat{C}_{ak})w_k^m(\xi)x_{ak}^m(\xi) - \sum_{i\in H} f_i y_i\\
&\qquad - \max_{s\in S_r}\Big\{\sum_{(k,m)\in U_r(s)}\sum_{a\in A_k}(\hat{r}_k^m - \hat{C}_{ak})w_k^m(\xi)x_{ak}^m(\xi)\Big\}.
\end{aligned} \tag{C.3}$$

Note that the maximization in (C.3) is equivalent to $\nu_\xi(x)$ as defined in (4.27), therefore

(C.1) is equivalent to

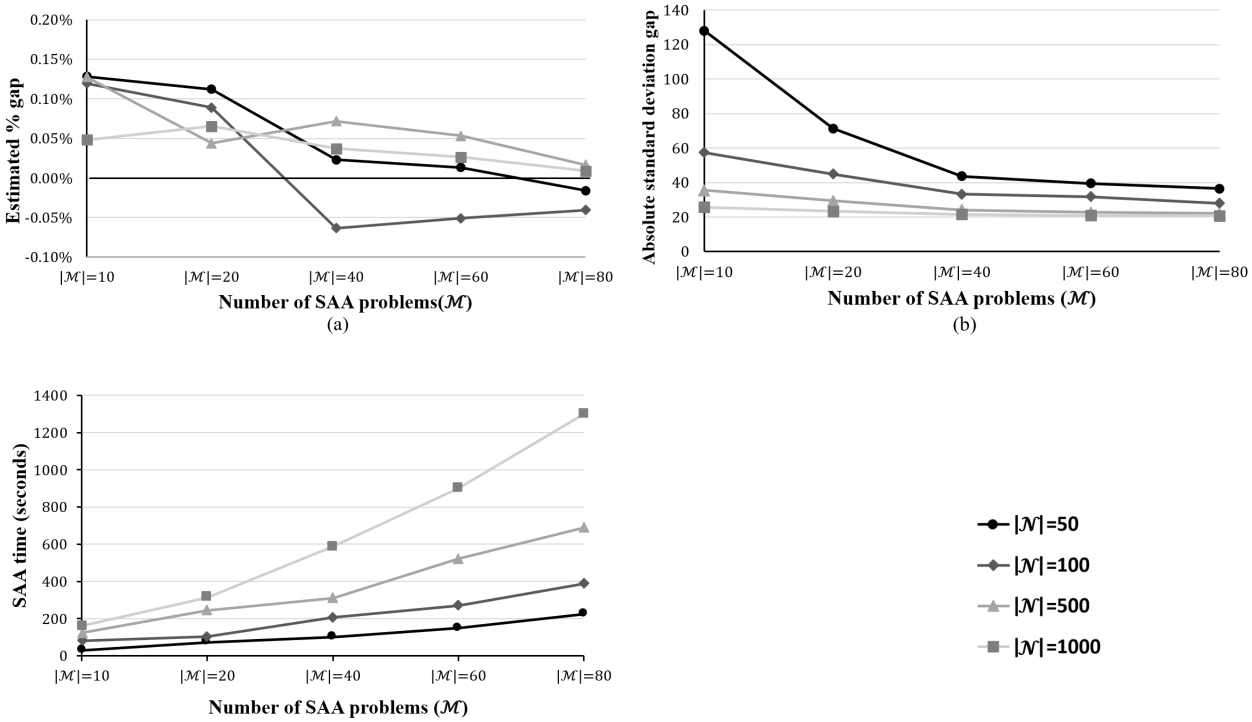
$$\hat{Z} - \max_{(x,y)\in\mho} \mathbb{E}_\xi \Big[\sum_{k\in K} \sum_{m\in M} \sum_{a\in A_k} (\bar{r}_k^m - \hat{C}_{ak}) w_k^m(\xi) x_{ak}^m(\xi) - \nu_\xi(x) \Big] - \sum_{i\in H} f_i y_i.$$

which is equivalent to the robust-stochastic model with max-min profit criterion (4.26). □

C.3 Practical Convergence of the SAA Algorithm

The aim of this section is to analyze the practical convergence of the SAA scheme in order to choose a sample size $|\mathcal{N}|$ and the number of replications $|\mathcal{M}|$ that provide the best trade-off between solution quality and computational time. To this end, we perform computational tests with sample sizes $|\mathcal{N}| \in \{50, 100, 500, 1000\}$ and a number of replications $|\mathcal{M}| \in \{10, 20, 40, 60, 80\}$. We select two 10-node instances of the AP dataset and generate independent samples as explained above, with ν set to 0.5. The sample size of $|\mathcal{N}'| = 10,000$ is used to evaluate the SAA gap.

Figures C.1 and C.2 plot the optimality gap, standard deviation for the optimality gap, and the computational time required for the SAA algorithm for different sample sizes $|\mathcal{N}|$ and $|\mathcal{M}|$, with AP10LL and AP10TL, respectively. Figures 1(a) and 2(a) clearly indicate that larger sample size result in smaller optimality gap on average. It is also observed that as the sample sizes $|\mathcal{N}|$ and $|\mathcal{M}|$ increase, the corresponding standard deviation for the optimality gap decreases (Figures 1b and 2b), whereas the corresponding computation time increases significantly (Figures 1c and 2c), for both AP10LL and AP10TL instances. In general, the largest sample size $|\mathcal{N}| = 1000$ provides the best average SAA gap with the least variation, and the sample size $|\mathcal{N}| = 50$ is the best in terms of the trade-off between solution quality and computational time. For this reason, we use sample sizes $|\mathcal{N}| = 50$ and $|\mathcal{M}| = 60$ during the rest of our computational experiments.



(a)

(b)

(c)

Figure C.1: Optimality gap, standard deviation for the optimality gap, and the total CPU time required for the SAA algorithm for different sample sizes $|\mathcal{N}|$ and $|\mathcal{M}|$ with AP10LL.

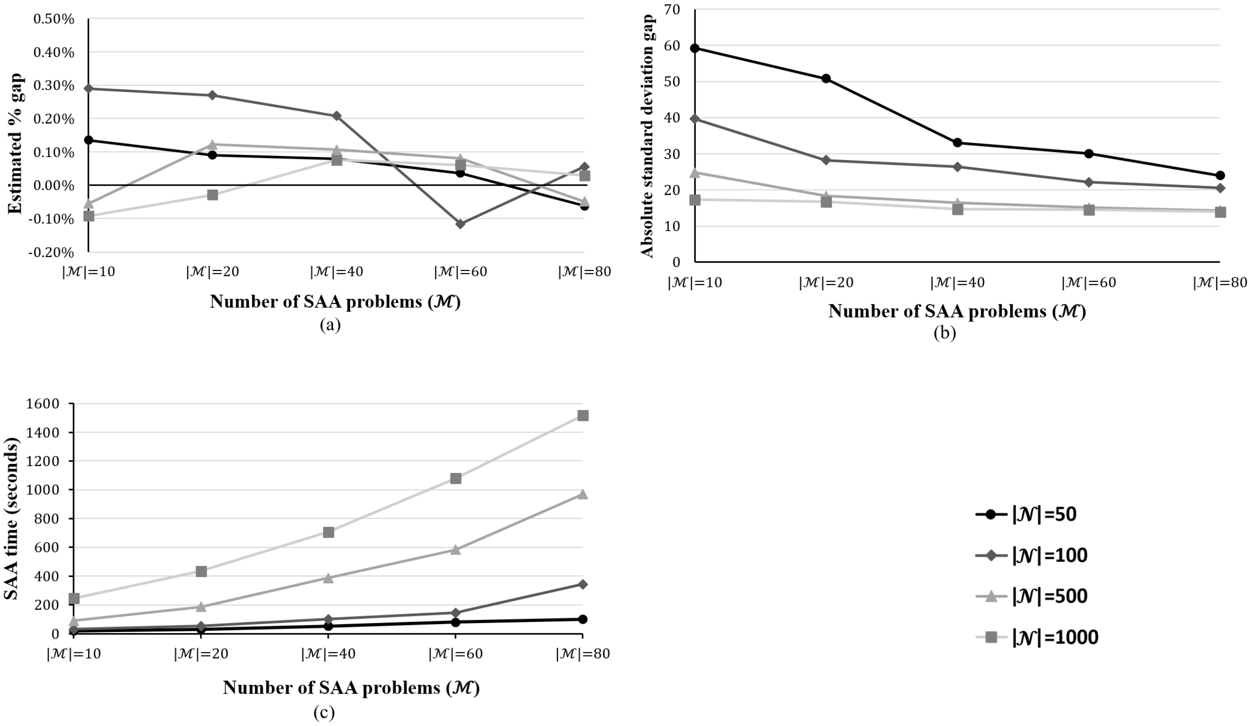


Figure C.2: Optimality gap, standard deviation for the optimality gap, and the total CPU time required for the SAA algorithm for different sample sizes $|\mathcal{N}|$ and $|\mathcal{M}|$ with AP10TL.

C.4 Supplementary Numerical Results

To evaluate the performance of the acceleration techniques proposed for SAA, we took runs with and without the implementation of the acceleration techniques on the 10–25 node instances from the AP dataset. We compare the computational times in Table C.1. The first two columns report the instance size and the coefficient of variation. The third and fourth columns labeled “Time (sec)” report the computation times in seconds without and with the implementation of the acceleration techniques, respectively. The last two rows report the averages.

The results provided in Table C.1 indicate that the algorithm performs more than two times faster on average with the implementation of the proposed acceleration techniques. In particular, for the larger-size instances, the improvement in CPU times goes up to five times.

C.5 Benders Decomposition for the Robust-Stochastic Model with Max-min Profit Criterion

For y set to a specific vector $y^t \in Y$ and for a given demand sample $\mathcal{N}$, the *primal subproblem* RSI-PSP($\mathcal{N}$) of the SAA counterpart of robust-stochastic model with max-min profit

$\lvert H \rvert$	ν	Time (sec)	
		Without acceleration	With acceleration
10LL	0.5	55.29	50.82
	1	47.38	41.76
10LT	0.5	18.42	15.39
	1	21.80	18.33
10TL	0.5	38.63	33.89
	1	53.46	53.48
10TT	0.5	17.34	14.97
	1	15.10	13.11
20LL	0.5	8,723.45	1,553.42
	1	6,601.72	2,713.17
20LT	0.5	2,892.03	1,442.36
	1	3,303.61	1,907.56
20TL	0.5	3,528.43	1,089.42
	1	4,202.39	1,173.79
20TT	0.5	249.41	101.11
	1	197.85	76.78
25LL	0.5	5,494.94	1,418.24
	1	6,653.34	2,014.32
25LT	0.5	15,617.96	3,218.51
	1	8,327.14	3,832.22
25TL	0.5	3,577.96	1,127.83
	1	4,382.44	1,613.32
25TT	0.5	5,053.09	1,420.91
	1	3,152.68	1,711.33
Average	0.5	3,772.25	957.24
	1	3,079.91	1,264.10

Table C.1: Computation times for the stochastic model with and without the implementation of the acceleration techniques for SAA.

criterion (4.32)-(4.34), denoted RS-I-PSP($\mathcal{N}$), can be formulated as:

$$\max \quad \frac{1}{|\mathcal{N}|}\Big[\sum_{n\in\mathcal{N}}\sum_{k\in K}\sum_{m\in M}\sum_{a\in A_k}(\bar{r}_k^m - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn} - (\gamma_r\mu^n + \sum_{n\in\mathcal{N}}\sum_{k\in K}\sum_{m\in M}\lambda_k^{mn})\Big] \tag{C.4}$$

$$\text{s.t.} \quad \hat{r}_k^m w_k^{mn}\sum_{a\in A_k} x_{ak}^{mn} - \mu^n - \lambda_k^{mn} \leq 0 \qquad k\in K, m\in M, n\in\mathcal{N} \tag{C.5}$$

$$\sum_{a\in A_k} x_{ak}^{mn} \leq 1 \qquad k\in K, m\in M, n\in\mathcal{N} \tag{C.6}$$

$$\sum_{a\in A_k:i\in a} x_{ak}^{mn} \leq y_i^t \qquad i\in H, k\in K, m\in M, n\in\mathcal{N} \tag{C.7}$$

$$\sum_{k\in K}\sum_{m\in M}\sum_{a\in A_k:i\in a} w_k^{mn}x_{ak}^{mn} \leq \Gamma_i y_i^t \qquad i\in H, n\in\mathcal{N} \tag{C.8}$$

$$x_{ak}^{mn}, \lambda_k^{mn}, \mu^n \geq 0 \qquad k\in K, m\in M, a\in A_k, n\in\mathcal{N}. \tag{C.9}$$

Observe that RS-I-PSP($\mathcal{N}$) can be decomposed into $|\mathcal{N}|$ independent subproblems, one for each $n \in \mathcal{N}$. Let β_k^{mn}, α_k^{mn}, u_{ik}^{mn}, and b_i^n be the dual variables associated with constraints (C.5)-(C.8), respectively. The *dual subproblem* associated with scenario $n \in \mathcal{N}$, denoted RS-I-DSP($\mathcal{N}, n$), can then be formulated as:

$$\min \quad \sum_{k\in K}\sum_{m\in M} \alpha_k^{mn} + \sum_{i\in H} y_i^t (\sum_{k\in K}\sum_{m\in M} u_{ik}^{mn} + \Gamma_i b_i^n) \tag{C.10}$$

$$\text{s.t.} \quad \hat{r}_k^m w_k^{mn} \beta_k^{mn} + \alpha_k^{mn} + u_{ik}^{mn} + u_{jk}^{mn} + w_k^{mn}(b_i^n + b_j^n) \geq (\bar{r}_k^m - \hat{C}_{ijk}) w_k^{mn}$$

$$k \in K, m \in M, (i,j) \in A_k : i \neq j \tag{C.11}$$

$$\hat{r}_k^m w_k^{mn} \beta_k^{mn} + \alpha_k^{mn} + u_{ik}^{mn} + w_k^{mn} b_i^n \geq (\bar{r}_k^m - \hat{C}_{iik}) w_k^{mn} \quad k \in K, m \in M, i \in H \tag{C.12}$$

$$\sum_{k\in K}\sum_{m\in M} \beta_k^{mn} \leq \gamma_r \tag{C.13}$$

$$\beta_k^{mn} \leq 1 \qquad k \in K, m \in M \tag{C.14}$$

$$\beta_k^{mn}, \alpha_k^{mn}, u_{ik}^{mn}, b_i^n \geq 0 \qquad k \in K, m \in M, i \in H. \tag{C.15}$$

Note that the dual subproblem is always feasible and bounded, hence, it attains its optimum at one of its extreme points. Let $P_{\mathcal{N}}^n$ be the polyhedron defined by (C.11)-(C.15) for $n \in \mathcal{N}$ and $Ex(P_{\mathcal{N}}^n)$ its set of extreme points. Since the subproblem can be decomposed by each scenario $n \in \mathcal{N}$, the Benders optimality cuts can be separated by each $n \in \mathcal{N}$. Hence, the Benders *master problem* RS-I-MP($\mathcal{N}$) can be reformulated as:

$$\max \frac{1}{|\mathcal{N}|} \sum_{n\in\mathcal{N}} \eta^n - \sum_{i\in H} f_i y_i \tag{C.16}$$

$$\text{s.t. } \eta^n \leq \sum_{k\in K}\sum_{m\in M} \alpha_k^{mn} + \sum_{i\in H} y_i(\Gamma_i b_i^n + \sum_{k\in K}\sum_{m\in M} u_{ik}^{mn}) \; n \in \mathcal{N}, (\beta^n, \alpha^n, u^n, b^n) \in Ex(P_{\mathcal{N}}^n) \tag{C.17}$$

$$y \in Y. \tag{C.18}$$

C.5.1 Solving the Benders Subproblem

The two-phase procedure described in §3.2.2 can be adapted to solve the subproblem for each demand scenario $n \in \mathcal{N}$. In Phase I, we remove the variables u_{ik}^{mn} and b_i^n associated with $i \in \bar{H}^t$ and calculate the values of the remaining variables. Note that when $i \in H^t$, constraints (C.6) and (C.8) imply constraints (C.7). Consequently, there exists an optimal solution where the dual values associated with constraints (C.7) (i.e. u_{ik}^{mn}) are equal to 0 for $i \in H^t$. Thus, the Phase I subproblem, denoted RS-I-DSP-I($\mathcal{N}, n$), can be formulated as:

$$\begin{aligned}
\min \quad & \sum_{k \in K} \sum_{m \in M} \alpha_k^{mn} + \sum_{i \in H^t} \Gamma_i b_i^n \\
\text{s.t.} \quad & \hat{r}_k^m w_k^{mn} \beta_k^{mn} + \alpha_k^{mn} + w_k^{mn}(b_i^n + b_j^n) \geq (\bar{r}_k^m - \hat{C}_{ijk}) w_k^{mn} && k \in K, m \in M, (i,j) \in A_k^t \\
& \hat{r}_k^m w_k^{mn} \beta_k^{mn} + \alpha_k^{mn} + w_k^{mn} b_i^n \geq (\bar{r}_k^m - \hat{C}_{iik}) w_k^{mn} && k \in K, m \in M, i \in H^t \\
& \sum_{k \in K} \sum_{m \in M} \beta_k^{mn} \leq \gamma_r \\
& \beta_k^{mn} \leq 1 && k \in K, m \in M \\
& \beta_k^{mn}, \alpha_k^{mn}, b_i^n \geq 0 && k \in K, m \in M, i \in H^t
\end{aligned}$$

where $A_k^t = \{(i,j) \in A_k \cap H^t \times H^t : i \neq j\}$. Once the optimal values of all the variables in the Phase I subproblem are obtained, the optimal value of the rest of variables will be computed in Phase II. Observe that if the value of the β_k^{mn} is given for commodity (k, m), then constraints (C.11) and (C.12) associated with this commodity can be rewritten as:

$$\begin{aligned}
\alpha_k^{mn} + u_{ik}^{mn} + u_{jk}^{mn} + w_k^{mn}(b_i^n + b_j^n) &\geq (\bar{r}_k^m - \hat{r}_k^m \beta_k^{mn} - \hat{C}_{ijk}) w_k^{mn} && (i,j) \in A_k : i \neq j \\
\alpha_k^{mn} + u_{ik}^{mn} + w_k^{mn} b_i^n &\geq (\bar{r}_k^m - \hat{r}_k^m \beta_k^{mn} - \hat{C}_{iik}) w_k^{mn} && i \in H.
\end{aligned}$$

Note that β_k^{mn} expresses the extent to which revenue of commodity (k, m) is subject to uncertainty. Consequently, obtaining the optimal value of the β-variables in Phase I implies resolving the data uncertainty in Phase I. Hence, in Phase II, we work with the realized revenue r_k^m (i.e. $r_k^m = \bar{r}_k^m - \hat{r}_k^m \beta_k^{mn}$) for each commodity (k, m). Thus, the Phase II feasibility problem is formulated as

$$u_{ik}^{mn} + u_{jk}^{mn} + w_k^{mn}(b_i^n + b_j^n) \geq \rho_{ijk}^{mn} \qquad k \in K, m \in M, (i,j) \in \bar{A}_k^t \tag{C.19}$$

$$u_{ik}^{mn} + w_k^{mn} b_i^n \geq \rho_{iik}^{mn} \qquad k \in K, m \in M, i \in \bar{H}^t \tag{C.20}$$

$$u_{ik}^{mn}, b_i^n \geq 0 \qquad k \in K, m \in M, i \in \bar{H}^t \tag{C.21}$$

where $\bar{A}_k^t = \{(i,j) \in A_k \cap \bar{H}^t \times \bar{H}^t : i \neq j\}$, and

$$\rho_{ijk}^{mn} = (r_k^m - \hat{C}_{ijk})w_k^{mn} - \alpha_k^{mn} \qquad (i,j) \in \bar{A}_k^t$$

$$\rho_{iik}^{mn} = \max\{\max_{j \in H_i^t}\{(r_k^m - \hat{C}_{ijk})w_k^{mn} - u_{jk}^{mn} - w_k^{mn} b_j^n\}, (r_k^m - \hat{C}_{iik})w_k^{mn}\} - \alpha_k^{mn} \qquad i \in \bar{H}^t,$$

in which $H_i^t = \{j \in H^t : (i,j) \in A_k \text{ or } (j,i) \in A_k\}$. To generate strong cuts, we formulate a multi-objective problem and adapt the lexicographic procedure described in §3.2.5 to efficiently solve this problem.

C.5.2 Acceleration Techniques

The acceleration techniques introduced in §4.1.3 can be used for the robust-stochastic problem with max-min profit criterion with slight modification. For completeness, we present the technique in this section. The acceleration technique for SAA is based on the observation that the cuts generated in solving sample $\hat{\mathcal{N}}$ can be transformed into valid cuts for sample $\mathcal{N}$. More specifically, in solving sample $\mathcal{N}$, we can retrieve feasible solutions for RS-I-DSP$(\mathcal{N}, n)$ for scenario n of sample $\mathcal{N}$ from the solutions contained in $P_{\hat{\mathcal{N}}}^{\hat{n}}$ for $\hat{n} \in \hat{\mathcal{N}}$.

Let $(\beta^{\hat{n}}, \alpha^{\hat{n}}, u^{\hat{n}}, b^{\hat{n}}) \in P_{\hat{\mathcal{N}}}^{\hat{n}}$ be a feasible solution for RS-I-DSP$(\hat{\mathcal{N}}, \hat{n})$, and $w_k^{m\hat{n}}$ be the demand for commodity $k \in K$ of class $m \in M$ under scenario $\hat{n} \in \hat{\mathcal{N}}$. It can easily be shown that

$(\beta^n, \alpha^n, u^n, b^n)$ defined by (C.22)-(C.25) is feasible for RS-I-DS($\mathcal{N}, n$):

$$\beta_k^{mn} = \beta_k^{m\hat{n}} \qquad k \in K, m \in M \tag{C.22}$$

$$\alpha_k^{mn} = \frac{w_k^{mn}}{w_k^{m\hat{n}}} \alpha_k^{m\hat{n}} \qquad k \in K, m \in M \tag{C.23}$$

$$u_{ik}^{mn} = \frac{w_k^{mn}}{w_k^{m\hat{n}}} u_{ik}^{m\hat{n}} \qquad k \in K, m \in M, i \in H \tag{C.24}$$

$$b_i^n = b_i^{\hat{n}} \qquad i \in H \tag{C.25}$$

Consequently, the solution obtained by (C.22)-(C.25) yields a valid cut for RS-I-MP($\mathcal{N}$). To avoid overloading the master problem with too many cuts, we restrict the algorithm to selecting one potentially best demand scenario $\hat{n}^* \in \hat{\mathcal{N}}$ as defined in (4.25).

C.6 Benders Decomposition for the Robust-Stochastic Model with Min-max Regret Criterion

In accordance with the robust-stochastic model with max-min profit criterion, we again assume that the hub location decisions are handled in the master problem and the rest is left to the subproblem. However, as we demonstrate in this section, solving the SAA counterpart of the min-max regret stochastic model (4.44)-(4.45) is more difficult than solving the robust-stochastic model with max-min profit criterion (4.32)-(4.34).

For a given demand scenario n of sample $\mathcal{N}$ and revenue scenario $s \in S_r$, let $\bar{Z}^{ns}$ be an estimation of the optimal value of (4.35)-(4.40). With y set to a specific vector $y^t \in Y$, the *primal subproblem* RS-II-PSP($\mathcal{N}$) reads as:

$$\max \quad \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}} \bar{V}^n \tag{C.26}$$

$$\text{s.t.} \quad \text{(C.6)–(C.8)}$$

$$\bar{V}^n - \sum_{n\in\mathcal{N}}\sum_{k\in K}\sum_{m\in M}\sum_{a\in A_k}(r_k^{ms} - \hat{C}_{ak})w_k^{mn}x_{ak}^{mn} \leq -\bar{Z}^{ns} \qquad n\in\mathcal{N}, s\in S_r \tag{C.27}$$

$$x_{ak}^{mn} \geq 0 \qquad k\in K, m\in M, a\in A_k, n\in\mathcal{N}. \tag{C.28}$$

Observe that RS-II-PSP($\mathcal{N}$) can be decomposed into $|\mathcal{N}|$ independent subproblems, one for each $n \in \mathcal{N}$. Let α_k^{mn}, u_{ik}^{mn}, b_i^n, and ω^{ns} be the dual variables associated with constraints (C.6)-(C.8) and (C.27), respectively. For a given demand scenario $n \in \mathcal{N}$, the *dual subproblem* RS-II-DSP($\mathcal{N}, n$) can then be stated as:

$$\min \quad \sum_{k\in K}\sum_{m\in M}\alpha_k^{mn} + \sum_{i\in H} y_i^t(\Gamma_i b_i^n + \sum_{k\in K}\sum_{m\in M}u_{ik}^{mn}) - \sum_{s\in S_r}\bar{Z}^{ns}\omega^{ns} \tag{C.29}$$

$$\text{s.t.} \quad \sum_{s\in S_r}\omega^{ns} = 1 \tag{C.30}$$

$$\alpha_k^{mn} + u_{ik}^{mn} + u_{jk}^{mn} + w_k^{mn}(b_i^n + b_j^n) \geq \sum_{s\in S_r}\omega^{ns}(r_k^{ms} - \hat{C}_{ijk})w_k^{mn}$$

$$k\in K, m\in M, (i,j)\in A_k : i\neq j \tag{C.31}$$

$$\alpha_k^{mn} + u_{ik}^{mn} + w_k^{mn}b_i^n \geq \sum_{s\in S_r}\omega^{ns}(r_k^{ms} - \hat{C}_{iik})w_k^{mn} \qquad k\in K, m\in M, i\in H \tag{C.32}$$

$$\alpha_k^{mn}, u_{ik}^{mn}, b_i^n, \omega^{ns} \geq 0 \qquad k\in K, m\in M, i\in H, s\in S_r. \tag{C.33}$$

Let $\bar{P}_{\mathcal{N}}^n$ denote polyhedron defined by the feasible region of RS-II-DSP($\mathcal{N}, n$) for $n \in \mathcal{N}$, and let $Ex(\bar{P}_{\mathcal{N}}^n)$ be the set of its extreme points. Each demand scenario $n \in \mathcal{N}$ can provide a Benders cut; hence, the Benders *master problem* RS-II-MP($\mathcal{N}$) can be reformulated as below:

$$\max \quad \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}}\eta^n - \sum_{i\in H} f_i y_i \tag{C.34}$$

$$\text{s.t.} \quad \eta^n \le \sum_{k\in K}\sum_{m\in M}\alpha_k^{mn} + \sum_{i\in H} y_i(\Gamma_i b_i^n + \sum_{k\in K}\sum_{m\in M} u_{ik}^{mn}) - \sum_{s\in S_r}\bar{Z}^{ns}\omega^{ns}$$

$$n\in\mathcal{N}, (\alpha^n, u^n, b^n, \omega^n) \in Ex(\bar{P}^n_{\mathcal{N}}) \tag{C.35}$$

$$y \in Y. \tag{C.36}$$

An overview of the BD algorithm for the min-max regret stochastic model is presented in Algorithm C.1.

Algorithm C.1 Benders Decomposition for the robust-stochastic model with min-max regret criterion

```
$UB \leftarrow +\infty, LB \leftarrow -\infty, t \leftarrow 1$
$\bar{P}^n_{\mathcal{N}} \leftarrow \emptyset \quad \forall n \in \mathcal{N}$
while $LB - UB < \zeta$ do
    SOLVE RS-II-MP($\mathcal{N}$) and obtain $y^t$ and $Z^t_{MP}$
    $UB \leftarrow Z^t_{MP}$
    for $n$ in $\mathcal{N}$ do
        SOLVE RS-II-DSP($\mathcal{N}, n$) with $y = y^t$ and obtain $(\alpha^n, u^n, b^n, \omega^n)^t$ and $Z^{tn}_{DS}$
        $\bar{P}^n_{\mathcal{N}} \leftarrow \bar{P}^n_{\mathcal{N}} \cup \{(\alpha^n, u^n, b^n, \omega^n)^t\}$
    end for
    $LB \leftarrow \max\{LB, \frac{1}{|\mathcal{N}|}\sum_{n\in\mathcal{N}} Z^{tn}_{DS} - \sum_{i\in H} f_i y^t_i\}$
    $t \leftarrow t + 1$
end while
```

C.6.1 Solving the Benders Subproblem

For a given demand scenario $n \in \mathcal{N}$, we solve RS-II-DSP($\mathcal{N}, n$) in two sequential phases based on the set of open/closed hubs. In Phase I, the optimal value of the α- and ω-variables, along with the value of u_{ik}^{mn} and b_i^n for $i \in H^t$ are calculated. Similar to the robust-stochastic model with max-min profit criterion, it can be shown that the optimal value of u_{ik}^{mn} for $i \in H^t$ is equal to 0. Hence, the Phase I subproblem, denoted RS-II-DSP-I

$(\mathcal{N}, n)$, can be formulated as

$$
\begin{aligned}
\min \quad & \sum_{k \in K} \sum_{m \in M} \alpha_k^{mn} + \sum_{i \in H^t} \Gamma_i b_i^n - \sum_{s \in S_r} \bar{Z}_s^n \omega^{ns} & \\
\text{s.t.} \quad & \sum_{s \in S_r} \omega^{ns} = 1 & \\
& \alpha_k^{mn} + w_k^{mn}(b_i^n + b_j^n) \geq \sum_{s \in S_r} \omega^{ns}(r_k^{ms} - \hat{C}_{ijk}) w_k^{mn} & k \in K, m \in M, (i,j) \in A_k^t \\
& \alpha_k^{mn} + w_k^{mn} b_i^n \geq \sum_{s \in S_r} \omega^{ns}(r_k^{ms} - \hat{C}_{iik}) w_k^{mn} & k \in K, m \in M, i \in H^t \\
& \alpha_k^{mn}, b_i^n, \omega^{ns} \geq 0 & k \in K, m \in M, i \in H^t, s \in S_r.
\end{aligned}
$$

Upon computing the optimal value of the Phase I variables, we obtain the optimal value of the rest of variables (i.e. u_{ik}^{mn} and b_i^n for $i \in \bar{H}^t$) in Phase II. As per the robust-stochastic version with max-min profit criterion, in Phase II, we work with the adjusted revenue (i.e. $r_k^m = \sum_{s \in S_r} \omega^s r_k^{ms}$) for each commodity (k, m). Therefore, for a given $n \in \mathcal{N}$, the Phase II subproblem can be formulated as the linear program (C.19)-(C.21) given for the Phase II subproblem of the robust-stochastic version with max-min profit criterion, which can be solved as a series of LP-relaxations of knapsack problems using the same sequential procedure.

C.6.2 Acceleration Techniques

As per the robust-stochastic model with max-min profit criterion, we can enhance the performance of the BD algorithm by employing variable fixing techniques and the SAA acceleration techniques proposed earlier. However, the main difficulty incurred by the min-max regret stochastic model is the need for calculating the optimal values of the deterministic counterparts formulated as (4.35)-(4.40) for each pair of demand and revenue scenario (n, s). More specifically, each replication of the SAA requires computing the $\bar{Z}^{ns}$ values $|\mathcal{N}| \times |S_r|$ times. Hence, cardinality of the sets $\mathcal{N}$ and S_r drastically affect the computational efficiency of solving the min-max regret stochastic problem. Therefore, solving the min-max regret stochastic problems demand more carefully devised acceleration techniques.

The SAA counterpart of the min-max regret stochastic model exhibits a more repetitive structure than that of the stochastic and max-min profit stochastic models, in that solving the min-max regret stochastic problem for each sample $\mathcal{N}$ requires obtaining the $\bar{Z}^{ns}$ values for each demand scenario $n \in \mathcal{N}$ and each revenue scenario $s \in S_r$. Although this additional step requires extra computational effort, if treated carefully, its repetitive structure can be efficiently exploited for speeding up the SAA algorithm.

For each demand scenario $n \in \mathcal{N}$ and each revenue scenario $s \in S_r$, $\bar{Z}^{ns}$ can be obtained by solving the deterministic formulation (4.35)-(4.40) using a BD algorithm as detailed in §3.2. For y set to a specific vector $y^t \in Y$, the *dual subproblem* DSP$(\mathcal{N}, n, s)$ is formulated as:

$$\min \sum_{k \in K} \sum_{m \in M} \alpha_k^m + \sum_{i \in H} y_i^t (\Gamma_i b_i + \sum_{k \in K} \sum_{m \in M} u_{ik}^m) \tag{C.37}$$

$$\text{s.t. } \alpha_k^m + u_{ik}^m + u_{jk}^m + w_k^{mn}(b_i + b_j) \geq (r_k^{ms} - \hat{C}_{ijk}) w_k^{mn} \quad k \in K, m \in M, (i,j) \in A_k : i \neq j \tag{C.38}$$

$$\alpha_k^m + u_{ik}^m + w_k^{mn} b_i \geq (r_k^{ms} - \hat{C}_{iik}) w_k^{mn} \quad k \in K, m \in M, i \in H \tag{C.39}$$

$$\alpha_k^m, u_{ik}^m, b_i \geq 0 \quad k \in K, m \in M, i \in H. \tag{C.40}$$

Our proposed acceleration technique for the min-max regret stochastic model is four-fold: (i) generating valid cuts for solving scenario pair (n, s) from the cuts generated for solving scenario pair $(\hat{n}, s)$, (ii) generating valid cuts for solving scenario pair (n, s) from the cuts generated for solving scenario pair $(n, \hat{s})$, (iii) approximating the $\bar{Z}^{ns}$ values, and (iv) generating valid cuts for solving the min-max regret stochastic problem from the cuts generated for obtaining the $\bar{Z}^{ns}$ values. We introduce the following proposition for step (i):

Proposition C.1. *Let $(\alpha^{\hat{n}s}, u^{\hat{n}s}, b^{\hat{n}s})$ be a feasible solution for DSP$(\hat{\mathcal{N}}, \hat{n}, s)$, and $w_k^{m\hat{n}}$ be the demand for commodity $k \in K$ of class $m \in M$ under scenario $\hat{n} \in \hat{\mathcal{N}}$. $(\alpha^{ns}, u^{ns}, b^{ns})$*

defined by (C.41)-(C.43) *is feasible for DSP*$(\mathcal{N}, n, s)$*:*

$$b_i^{ns} = b_i^{\hat{n}s} \qquad i \in H \tag{C.41}$$

$$\alpha_k^{mns} = \frac{w_k^{mn}}{w_k^{m\hat{n}}} \alpha_k^{m\hat{n}s} \qquad k \in K, m \in M \tag{C.42}$$

$$u_{ik}^{mns} = \frac{w_k^{mn}}{w_k^{m\hat{n}}} u_{ik}^{m\hat{n}s} \qquad k \in K, m \in M, i \in H \tag{C.43}$$

Proof. From (C.42) and (C.43), we obtain $\alpha_k^{m\hat{n}s} = \frac{w_k^{m\hat{n}}}{w_k^{mn}} \alpha_k^{mns}$ and $u_{ik}^{m\hat{n}s} = \frac{w_k^{m\hat{n}}}{w_k^{mn}} u_{ik}^{mns}$, respectively. Feasibility of $(\alpha^{ns}, u^{ns}, b^{ns})$ for DSP$(\mathcal{N}, n, s)$ can easily be verified by replacing $b_i^{\hat{n}s}$, $\alpha_k^{m\hat{n}s}$, and $u_{ik}^{m\hat{n}s}$ respectively with b_i, $\frac{w_k^{m\hat{n}}}{w_k^{mn}} \alpha_k^{mns}$, and $\frac{w_k^{m\hat{n}}}{w_k^{mn}} u_{ik}^{mns}$, in constraints (C.38)-(C.40) associated with DSP$(\hat{\mathcal{N}}, \hat{n}, s)$. □

Consequently, the feasible solution obtained for DSP$(\mathcal{N}, n, s)$ defined by (C.41)-(C.43) provides a valid cut for solving the demand scenario $n \in \mathcal{N}$ and revenue scenario $s \in S_r$. Note that the same proposition holds when $\hat{\mathcal{N}} = \mathcal{N}$.

Similarly, Proposition C.2 shows how step (ii) can be achieved by generating feasible solutions for DSP$(\mathcal{N}, n, s)$ from the feasible solutions of DSP$(\mathcal{N}, n, \hat{s})$.

Proposition C.2. *Let* $(\alpha^{n\hat{s}}, u^{n\hat{s}}, b^{n\hat{s}})$ *be a feasible solution for DSP*$(\mathcal{N}, n, \hat{s})$*, then* $(\alpha^{ns}, u^{ns}, b^{ns})$ *defined by* (C.44)-(C.46) *is feasible for DSP*$(\mathcal{N}, n, s)$*:*

$$b_i^{ns} = b_i^{n\hat{s}} \qquad i \in H \tag{C.44}$$

$$\alpha_k^{mns} = \max\{0, \alpha_k^{mn\hat{s}} + w_k^{mn}(r_k^{ms} - r_k^{m\hat{s}})\} \qquad k \in K, m \in M \tag{C.45}$$

$$u_{ik}^{mns} = u_{ik}^{mn\hat{s}} \qquad k \in K, m \in M, i \in H \tag{C.46}$$

Proof. Replacing $b_i^{n\hat{s}}$ and $u_{ik}^{mn\hat{s}}$ respectively with b_i^{ns} and u_{ik}^{mns} in constraints (C.38) and

(C.39) of DSP$(\mathcal{N}, n, \hat{s})$, and adding $w_k^{mn}(r_k^{ms} - r_k^{m\hat{s}})$ to both sides of these constraints yields

$$\alpha_k^{mn\hat{s}} + w_k^{mn}(r_k^{ms} - r_k^{m\hat{s}}) + u_{ik}^{mns} + u_{jk}^{mns} + w_k^{mn}(b_i^{ns} + b_j^{ns}) \geq (r_k^{ms} - \hat{C}_{ijk})w_k^{mn}$$
$$\alpha_k^{mn\hat{s}} + w_k^{mn}(r_k^{ms} - r_k^{m\hat{s}}) + u_{ik}^{mns} + w_k^{mn}b_i^{ns} \geq (r_k^{ms} - \hat{C}_{iik})w_k^{mn}.$$

Hence, any $\alpha_k^{mns} \geq 0$ satisfying $\alpha_k^{mns} \geq \alpha_k^{mn\hat{s}} + w_k^{mn}(r_k^{ms} - r_k^{m\hat{s}})$ provides a feasible solution to DSP$(\mathcal{N}, n, s)$. □

The valid cuts obtained by these propositions accelerate the BD algorithm for calculating the $\bar{Z}^{ns}$ values; however, computing the optimal values for all scenario pairs in all replications is computationally burdensome and also unnecessary. Note that sufficiently close demand scenarios are likely to result in the same optimal hub locations. Therefore, we only calculate the optimal $\bar{Z}^{ns}$ values for the first replication of the SAA algorithm, and approximate the $\bar{Z}^{ns}$ values (step iii) for the consequent replications as follows.

Let $\mathcal{N}_t$ denote the realized demand sample $\mathcal{N}$ at replication t of the SAA algorithm, for $t = 1, \ldots, \mathcal{M}$. At replication $t > 1$, for a given demand scenario $n \in \mathcal{N}_t$, let $\hat{n}$ be the closest demand scenario to n among the demand scenarios in the first replication, as selected via (4.25). We estimate $\bar{Z}^{ns}$ by fixing y at $\hat{y}^{\hat{n}s}$, where $\hat{y}^{\hat{n}s}$ is the optimal location of the hubs under the scenario pair $(\hat{n}, s)$ for $\hat{n} \in \mathcal{N}_1$ and $s \in S_r$.

Once the $\bar{Z}^{ns}$ values are obtained, we generate valid cuts for the min-max regret stochastic problem using the following proposition (step iv):

Proposition C.3. *Let $(\alpha^{n\hat{s}}, u^{n\hat{s}}, b^{n\hat{s}})$ be a feasible solution for DSP$(\mathcal{N}, n, \hat{s})$ for some particular $\hat{s} \in S_r$, then $(\alpha^n, u^n, b^n, \omega^n)$ is feasible for RS-II-DSP$(\mathcal{N}, n)$, where $(\alpha^n, u^n, b^n) = (\alpha^{n\hat{s}}, u^{n\hat{s}}, b^{n\hat{s}})$ and $\omega^{ns} = 1$ if $s = \hat{s}$, and $\omega^{ns} = 0$ if $s \neq \hat{s}$ for each $s \in S_r$.*

Proof. The proof can be easily verified by noting that the feasible region of DSP$(\mathcal{N}, n, \hat{s})$ is equivalent to the feasible region of RS-II-DSP($\mathcal{N}, n$) when $\omega^{n\hat{s}}$ is set to 1. □

Note that, if we eliminate a set of hubs through variable fixing techniques, the dual variables associated with those hubs will not be computed in the subproblem. Hence, for the dual solutions obtained by a scenario pair (n, s) to be usable for another scenario pair (or for the min-max regret stochastic problem), we cannot employ the variable fixing techniques. Therefore, we sacrifice the first demand scenario of the first replication and obtain the complete dual solutions for each revenue scenario without fixing any variables. These solutions are then used for the other scenario pairs (within current replication or subsequent replications) using Propositions C.1 and C.2 as well as for the min-max regret problems using Proposition C.3.

The proposed accelerated SAA algorithm is detailed in Algorithm C.2. We denote each demand scenario by an integer n and each revenue scenario by an integer s. Moreover, P_1^s denotes the set of dual solutions obtained in solving the first demand scenario of the first replication under the revenue scenario $s \in S_r$. At replication h, $\bar{P}_h^{ns}$ consists of feasible solutions for scenario pair $(n, s) \in \mathcal{N}_t \times S_r$ obtained by converting the solutions contained in P_1^s.

Algorithm C.2 Accelerated SAA for the min-max regret stochastic model

```
for $h$ in $1, \ldots, |\mathcal{M}|$ do
    if $h = 1$ then
        for $s \in S_r$ do
            $P_1^s \leftarrow \emptyset$
            for $n \in \mathcal{N}_1$ do
                $\bar{P}_1^{ns} \leftarrow \emptyset$
                if $n = 1$ then
                    if $s > 1$ then
                        Convert the solutions contained in $P_1^{\hat{s}}$ for each $\hat{s} < s$ to feasible solutions for DSP$(\mathcal{N}_1, n, s)$ using Proposition C.2 and add them to $P_1^s$.
                    end if
                    Calculate $\bar{Z}^{ns}$ using the initial cuts associated with $P_1^s$.
                    Store the obtained dual solutions in $P_1^s$.
                    $\bar{P}_1^{ns} \leftarrow P_1^s$
                else
                    Convert the solutions contained in $P_1^s$ to feasible solutions for DSP$(\mathcal{N}_1, n, s)$ using Proposition C.1 and store the obtained solutions in $\bar{P}_1^{ns}$.
                    Generate initial cuts for $(n, s)$ using the solutions contained in $\bar{P}_1^{ns}$.
                    Calculate $\bar{Z}^{ns}$ using the generated initial cuts.
                end if
            end for
        end for
    else
        for $n \in \mathcal{N}_t$ do
            Let $\hat{n}$ be the closest demand scenario in $\mathcal{N}_1$ to $n$ obtained via (4.25).
            for $s \in S_r$ do
                Approximate $\bar{Z}^{ns}$ using $\hat{y}^{\hat{n}s}$.
                Convert the solutions contained in $P_1^s$ to feasible solutions for DSP$(\mathcal{N}_h, n, s)$ using Proposition C.1 and store the obtained solutions in $\bar{P}_h^{ns}$.
            end for
        end for
    end if
    Obtain initial cuts for the min-max regret stochastic model associated with $\mathcal{N}_h$ using the solutions contained in $\bar{P}_h^{ns}$ using Proposition C.3.
    SOLVE the min-max regret stochastic model using Algorithm C.1.
end for
```