# Network Design under Uncertainty and Demand Elasticity

Carlos Zetina

A Thesis
in
the Department
of
Mechanical, Industrial, and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements
for the Degree of
Doctor of Philosophy (Industrial Engineering) at
Concordia University
Montréal, Québec, Canada

November 2018

**CONCORDIA UNIVERSITY**

**SCHOOL OF GRADUATE STUDIES**

This is to certify that the thesis prepared

By: Carlos Zetina

Entitled: Network Design under Uncertainty and Demand Elasticity

and submitted in partial fulfillment of the requirements for the degree of

Doctor Of Philosophy (Industrial Engineering)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

_____ Chair
Dr. Christopher W. Trueman

_____ External Examiner
Dr. Bernard Fortz

_____ External to Program
Dr. Bernard Gendron

_____ Examiner
Dr. Brigitte Jaumard

_____ Examiner
Dr. Masoumeh Kazemi Zanjani

_____ Thesis Co-Supervisor
Dr. Ivan Contreras

_____ Thesis Co-Supervisor
Dr. Jean-François Cordeau

Approved by _____
Dr. Ali Dolatabadi, Graduate Program Director

December 20, 2018 _____
Dr. Amir Asif, Dean
Gina Cody School of Engineering and Computer Science

# Abstract

## Network design under uncertainty and demand elasticity

**Carlos Zetina, Ph.D.**

**Concordia University, 2018**

Network design covers a large class of fundamental problems ubiquitous in the fields of transportation and communication. These problems are modelled mathematically using directed graphs and capture the trade-off between initial investment in infrastructure and operational costs. This thesis presents the use of mixed integer programming theory and algorithms to solve network design problems and their extensions. We focus on two types of network design problems, the first is a hub location problem in which the initial investments are in the form of fixed costs for installing infrastructure at nodes for them to be equipped for the transhipment of commodities. The second is a fixed-charge multicommodity network design problem in which investments are in the form of installing infrastructure on arcs so that they may be used to transport commodities.

We first present an extension of the hub location problem where both demand and transportation cost uncertainty are considered. We propose mixed integer linear programming formulations and a branch-and-cut algorithm to solve robust counterparts for this problem. Comparing the proposed models' solutions to those obtained from a commensurate stochastic counterpart, we note that their performance is similar in the risk-neutral setting while solutions from the robust counterparts are significantly superior in the risk-averse setting.

We next present exact algorithms based on Benders decomposition capable of solving large-scale instances of the classic uncapacitated fixed-charge multicommod-

ity network design problem. The method combines the use of matheuristics, general mixed integer valid inequalities, and a cut-and-solve enumeration scheme. Computational experiments show the proposed approaches to be up to three orders of magnitude faster than the state-of-the-art general purpose mixed integer programming solver.

Finally, we extend the classic fixed-charge multicommodity network design problem to a profit-oriented variant that accounts for demand elasticity, commodity selection, and service commitment. An arc-based and a path-based formulation are proposed. The former is a mixed integer non-convex problem solved with a general purpose global optimization solver while the latter is an integer linear formulation with exponentially many variables solved with a hybrid matheuristic. Further analysis shows the impact of considering demand elasticity to be significant in strategic network design.

# Acknowledgements

Beyond the goal of obtaining the degree of Doctor of Philosophy, for me the most important aspects of the last four plus years have been the love and support from the people around me. Without this, I think it can be very easy to lose oneself in this path we've chosen. The completion of this program and the contributions of this thesis have been influenced by many people in different ways.

First of all, I would like to thank my supervisors, Dr. Ivan Contreras and Dr. Jean-François Cordeau. Their patience, guidance, and support have been instrumental to this thesis and my personal growth as a scientist. Ivan's seemingly infinite inventory of ideas has stimulated my scientific curiosity and inculcated the patience to keep trying other innovative ideas until the problem finally yields. His selflessness with time and patience when explaining were critical especially during my first year in the program. I can't thank him enough for taking me under his wing.

Jean-François's approach to research has shaped my own methodology and writing style. His advice on the aspects of academic life and our roles as researchers and mind-molders is invaluable; not to mention his impressive time management skills which I hope to be able to replicate later on in my career. I am greatly indebted to him for his support and mentorship as they have helped make me become a more complete academician. Finally, I would like to thank both my supervisors for their unconditional support. Their comprehension and understanding of my many side-projects made these last few years the most enjoyable and productive of my life. Thank you!

I am greatly indebted to the person who has been by my side throughout this whole experience, my wife Ana. She is my unfailing supporter, motivational coach, my rock, and most importantly my tag-team partner on this journey of life. Without you, none of this would have been possible. Thank you for the sacrifices you made to be by my side; for listening to me when I rambled on about a new idea; but most of all thank you for the love and happiness you've brought to our lives. Thank you, cari! I look forward to repaying all you've done for us in the years ahead.

To my mom Clara Rosa, who taught me the values of hard work and discipline, I can't thank you enough. Your love and support mean the world to me. Without them, I would not be where I am today. To my sister Clarissa Isela, thank you for being a shoulder to lean on, a confidant I can laugh with, and an infinite source of courage. You're an example of true resilience, pushing through despite the hurdles life has thrown at you. You have and always will be an inspiration to me. To my aunt Francisca, thank you for your unconditional support and encouragement and for

spoiling me whenever I'm back home. You've been my supporter through thick and thin with a constant smile. Thank you for always making me feel special. To my father Carlos a.k.a. Wach, who despite not being with us is ever-present. Your life is an example to live by, showing kindness and compassion to everyone around you. Your premature passing is a constant reminder that tomorrow is not guaranteed, so love today. Thank you for these and many other valuable lessons that have molded my character. To the rest of my family, thank you for always making me feel so welcomed whenever I visit despite long periods of absence. It warms my heart to feel so loved although it makes the process of leaving again more difficult.

An enormous thank you to all the friends that became family here in Montréal. To Sabrina, Ana, and Juliana for always making us feel like family. Your love and warmth made the long winter months feel ever so cozy. We'll always look back fondly on our time at the "Red doors on Berri". To the Ortiz-Sierra, González-Sierra and Contreras-Talonia families with whom we've made such wonderful memories. Thank you for sharing these precious moments with us. We look forward to many more to come. To our fabulous, eclectic "Drama llama" group with whom we've shared countless dinners, parties, and outings. Wael, Jerry, Furkan, Sonia and Steve, thank you for making these last couple of years so intimate and fun. You guys are living proof that there's always time to hang out with friends. A special thanks to my good friend Camilo whose humor and inside jokes made the PhD experience both in the lab and out so much more enjoyable; despite me still not being able to detect his sarcasm at times.

I'd also like to thank my other labmates Andrés, Moayad, Ibrahim, Maryam Hemmati, Maryam Haghi, Pedram, and Masoud for the long conversations and experiences shared both in and out of the lab; Prof. Terekhov, Prof. Cherkesly, Hyame, Sara, Wael, Tan, Gabriel, Abeer, Hiba, Rafik, and Mohammad for sharing with me the growing pains of the Montréal Operations Research Student Chapter and TORCH; and all my other friends and colleagues who shared words of encouragement during my PhD program. Thank you!

I would also like to thank Concordia University, the Centre Interuniversitaire de Recherche sur les Réseaux d'entreprise, la Logistique et le Transport (CIRRELT), and my supervisors for their financial support during this program. Finally, to everyone whom I did not mention by name but that contributed toward this work, thank you.

*To my family*

*"We can propose many sophisticated algorithms and a theory but the final test of a theory is its capacity to solve the problems which originated it."*

*-George Dantzig*

# Contribution of Authors

This dissertation is presented under the *manuscript-based* format. It contains three articles that have been accepted for publication or are under revision in different journals. These were submitted in the following chronological order. The first article titled "Robust uncapacitated hub location" was published online in June 2017 in the journal *Transportation Research Part B: Methodological*. The second manuscript titled "Exact algorithms for multicommodity uncapacitated fixed-charge network design" was submitted to the journal *Computers & Operations Research* in October 2018. Finally, the third manuscript titled "Profit-oriented network design with demand elasticity" was submitted for publication to *Transportation Research Part B: Methodological* in October 2018.

All three manuscripts are co-authored with Dr. Ivan Contreras and Dr. Jean-François Cordeau who established research guidelines and reviewed the papers before submission. The first manuscript also includes Dr. Ehsan Nikbakhsh who initially worked on the project. The author of this thesis acted as the principal researcher with the corresponding duties such as the development of formulations and algorithms as well as the coding of solution methods and analysis of computational results along with writing the first drafts.

# Contents

# List of Figures

# List of Tables

# Chapter 1

# Introduction

Networks have played an important role in both the advancement of theory and application. Their use is ubiquitous in several fields ranging from engineering, chemistry, physics, production and manufacturing, telecommunications, and transportation. Their mathematical abstraction, known as graphs, permeated the fields of applied mathematics, computer science and operations research leading to significant theoretical progress. They played an important role in the development of computational complexity theory, representing many of the first twenty-one NP-complete problems presented by Richard Karp and leading to the development of elegant results such as the blossom algorithm by Jack Edmonds and the maximum flow/minimum cut theorem by Lester Ford, Ray Fulkerson and George Dantzig. The study of classic network problems has greatly accelerated our progress in modelling and understanding the world. Addressing more complex and realistic extensions is thus a promising avenue of research with the potential of leading to theoretical and application breakthroughs.

Network design problems incorporate the decisions of investing in infrastructure and exploiting it to render a service. They lie at the heart of strategic and tactical planning in areas such as personnel scheduling, logistics and service network design,

healthcare, and transportation. Network design problems are defined on a directed graph $G = (N, A)$ where $N$ is a set of nodes, $A$ is a set of arcs and $K$ is a set of commodities each defined by the tuple $(o_k, d_k, W_k)$ representing the origin, destination, and demand quantity of a commodity $k \in K$, respectively. This notation will be used throughout the entire thesis independent of the particular variant or extension being studied. We focus on two types of network design problems, the first is hub location in which the initial investments are in the form of fixed costs for installing infrastructure at nodes for them to be equipped for the transhipment of commodities. The second is fixed-charge multicommodity network design in which investments are in the form of installing infrastructure on arcs so that they may be used to transport commodities.

This thesis addresses three main aspects of network design: parameter uncertainty, demand elasticity, and computational efficiency. The first two are addressed by extending the assumption of known fixed parameter values (e.g. demand quantity and transportation costs) with uncertainty sets and gravity models. These are modelled using mixed integer programs and solved using algorithms tailored to the corresponding formulation. On the other hand, computational efficiency is addressed by solving large-scale instances of a fundamental network design problem using exact algorithms that combine decomposition methods, matheuristics, general mixed integer cuts and cut-and-solve/local branching enumeration schemes.

The contributions of this thesis can be categorized as follows:

- Problem modelling.
  - The development and solution of robust counterparts for hub location problems considering demand and transportation cost uncertainty both separately and, for the first time, simultaneously.
  - The creation of a new line of research dealing with the incorporation of

demand elasticity to strategic, long-term fixed-charge network design.

- Algorithmic development.
  - The adaptation of general mixed integer cuts to algorithms for large-scale optimization such as Benders decomposition.
  - The combination of a local branching enumeration scheme and Benders decomposition.
  - The presentation of a hybrid matheuristic that combines metaheuristics and concepts from column generation.
- Managerial insights
  - Evidence of the stable performance of solutions from robust counterparts across both risk-averse and risk-neutral settings.
  - Analysis of the impact of incorporating demand elasticity in strategic network design.

This thesis consists of four more chapters, three of which correspond to manuscripts submitted for publication in the peer-reviewed journals *Computers & Operations Research* and *Transportation Research Part B: Methodological*. Chapter 2 presents an extension of the hub location problem where both demand and transportation cost uncertainty are considered. These are modelled with mixed integer linear programming formulations and solved using a branch-and-cut algorithm. A comparison of the proposed models' solutions to those obtained from a commensurable stochastic variant is done to obtain insights on the nuances of both approaches. Chapter 3 proposes exact algorithms based on Benders decomposition capable of solving large-scale instances of the classic uncapacitated fixed-charge multicommodity network design problem. The methods combine the use of several mixed integer programming tools including matheuristics, general mixed integer cuts for a Benders reformulation, and

a cut-and-solve/local branching enumeration scheme. Computational experiments show the proposed algorithms to be up to three orders of magnitude faster than the state-of-the-art general purpose mixed integer programming solver.

Chapter 4 extends the classic fixed-charge multicommodity network design problem to a profit-oriented variant that accounts for demand elasticity, commodity selection, and service commitment. This line of research seeks to break the current paradigm of considering parameter uncertainty as a characteristic independent of the decisions taken within the optimization process. An arc-based and a path-based formulation are proposed. The former is a mixed integer non-convex problem solved with a general purpose global optimization solver while the latter is an integer linear formulation with exponentially many variables solved with a hybrid matheuristic. An analysis on the impact of considering demand elasticity shows it to be an important factor often ignored in strategic network design. Finally, Chapter 5 presents conclusions and potential impact of the research results presented in this thesis.

# Chapter 2

# Robust Uncapacitated Hub Location


## Abstract

In this paper, we present robust counterparts for uncapacitated hub location problems in which the level of conservatism can be controlled by means of a budget of uncertainty. We study three particular cases for which the parameters are subject to interval uncertainty: demand, transportation cost, and both simultaneously. We present mixed integer programming formulations for each of these cases and a branch-and-cut algorithm to solve the latter. We present computational results to evaluate the performance of the proposed formulations when solved with a general purpose solver and study the structure of the solutions to each of the robust counterparts. We also compare the performance between solutions obtained from a commensurable stochastic model and those from our robust counterparts in both risk neutral and worst-case settings.

## 2.1 Introduction

In many transportation, telecommunications and computer networks, direct routing of commodities between a large number of origin-destination (O/D) pairs is not possible due to economic and technological constraints. Instead, hub-and-spoke networks are commonly used to connect O/D pairs so as to efficiently route flows by using a small number of links. The key feature of these networks is the use of centralized units, known as hub facilities, for consolidation, sorting and transshipment purposes. *Hub location problems* (HLPs) consider the design of hub-and-spoke networks by locating a set of hub facilities, activating a set of inter-hub links, and routing a predetermined set of commodities through the network while optimizing a cost-based (or service-based) objective.

Applications of HLPs in the design of transportation and distribution systems are abundant. These include air freight and passenger travel, postal delivery, express package delivery, trucking, and rapid transit systems. Since the seminal work of O'Kelly [143], hub location has evolved into a rich research area. Early works focused mostly on a first generation of HLPs which are analogue to fundamental discrete facility location problems, while considering a set of assumptions (hubs fully interconnected, no direct connections, constant discount factor, all commodities must be routed, etc.) that allow to simplify network design decisions [see, 40, 46, for a discussion]. Recent works have studied more complex models that relax some of these assumptions and incorporate additional features of real-life applications. For instance, *hub arc location problems* (HALPs) [41] extend HLPs by relaxing full interconnection of hub nodes and incorporating hub arc selection decisions. *Hub network design problems with profits* [4] further extend HALPs by integrating within the decision-making process additional network design decisions on the nodes and commodities that have to be served. Other models consider: the design of multimodal networks [5, 164], competition and collaboration [129, 145], capacitated networks [51, 57], flow

dependent discounted costs [144, 169], and the design of particular network topologies [52, 134], among other things. We refer the reader to Campbell and O'Kelly [40], Zanjirani Farahani et al. [178], and Contreras [46] for recent surveys on hub location.

In most HLPs considered in the literature, the input parameters are assumed to be known and deterministic. In practice, however, this assumption is unrealistic. Long-term strategic decisions such as the location of hub facilities have to be made under high uncertainty on future conditions for relevant parameters (i.e., costs, demands and distances) that have a direct impact on the performance of hub networks. In some cases, some probabilistic information is known for these parameters and can be used to minimize the total expected cost by using stochastic programming techniques. However, in other cases, no information about their probability distributions is known except for the specification of intervals containing the uncertain values and thus, one must rely on robust optimization techniques to design hub networks which are robust in the sense that they can perform well even in the worst-case scenarios that may arise.

In this paper we show how discrete robust optimization techniques can be used in hub location to incorporate both independently and jointly demand and transportation costs uncertainties when the only available information is an interval of uncertainty. In particular, we study several robust counterparts for one of the most fundamental problems in hub location research, the *uncapacitated hub location problem with multiple assignments* (UHLP). In this problem, a predetermined set of commodities has to be routed via a set of hubs. It is assumed that open hubs are fully interconnected with more effective pathways, which allow a flow-independent discount factor to be applied to the transportation costs between hub nodes. The number of hubs to locate is not known in advance, but a setup cost for each hub facility is considered. It is also assumed that the incoming and outgoing flows at hubs as well as the flow routed through each link of the network are unbounded. Commodities

having the same origin but different destinations can be routed through different sets of hubs, i.e., a multiple assignment strategy is allowed. Demand between O/D pairs and transportation costs are assumed to be known and deterministic. The objective is to minimize the sum of the hub setup costs and of demand transportation costs over the solution network. To the best of our knowledge, the most efficient formulations for the UHLP are those of Hamacher et al. [100], Marín et al. [133], and Contreras and Fernández [47], whereas the best exact algorithm is the Benders decomposition of Contreras et al. [50].

The main contributions of this paper are the following. We introduce three different robust counterparts of the UHLP. The first is the *robust uncapacitated hub location problem with uncertain demands* (UHLP-D) in which demands between O/D pairs are considered to be uncertain values lying in a known interval. The second is the *robust uncapacitated hub location problem with uncertain transportation costs* (UHLP-TC) in which the transportation costs for all links of the network are uncertain values lying in a known interval and independent for each link. The third is the *robust uncapacitated hub location problem with uncertain demands and transportation costs* (UHLP-DTC) where uncertainty exists in both demands between O/D pairs and transportation costs for all links. This problem considers that the uncertainties of both classes of parameters are independent from each other. In these robust counterparts of the UHLP, the objective is to minimize the sum of the hub setup costs and of demand transportation costs in the worst-case scenario that may arise for the uncertain parameters. In the spirit of Bertsimas and Sim [25], we use a budget of uncertainty to allow decision-makers to control the desired level of conservatism in an independent way for both demand and transportation costs.

For each of the proposed robust models, we present mathematical programming formulations which are non-linear due to the min-max nature of the objective functions. For the case of UHLP-D and UHLP-TC, we use a dual transformation to

reformulate them as compact *mixed integer linear programs* (MIP) having a polynomial number of variables and constraints. However, for the case of UHLP-DTC this transformation cannot be used due to the interaction of demand and transportation costs parameters in the objective function. We show how the UHLP-DTC can be modeled as an MIP with a polynomial number of variables but an exponential number of constraints. As a result, we develop a simple branch-and-cut algorithm to handle this formulation. We perform extensive computational experiments on several sets of benchmark instances to assess the computational performance of the proposed MIP formulations when solved with a general purpose solver and our branch-and-cut algorithm. We study the effects of the intervals of uncertainty and of the budgets of uncertainty in the structure of optimal solution networks. In addition, we compare the performance between solution networks obtained from a deterministic model, a commensurable stochastic model and those from our robust counterparts in both risk neutral and worst-case settings.

The remainder of the paper is organized as follows. Section 2.2 provides a literature review on hub location problems dealing with uncertainty. In Section 2.3 we introduce the three considered robust counterparts of the UHLP. Section 2.4 describes the computational experiments we have run. The results produced by each model are presented and analyzed. Conclusions follow in Section 2.5.

## 2.2 Literature review

Discrete facility location problems under uncertainty have been widely studied in the literature. Louveaux [125], Snyder [167], and more recently Correia and Saldanha-da-Gama [56] provide comprehensive reviews on modeling approaches for stochastic and robust facility location. However, much less work has been done to study how different uncertainty aspects can be taken into account when designing hub networks.

One of the first is the paper by Marianov and Serra [130] for an application in the airline industry. The authors model hubs as M/D/c queuing systems to derive an analytic expression for the probability of a number of customers in the system. This expression is then represented in the model as a constraint that limits the number of airplanes in the system. Sim et al. [166] consider the $p$-hub center problem with travel times following a normal distribution. The authors use chance constraints to incorporate service level guarantees. Their model takes into account the uncertainty in travel times when designing the network so that the maximum travel time through the network is minimized.

Yang [176] proposes a two-stage stochastic model for air freight hub location under a finite set of possible demands. The locational decisions are treated in the first stage while the routing is the second stage. Data from the air freight market in Taiwan and China is used to test the proposed model. Contreras et al. [50] show how the UHLP can be stated as a two-stage integer stochastic program with recourse in the presence of uncertainty in demands and transportation costs. Three different variants are introduced. The first assumes demand between each O/D pair to be stochastic. The second considers that uncertainty in transportation costs is given by a single parameter equally influencing all links of the network. The third focuses on the more general case in which the uncertainty of transportation costs is independent for each link of the network. It is shown that the first two variants are equivalent to their associated expected value problem in which uncertain demand and transportation costs are replaced by their expectation. However, this situation does not hold for the third case. The authors present a sample average approximation method based on Monte Carlo simulation to obtain an estimate on the stochastic solution. Alumur et al. [6] study the uncapacitated hub location problem both with single and multiple assignments under uncertain setup costs for the hubs and demands. The first class of models deals with uncertainty in the setup costs assuming there is no known

probability distribution for their random parameters. The authors propose the use of a min-max regret model where the objective deals with the minimization of the worst-case regret over a finite set of scenarios. The second class focuses on uncertainty in demand and uses a two-stage stochastic program with recourse. However, as shown in Contreras et al. [50], these problems are actually equivalent to their associated expected value problem. The third class considers uncertainty in both setup costs and demands and are modeled as two-stage min-max regret programs with recourse.

Ghaffari-Nasab et al. [92] present robust capacitated hub location problems with both multiple and single assignments in which demand uncertainty is modeled with an interval of uncertainty. The authors consider the uncertainty in the capacity constraints and use a budget of uncertainty for each of them. However, they do not consider demand uncertainty in the objective function and use the nominal demand value instead. Habibzadeh Boukani et al. [99] study the same capacitated hub location problems with multiple and single assignments but now the uncertainties relate to the setup costs and capacities. The authors present min-max regret models involving five scenarios for each uncertain parameter. Shahabi and Unnikrishnan [165] present robust counterparts for uncapacitated hub location problems with both multiple and single assignments with demand uncertainty. Demand is assumed to be affinely dependent on a known mean and a number of independent random variables, i.e., ellipsoidal demand uncertainty. The authors propose mixed integer conic quadratic programming formulations and a linear relaxation strategy. Given the inherent difficulty for solving this class of mathematical programs, instances with up to 25 nodes are solved with a general purpose solver. Merakli and Yaman [136] study robust uncapacitated $p$-hub median problems with multiple assignments under polyhedral demand uncertainty. The authors use a hose uncertainty model and a hybrid model to characterize demand uncertainty. The former considers that the only available information is an upper limit on the total flow adjacent to each node, while the

latter incorporates lower and upper bounds on each O/D flow. The authors present MIP formulations and a Benders algorithm to solve these problems for instances with up to 150 nodes. We would like to highlight that, compared to our proposed models, none of these papers dealing with robust optimization in hub location considers uncertainty in transportation costs and its interaction with demand uncertainty.

Demand uncertainty has also been studied when designing hub networks from a congestion perspective. Elhedhli and Hu [72] study a hub location model that considers hub congestion costs, caused by delays at hub facilities, as an exponential function of the hub flow. Elhedhli and Wu [73] propose a different approach in which hubs behave as a queue with exponential service rate determined by its capacity. The congestion cost is modeled using a Kleinrock average delay function. de Camargo and Miranda [67] extend the previous models by considering two different perspectives: a network owner perspective in which the the goal is to design a network with the least congestion cost, and a user perspective where the goal is the minimization of the maximum congestion effect. Other works have considered reliability issues. An et al. [7] and Azizi et al. [11] present models in which disruptions at hub facilities are taken into account when designing hub networks. The proposed models mitigate the resulting hub unavailability by using backup hubs and alternative routes for demands. Finally, Sun and Zheng [168] study a probability model devised to identify promising hub locations for liner shipping networks.

We conclude this section with a brief overview of the most relevant methodological developments in the field of robust discrete optimization pertinent to our work. This field currently consists of two modelling approaches: *static* and *adaptive* robust optimization [see 26]. In the former, all decisions must be taken before the uncertainty is revealed while in the latter, first presented in Ben-Tal et al. [21], some decisions are taken after. Bertsimas and Goyal [24] and Goemans and Vondrák [95] show that the value of this adaptability is bounded under some assumptions. However, adaptive

approaches are expected to significantly outperform the static models in multiperiod problems.

Bertsimas and Sim [25] develop static robust models in which the level of conservatism is controlled by a budget of uncertainty. These models can be solved by either using duality theory to obtain a compact *robust counterpart* with polynomially many constraints and variables or a cutting plane method as in Bienstock and Özbay [30], where only a finite set of the infinitely many scenarios are considered at a time and more are considered as the algorithm progresses by solving the so-called adversarial problem. Fischetti and Monaci [77] compare the performance of cutting plane methods and robust counterpart reformulations. They note that cutting plane methods are preferable for linear programs while robust counterparts perform better for mixed integer problems due to additional preprocessing and cut generation tools available in commercial solvers. A more extensive and systematic comparison of these approaches can be found in Bertsimas et al. [28].

Adaptive robust approaches provide a less conservative framework by allowing some decisions to be made after uncertain parameters have been revealed. Bertsimas et al. [27] and Lorca et al. [123] apply this approach in unit commitment problems using two-stage and multistage models, respectively, whereas Gabrel et al. [88] apply it to location-transportation problems. Recourse functions of two-stage adaptive robust models are bilinear programs with linear constraints and are thus in general NP-hard. Cutting plane methods are usually employed to solve the complete robust models exactly. The recourse function is solved by either outer approximation or reformulated as an MIP. Zeng and Zhao [180] propose a framework in which columns and rows are generated simultaneously. Their computational experiments show significant improvements with respect to solution time and iterations required to converge. We refer to Gabrel et al. [88], Bertsimas et al. [26] and Gorissen et al. [96] for overviews on the state of the art in robust optimization.

## 2.3 Robust uncapacitated hub location

Before presenting the considered robust models, we formally define their deterministic counterpart, the UHLP. Let $G = (N, A)$ be a complete directed graph where $N$ is the set of nodes and $A$ the set of arcs. Without loss of generality, let $N$ represent the set of potential hub locations, and $K$ the set of commodities each with an origin, destination in $N$ and demand, denoted by the triplet $(o(k), d(k), W_k)$. For each node $i \in N$, $f_i$ is the fixed setup cost for locating a hub. For each $(i, j) \in A$ the distances, or transportation costs $d_{ij} \geq 0$ are assumed to be symmetric and satisfy the triangle inequality. The UHLP consists of locating a set of hub facilities and of determining the routing of commodities demand through the hubs, with the objective of minimizing the total setup and transportation cost.

In the case of UHLP, given that open hubs are fully interconnected at no cost, distances satisfy the triangle inequality, and direct connections between non-hub nodes are not allowed, every O/D path will contain at least one and at most two hubs, i.e., $P_{ak} = (o(k), a_1, a_2, d(k))$, where $a = (a_1, a_2) \in A$ is a *hub arc* and $a_1, a_2$ is the ordered pair of hubs to which $o(k)$ and $d(k)$ are assigned, respectively. The unit transportation cost for routing commodity $k$ along path $P_{ak}$ is given by $C_{ak} = \chi d_{o(k)a_1} + \alpha d_{a_1 a_2} + \delta d_{a_2 d(k)}$, where $\chi$, $\alpha$ and $\delta$ represent the flow-independent collection, transfer and distribution costs along the path. To reflect economies of scale between hub nodes, we assume that $\alpha < \chi$ and $\alpha < \delta$.

For each $i \in N$, we define binary location variables $z_i$ equal to 1 if and only if a hub is located at $i$. For each $k \in K$ and $a \in A$, we also introduce continuous routing variables $x_{ak}$ equal to the fraction of commodity demand $W_k$ routed via first hub $a_1$ and second hub $a_2$. The UHLP can be stated as follows [100]:

$$\text{minimize} \quad \sum_{i \in N} f_i z_i + \sum_{k \in K} \sum_{a \in A} W_k C_{ak} x_{ak} \tag{2.1}$$

$$\text{subject to} \quad \sum_{a \in A} x_{ak} = 1 \qquad \forall k \in K \tag{2.2}$$

$$\sum_{a \in A: i \in a} x_{ak} \leq z_i \qquad \forall k \in K, i \in N \tag{2.3}$$

$$z_i \in \{0, 1\} \qquad \forall i \in N \tag{2.4}$$

$$x_{ak} \geq 0 \qquad \forall a \in A, k \in K. \tag{2.5}$$

The first term of the objective is the total setup cost of the hubs and the second term is the total transportation cost. Constraints (2.2) ensure that each commodity demand is fully routed. Constraints (2.3) prohibit commodities from being routed via a non-hub node, whereas (2.4)-(2.5) are the standard integrality and non-negativity constraints. This formulation contains $|N| + |K||A|$ variables and $|K| + |N||K|$ constraints. Given that there are no capacity constraints on the hubs or links of the network, there always exists an optimal solution of (2.1)-(2.5) in which all $x_{ak}$ are integer. That is, each $k$ is fully routed on a single path. As we will later see in Section 2.4, this property does not necessarily hold for some of the studied robust counterparts.

We next present three robust variants of UHLP that incorporate uncertainty on demands $W_k$ and transportation costs $C_{ak}$. We note that an interesting characteristic of formulation (2.1)-(2.5) is that the uncertain parameters only appear in the objective function and not in the constraints defining the feasible region. As a consequence, when using (2.1)-(2.5) as a basis to model the robust counterparts the uncertain parameters will not affect the feasibility of the problem. This is not necessarily true when using other existing MIP formulations for the UHLP as the demand and transportation costs do appear in the constraint matrix [46].

In a similar fashion to the approach used in Bertsimas and Sim [25], we use a budget of uncertainty to allow decision-makers to control the desired level of conservatism. However, instead of defining a budget for the maximum number of variables for which the objective coefficient is allowed to differ from its nominal value, as in Bertsimas and Sim [25], we use two budgets of uncertainty to control the maximum number of commodity demands and transportation costs, respectively, whose value is allowed to differ from its nominal value. This is an important modeling feature due to the fact that objective coefficients of $x_{ak}$ variables are defined as functions of several uncertain parameters, i.e., $W_k C_{ak} = W_k(\chi d_{d(k)a_1} + \alpha d_{a_1 a_2} + \delta d_{a_2 d(k)})$. As we will show, this makes the proper modeling of the robust counterparts more involved, as compared to the approach of Bertsimas and Sim [25], given that each uncertain parameter appears in several objective coefficients associated with different routing variables $X_{ak}$.

### 2.3.1 Case A: Uncertain demands

We consider the UHLP-D in which demand is subject to interval uncertainty. For each commodity $k \in K$, let $W_k \in [W_k^L, W_k^L + W_k^\Delta]$ be the interval of uncertainty of $W_k$ where $W_k^L$ is its nominal value and $W_k^\Delta \geq 0$ its deviation. Let $h_W$ denote the uncertainty budget on the maximum number of demand parameters $W_k$ whose value is allowed to differ from its nominal value. The UHLP-D can be stated as:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in N} f_i z_i + \sum_{k \in K} \sum_{a \in A} W_k^L C_{ak} x_{ak} + \max_{S_W \subseteq K : |S_W| \leq h_W} \{ \sum_{k \in S_W} \sum_{a \in A} W_k^\Delta C_{ak} x_{ak} \} \\ \text{subject to} \quad & (x, z) \in \Theta, \end{aligned}$$

where $\Theta = \{(x, z) : (2.2) - (2.5) \text{ are satisfied}\}$. Given a solution $(x, z) \in \Theta$, the goal of the inner maximization of the objective function is to select a subset of commodities $S_W \subseteq K$ such that their demand perturbations $W_k^\Delta$ maximize the total cost. Refor-

mulating this inner problem as a mathematical program by introducing the binary variable $u_k \in \{0,1\}$ for each $k \in K$, we obtain

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in N} f_i z_i + \sum_{k\in K}\sum_{a\in A} W_k^L C_{ak} x_{ak} + \max\{\sum_{k\in K}\sum_{a\in A} W_k^\Delta C_{ak} x_{ak} u_k\} \\
\text{subject to} \quad & \sum_{k\in K} u_k \le h_W && (2.6) \\
& 0 \le u_k \le 1 \qquad \forall k \in K && (2.7) \\
& (x,z) \in \Theta.
\end{aligned}$$

Note that the integrality conditions of $u_k$ variables have been relaxed given that the constraint matrix of (2.6) is totally unimodular. Let $(\mu, \lambda)$ denote the vector of dual multipliers of constraints (2.6)-(2.7) of appropriate dimension. Dualizing the inner maximization problem we obtain the following MIP formulation for the UHLP-D:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i\in N} f_i z_i + \sum_{k\in K}\sum_{a\in A} W_k^L C_{ak} x_{ak} + \sum_{k\in K} \lambda_k + h_W \mu \\
\text{subject to} \quad & \lambda_k + \mu \ge \sum_{a\in A} C_{ak} W_k^\Delta x_{ak} \qquad \forall k \in K && (2.8) \\
& \lambda_k, \mu \ge 0 \qquad \forall k \in K \\
& (x,z) \in \Theta.
\end{aligned}$$

The above formulation contains $|K|+1$ additional continuous variables and $2|K|+1$ additional constraints compared to the deterministic UHLP.

### 2.3.2 Case B: Uncertain transportation costs

We now consider the UHLP-TC in which transportation costs are subject to interval uncertainty. For each arc $(i,j) \in A$, let $d_{ij} \in [d_{ij}^L, d_{ij}^L + d_{ij}^\Delta]$ be the interval of uncertainty of $d_{ij}$, where $d_{ij}^L$ is its nominal value and $d_{ij}^\Delta \geq 0$ its deviation. Observe that now each coefficient of the routing variables $x_{ak}$ involves up to three uncertain parameters, i.e., $C_{ak} = \chi d_{d(k)a_1} + \alpha d_{a_1 a_2} + \delta d_{a_2 d(k)}$. To simplify the exposition of the model, the transportation cost associated to each path $P_{ak}$ is stated as

$$C_{ak} = \sum_{(i,j)\in P_{ak}} \kappa_{ak}^{ij} d_{ij} = \sum_{(i,j)\in A} \kappa_{ak}^{ij} d_{ij},$$

where

$$\kappa_{ak}^{ij} = \begin{cases} \chi & \text{if } (i,j) = (o(k), a_1) \in P_{ak}, \\ \alpha & \text{if } (i,j) = (a_1, a_2) \in P_{ak}, \\ \delta & \text{if } (i,j) = (a_2, d(k)) \in P_{ak}, \\ 0 & \text{otherwise,} \end{cases}$$

for each $a \in A$, $k \in K$ and $(i,j) \in A$. In addition, for each $a \in A$ and $k \in K$, we define the nominal transportation cost as $C_{ak}^L = \sum_{(i,j)\in P_{ak}} \kappa_{ak}^{ij} d_{ij}^L$. Let $h_d$ denote the uncertainty budget of the maximum number of transportation cost coefficients whose values are allowed to differ from their nominal value. The UHLP-TC can be stated as:

$$\text{minimize } \sum_{i\in N} f_i z_i + \sum_{k\in K}\sum_{a\in A} W_k C_{ak}^L x_{ak} + \max_{S_d \subseteq A : |S_d| \leq h_d} \{ \sum_{k\in K}\sum_{a\in A} \sum_{(i,j)\in P_{ak}\cap S_d} d_{ij}^\Delta \kappa_{ak}^{ij} W_k x_{ak} \}$$

subject to $(x,z) \in \Theta$.

Given a solution $(x,z) \in \Theta$, the goal of the inner maximization is to select the subset of arcs $S_d \subseteq A$ whose transportation costs perturbations $d_{ij}^\Delta$ maximize the total cost. Reformulating this inner problem as a mathematical program by introducing

the binary variable $w_{ij} \in \{0,1\}$ for each $(i,j) \in A$, we obtain

$$\begin{aligned}
\text{minimize} \quad & \sum_{i \in N} f_i z_i + \sum_{k \in K} \sum_{a \in A} W_k C^L_{ak} x_{ak} + \max\{\sum_{k \in K} \sum_{a \in A} \sum_{(i,j) \in P_{ak} \cap S_d} d^{\Delta}_{ij} \kappa^{ij}_{ak} W_k x_{ak} w_{ij}\} \\
\text{subject to} \quad & \sum_{(i,j) \in A} w_{ij} \leq h_d & (2.9) \\
& 0 \leq w_{ij} \leq 1 \qquad \forall (i,j) \in A & (2.10) \\
& (x,z) \in \Theta.
\end{aligned}$$

Similarly to UHLP-D, the integrality conditions of $w_{ij}$ variables can be relaxed given that the constraint matrix of (2.9) is totally unimodular. Let $(\mu, \lambda)$ denote the vector of dual multipliers of constraints (2.9)-(2.10) of appropriate dimension. Dualizing the inner maximization problem we obtain the following MIP formulation for the UHLP-TC:

$$\begin{aligned}
\text{minimize} \quad & \sum_{i \in N} f_i z_i + \sum_{k \in K} \sum_{a \in A} W_k C^L_{ak} x_{ak} + \sum_{(i,j) \in A} \lambda_{ij} + h_d \mu & \\
\text{subject to} \quad & \lambda_{ij} + \mu \geq \sum_{k \in K} \sum_{a \in A : (i,j) \in P_{ak}} d^{\Delta}_{ij} \kappa^{ij}_{ak} W_k x_{ak} & \forall (i,j) \in A \quad (2.11) \\
& \lambda_{ij}, \mu \geq 0 & \forall (i,j) \in A \\
& (x,z) \in \Theta.
\end{aligned}$$

The above formulation contains $|A|+1$ additional continuous variables and $2|A|+1$ additional constraints compared to the deterministic UHLP.

### 2.3.3 Case C: Uncertain demand and transportation costs

We now focus on the more general variant UHLP-DTC in which both demand and transportation costs are subject to interval uncertainty. Similar to previous models, let $W_k \in [W^L_k, W^L_k + W^{\Delta}_k]$ and $d_{ij} \in [d^L_{ij}, d^L_{ij} + d^{\Delta}_{ij}]$ denote the interval of uncertainty

for commodity demands and transportation costs, respectively, and $h_d$ and $h_W$ their respective uncertainty budgets. To simplify the exposition of the proposed model, we define

$$F(z,x) = \sum_{i \in N} f_i z_i + \sum_{k \in K} \sum_{a \in A} W_k^L C_{ak}^L x_{ak},$$

as the nominal cost function, i.e., set-up cost and nominal routing cost, and

$$\begin{aligned} R(z,x,S_W,S_d) \;=\; & \sum_{k \in S_W} \sum_{a \in A} W_k^{\Delta} C_{ak}^L x_{ak} + \sum_{k \in K} \sum_{a \in A} \sum_{(i,j) \in P_{ak} \cap S_d} \kappa_{ak}^{ij} W_k^L d_{ij}^{\Delta} x_{ak} \\ & + \sum_{k \in S_W} \sum_{a \in A} \sum_{(i,j) \in P_{ak} \cap S_d} \kappa_{ak}^{ij} W_k^{\Delta} d_{ij}^{\Delta} x_{ak}, \end{aligned}$$

the uncertain routing cost function. Given that the coefficients of the routing variables $x_{ak}$ contain the multiplication of the demand $W_k$ and some transportation costs $d_{ij}$, three different scenarios may arise and those correspond to each term of $R(z,x,S_W,S_d)$. The first term represents an additional cost caused by an increase only in demands $W_k$, whereas the second term relates to an additional cost due to an increase only in transportation costs $d_{ij}$. The third term corresponds to the situation in which there is an additional cost caused by a simultaneous increase of both demand and transportation costs. The UHLP-DTC can be stated as:

$$\begin{aligned} \text{minimize} \quad & F(z,x) + \max_{S_d \subseteq A, |S_d| \leq h_d, S_W \subseteq K, |S_W| \leq h_W} R(z,x,S_W,S_d) \\ \text{subject to} \quad & (x,z) \in \Theta. \end{aligned}$$

For a given solution $(\bar{x},\bar{z}) \in \Theta$, the goal of the inner maximization of the objective function is to select a subset of commodities $S_W \subseteq K$ and a subset of arcs $S_d \subseteq A$, such that their perturbations maximize the total cost. This inner program, denoted

as $SP$, can be reformulated as the following mathematical program:

$$
\begin{aligned}
(SP) \quad \text{maximize} \quad & \sum_{k\in K}\sum_{a\in A} W_k^{\Delta} C_{ak}^{L} \bar{x}_{ak} u_k + \sum_{k\in K}\sum_{a\in A}\sum_{(i,j)\in A_{ak}} \kappa_{ak}^{ij} W_k^{L} d_{ij}^{\Delta} \bar{x}_{ak} w_{ij} \\
& + \sum_{k\in K}\sum_{a\in A}\sum_{(i,j)\in A_{ak}} \kappa_{ak}^{ij} W_k^{\Delta} d_{ij}^{\Delta} \bar{x}_{ak} u_k w_{ij} \\
\text{subject to} \quad & \sum_{k\in K} u_k \le h_k \\
& \sum_{(i,j)\in A} w_{ij} \le h_d \\
& u_k \in \{0,1\} \qquad \forall k \in K, \\
& w_{ij} \in \{0,1\} \qquad \forall (i,j) \in A.
\end{aligned}
$$

SP is a nonlinear integer program which can be linearized using standard techniques. However, it can also be stated as the following combinatorial optimization problem. Given a bipartite graph $B = (K \cup A, E)$ where each node in $K$ and $A$ has a profit $a_k$ and $b_{ij}$, respectively, and each edge in $E$ has a profit $c_{ijk}$, select node subsets $S_1 \subseteq K$ and $S_2 \subseteq A$ whose cardinality do not exceed $|S_1| \le h_W$ and $|S_2| \le h_d$, so as to maximize the total profit, given by the sum of the profits of the nodes in $S_1$ and $S_2$ and of the edges $e \in S_1 \times S_2 \subseteq E$. SP is thus a generalization of the *heaviest k-subgraph problem* [39].

**Proposition 1.** *The SP problem is $\mathcal{NP}$-hard.*

**Proof** The result follows by reduction from the heaviest $k$-subgraph problem, which is known to be $\mathcal{NP}$-complete even for the case of bipartite graphs [see 55]. ■

The above result implies that we cannot use a dual transformation as in UHLP-D and UHLP-TC to obtain a MIP formulation for UHLP-DTC. However, we can still

reformulate it as an MIP by introducing an additional continuous variable $y$ that keeps track of the maximum transportation cost associated with different sets $S_W$ and $S_d$. The UHLP-DTC can thus be formulated as the following MIP:

$$\begin{aligned}
\text{minimize} \quad & F(z,x) + y \\
\text{subject to} \quad & y \geq R(z,x,S_W,S_d) \quad \forall S_d \subseteq A, \forall S_W \subseteq K, |S_d| \leq h_d, |S_W| \leq h_W \quad (2.12) \\
& y \geq 0 \quad (2.13) \\
& (x,z) \in \Theta.
\end{aligned}$$

Observe that the above formulation contains a polynomial number of variables - it only contains one extra continuous variable with respect to the deterministic UHLP. However, it requires an exponential number of constraints (2.12) whose separation problem is equivalent to the solution of SP, which is $\mathcal{NP}$-hard. Therefore, we resort to a cutting-plane algorithm to handle this formulation with a general purpose solver. Details are given in the next sub-section.

### 2.3.4 A cutting plane algorithm for the UHLP-DTC

To solve the more challenging UHLP-DTC using the MIP formulation given in Section 3.3, we need to handle the huge number of *robust* constraints (2.12) in an efficient way. For this reason, we develop a branch-and-cut algorithm based on this MIP. The idea is to solve its LP relaxation with a cutting-plane algorithm by initially relaxing all of constraints (2.12) and then iteratively adding only those violated by the current LP solution. When no more violated inequalities (2.12) exist, within a threshold value of $10^{-4}$, or when the objective function has not improved by at least 0.5% in the last seven iterations, we resort to CPLEX for solving the resulting formulation by enumeration, using a call-back function for generating additional violated inequalities at some nodes of the enumeration tree.

In our implementation, we solve the following linearized version of SP with CPLEX to find the most violated inequality, if any.

$$
\begin{aligned}
(SP) \quad \text{maximize} \quad & \sum_{k\in K}\sum_{a\in A} W_k^{\Delta} C_{ak}^{L} \bar{x}_{ak} u_k + \sum_{k\in K}\sum_{a\in A}\sum_{(i,j)\in A_{ak}} \kappa_{ak}^{ij} W_k^{L} d_{ij}^{\Delta} \bar{x}_{ak} w_{ij} \\
& + \sum_{k\in K}\sum_{a\in A}\sum_{(i,j)\in A_{ak}} \kappa_{ak}^{ij} W_k^{\Delta} d_{ij}^{\Delta} \bar{x}_{ak} v_{ijk} \\
\text{subject to} \quad & \sum_{k\in K} u_k \le h_k \\
& \sum_{(i,j)\in A} w_{ij} \le h_d \\
& v_{ijk} \le u_k \qquad \forall (i,j) \in A, k \in K \qquad (2.14) \\
& v_{ijk} \le w_{ij} \qquad \forall (i,j) \in A, k \in K \qquad (2.15) \\
& v_{ijk} \in \{0,1\} \qquad \forall (i,j) \in A, k \in K \qquad (2.16) \\
& u_k \in \{0,1\} \qquad \forall k \in K, \\
& w_{ij} \in \{0,1\} \qquad \forall (i,j) \in A.
\end{aligned}
$$

The linearized model is obtained by substituting the nonlinear term $u_k w_{ij}$ with the binary variable $v_{ijk}$ and including constraints (2.14), (2.15) and (2.16). The usual linearization constraints $u_k + w_{ij} - v_{ijk} \le 1, \forall k \in K, (i,j) \in A$, are not necessary because of the non-negative coefficients of the $v_{ijk}$ variables and the maximization of the objective function.

Note that the parameters of $SP$ depend solely on the routes taken by commodities through the network. This means that even for a fixed set of open hubs there exists an exponential number of $SP$ problems defined by the possible routings of this hub configuration. To avoid solving $SP$ a substantial number of times, we do so at most once for fractional solutions $(\bar{z}, \bar{x}, \bar{y})$ obtained from nodes of the enumeration tree at a depth divisible by three. This rule avoids solving $SP$ an excessive number of times and prevents the underlying LPs from having a large number of dense constraints

(2.12).

The cutting plane algorithm used to solve UHLP-DTC is formally described in Algorithm 1 where the following notation is used: $\Xi$ is the pool of constraints (2.12) generated thus far, $\delta$ and *Depth* are the identifier and depth of the enumeration tree node that is currently being explored, respectively. At each node $\delta$, we define $\Psi(\delta)$ as the set of branching constraints and $\rho(\delta)$ as the number of constraints of type (2.12) added at $\delta$. Let UHLP-DTC$_{LP}(\delta)$ denote the MIP formulation of UHLP-DTC where constraints (2.12) are replaced by $\Xi$, integrality of the location variables $z$ is relaxed, and the constraints in $\Psi(\delta)$ are added. Finally, *tol* is a given cut tolerance and $\phi$ is a function whose value is 0 if the objective function value of UHLP-DTC$_{LP}(0)$ has not improved by at least 0.5% in the last seven iterations and 1 otherwise.

**Algorithm 1** Branch-and-cut algorithm for UHLP-DTC

**Require: 0**: Initialization
  $\delta = 0$, $\Xi = \emptyset$, $\Psi(\delta) = \emptyset$, $Depth = 0$, $\rho(\delta) = 0$, $\phi = 1$, $tol$=1E-4
**Require: 1**:
  Solve UHLP-DTC$_{LP}(\delta)$ and denote $(\bar{z}, \bar{x}, \bar{y})_\delta$ as the optimal solution
  Update $\phi$
  **if** [( $Depth = 0$ and $\phi = 1$ ) or ( $Depth$ mod 3=0 and $\rho(\delta) < 1$ ) ] **then**
    `Goto` **Step 2**
  **else**
    `Goto` **Step 3**
  **end if**
**Require: 2**:
  Solve $SP(\bar{z}, \bar{x}, \bar{y})_\delta$ and denote $(S_w^*(\bar{x}), S_d^*(\bar{x}))$ as the optimal solution
  **if** $[R(\bar{z}, \bar{x}, S_w^*(\bar{x}), S_d^*(\bar{x})) - \bar{y} > tol]$ **then**
    Add constraint $y \geq R(z, x, S_W, S_d)$ to the cut pool $\Xi$
    $\rho(\delta) \leftarrow \rho(\delta) + 1$
    `Goto` **Step 1**
  **else**
    `Goto` **Step 3**
  **end if**
**Require: 3** Continue exploring the enumeration tree
  **if** [optimality gap is $> \epsilon$ ] **then**
    Update $\delta$, $Depth$ and $\Psi(\delta)$ accordingly and `Goto` **Step 1**
  **else**
    Stop with an $\epsilon$-optimal solution
  **end if**

## 2.4 Computational Experiments

In this section we describe the extensive computational experiments we have run in order to analyze the performance of the three studied robust counterparts of the UHLP. This section is structured as follows. We first describe the computational environment and the sets of benchmark instances we have used in our experiments. In Section 2.4.1 we give insights on the impact of the level of uncertainty (worst estimation error of the nominal value) in optimal solutions of the robust counterparts, whereas in Section 2.4.2 we study the effect of the budgets of uncertainty $h_d$ and $h_W$ on the solution networks and objective values. Section 2.4.3 provides numerical results to assess the usefulness of the proposed MIP formulations when solved with a general purpose solver and to compare the computational difficulty of the three robust counterparts. Finally, in Section 2.4.4 we compare the performance of optimal solution networks of the deterministic model, the robust counterparts and a commensurable stochastic model in both worst-case and risk-neutral settings. This provides some indication of the virtues and limitations of each approach when dealing with demand and transportation costs uncertainty.

All experiments were run on HP servers managed by Calcul Québec and Compute Canada with an Intel Xeon X5650 Westmere processor at 2.67 GHz and 24 GB of RAM under Linux environment. All MIP formulations were coded in C and solved using the callback library of CPLEX 12.6.0. We use a traditional (deterministic) branch-and-bound solution algorithm with all CPLEX parameters set to their default values with the exception that the number of threads was set to only one.

We have used three different sets of benchmark instances found in the literature to construct a testbed for our experiments. The first is the well-known CAB data set of the US Civil Aeronautics Board. These instances were obtained from the website *(http://www.researchgate.net/publication/269396247_cab100_mok)*. The data in the CAB set refers to 100 cities in the US. It provides Euclidean distances between cities,

$d_{ij}$, and the values of the service demand $W_k$ between each pair of cities, where $o(k) \neq d(k)$. Since CAB instances do not provide the setup costs for opening facilities, we use the setup costs $f_i$ generated by de Camargo et al. [68]. We selected the instances with $|N| = 25$. The second set is the classical AP (Australian Post) set of instances, which is the most commonly used in the hub location literature. They were obtained from the OR library *(http://people.brunel.ac.uk/∼mastjjb/jeb/orlib/phubinfo.html)*. Transportation costs are proportional to the Euclidian distances between 200 postal districts in the metro Sydney area and the values $W_k$ represent postal flows between pairs of districts. We selected instances with $|N| = 10$, 20, 25, 40, and 50. Finally, the third set is the *Set I* instances introduced in Contreras et al. [50] for the stochastic counterparts of the UHLP. We used all instances within this set with $|N| = 10$, 20, 30, 40, and 50. For each of the considered instances in the entire testbed, the nominal values $W_k^L$ and $d_{ij}^L$ were set to the associated deterministic values provided in each dataset, and the collection and distribution factors $\chi = \delta = 1$. To control the size of the generated intervals of uncertainty for $W_K$ and $d_{ij}$, we define an additional parameter $\Omega$ as the maximum possible deviation of the value of each parameter. For each $k \in K$, we independently set $W_k^\Delta$ by considering $W_k^\Delta \sim U\left[0, \Omega W_k^L\right]$, and for each $(i, j) \in A$ we independently set $d_{ij}^\Delta$ by considering $d_{ij}^\Delta \sim U\left[0, \Omega d_{ij}^L\right]$.

### 2.4.1 The Impact of Uncertainty on Optimal Solutions

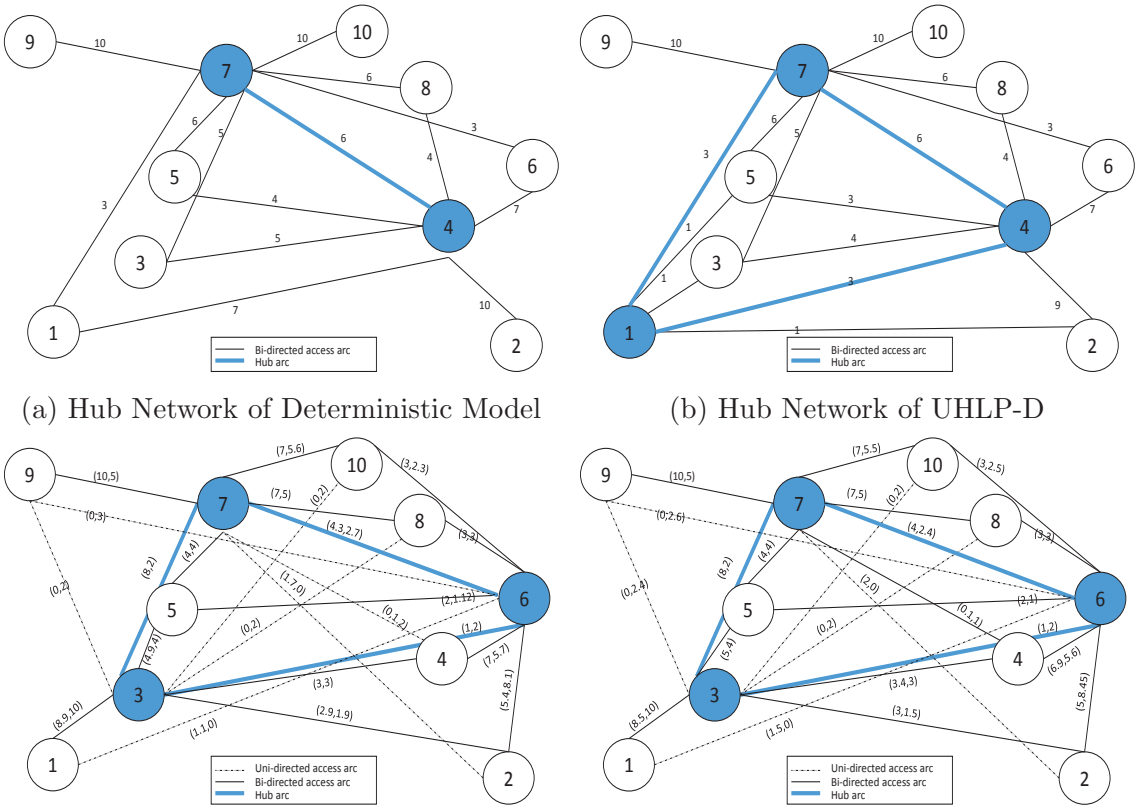
We first illustrate the differences between each of the robust counterparts' solutions and those from the deterministic model. Using a 10-node instance from the AP data set with $\Omega = 1.0$ and an uncertainty budget of $\sqrt{|A|}$ and $\sqrt{|K|}$ (10%), Figure 2.1 shows the resulting hub networks from the deterministic model and the three robust counterparts. These figures also show the number of commodities routed on each arc of the network. When this quantity is asymmetric, this information is presented as an ordered pair where the label $(a, b)$ on edge $(i, j)$ with $i > j$ is interpreted as $a$

commodities routed from $j$ to $i$ and $b$ commodities routed from $i$ to $j$.

The first evident difference is the selection of hub nodes. Note that each model, except UHLP-TC and UHLP-DTC, selects a different set of hubs. Another distinction between the robust and deterministic model solutions is the number of access arcs in the hub network. This is more palpable in the networks obtained from UHLP-TC and UHLP-DTC where most non-hub nodes use all hubs to route commodities. This comes as a result of a small percentage of commodities being routed on multiple paths through the network, see Figures 2.1c and 2.1d. This is surprising given that these problems do not consider any capacity constraints.

From a practical perspective, it is reasonable for a commodity to take multiple paths when faced with uncertain transportation costs. This strategy would allow for alternatives in the event that arcs on the cheapest route take their worst-case value. As a result, unlike the deterministic and UHLP-D solutions, the number of commodities routed on the arcs of UHLP-TC and UHLP-DTC solutions is asymmetric. For some access arcs, shown with dotted lines in Figures 2.1c and 2.1d, commodities are only sent in one direction.

To demonstrate the value of using robust counterparts to deal with uncertainty, we note the following. Using the deterministic model solution leads to a 13%, 20% and 34% cost increase when 10% of the commodities, arcs, and both simultaneously are allowed to take their worst-case value, respectively. On the other hand, the robust counterpart solutions of UHLP-D, UHLP-TC and UHLP-DTC are only 2%, 4%, and 4% more costly than the optimal solution of the deterministic model, respectively, if we assume no uncertainty exists. This shows the cost stability of networks obtained from the proposed robust counterparts compared to that from the deterministic model.




(a) Hub Network of Deterministic Model

(b) Hub Network of UHLP-D

(c) Hub Network of UHLP-TC

(d) Hub Network of UHLP-DTC

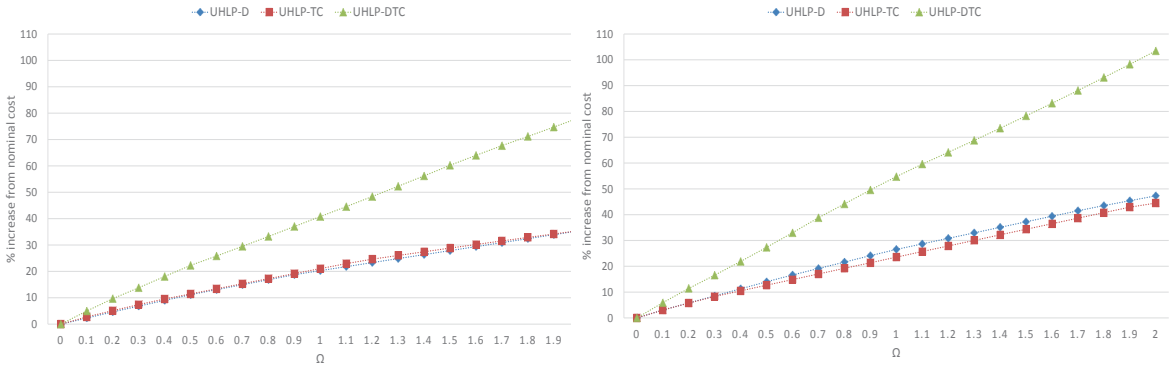
Figure 2.1: Hub Networks of robust optimization models

For the next set of experiments we use the *Set I* 20-node instances with a transfer discount factor $\alpha = 0.2$. We consider a 5% and 15% uncertainty budget, equivalent to a budget of $(\sqrt{|A|}, \sqrt{|K|})$ and $(3\sqrt{|A|}, 3\sqrt{|K|})$ respectively, for both the number of commodities and arcs. For each value, we generate instances with $\Omega \in \{0.1, 0.2, 0.3, ..., 2.0\}$ for the three robust counterparts. We note that smaller values of $\Omega$ correspond, on average, to narrower intervals whereas larger values to wider intervals.

Figure 2.2 illustrates the increase in the optimal solution values (equivalent to the total hub network cost) for the *Set I* instances of each robust counterpart as the intervals of uncertainty increase in size (i.e., $\Omega$ increases). In particular, they show the percentage increase of the optimal solution value with respect to the optimal value of

the deterministic (or nominal) UHLP.

For the considered instance, increasing $\Omega$ implies a higher optimal value for each of the robust counterparts. This is to be expected as larger values of $\Omega$ are associated with the possibility of wider intervals of uncertainty. It is interesting to note how the change in relative cost seems almost constant through the range of considered values of $\Omega$.




(a) 5% budget (b) 15% budget

Figure 2.2: Impact of interval of uncertainty for *Set I* 20-node instances with $\alpha = 0.2$

Another interesting characteristic is the interaction of the robust counterparts. At a 5% budget, UHLP-D is the least sensitive to the increase of $\Omega$ followed by UHLP-TC and the most sensitive being UHLP-DTC. At a 15% budget, the gap in behaviour between UHLP-D and UHLP-TC becomes smaller. In fact, the sensitivity relation is inverted, UHLP-D appears to be slightly more sensitive to $\Omega$ than UHLP-TC. This comes from the effect the uncertainty budget has on the robust counterparts. This is further explored in Section 2.4.2.

### 2.4.2 Impact of Budget of Uncertainty on Optimal Solutions

Our modeling approach allows decision-makers to select the level of conservatism of the robust counterparts by means of the uncertainty budgets $h_d$ and $h_W$. For some problems, it is unrealistic to assume that all values of the parameters will change

and require protection against this possibility. With this in mind, we study the effect that different uncertainty budgets have on the robust counterparts using the 20-node AP, 20-node *Set I* and 25-node CAB instances with $\Omega = 1.0$ and $\alpha = 0.2$. We test the models at budget values of $h_d \in \{0.05, 0.1, 0.15, ...1\}|A|$ and $h_W \in \{0.05, 0.1, 0.15, ...1\}|K|$. As before, we analyze the effects on UHLP-D, UHLP-TC, and UHLP-DTC with respect to the relative increase of the optimal solution value and the set of open hubs.

Figures 2.3, 2.4a and 2.4b give for the CAB, AP and *Set I* instances, respectively, the increase in the optimal solution values of each robust counterpart as the budget of uncertainty increases. In particular, they give the percentage increase of the optimal solution value with respect to the optimal value of the deterministic UHLP.

We note that while increasing the budget of uncertainty for UHLP-D leads to a greater optimal value for all budgets, for UHLP-TC there is a point after which the effect of increasing the budget becomes null and the objective value remains the same. In fact, we note from the figures that this crucial point tends to be at small values of the budget of uncertainty $h_d$. This can be partially explained by the fact that hub-and-spoke networks only use a small percentage of the total arcs of the underlying network in an optimal solution, thereby nullifying the additional protection acquired by using a higher budget. When taking into account both demand and transportation cost uncertainty in UHLP-DTC, we note a somehow similar behaviour. As the uncertainty budget becomes larger, the rate of increase in the objective function value decreases, although it does not completely become null before reaching a budget of 100%.

Another interesting characteristic is the interaction between these models. In particular, that between UHLP-D and UHLP-TC. Note that at budgets close to 0, the optimal value of UHLP-TC is greater than that of UHLP-D. However, as the budgets increase past the critical point at which the optimal value of UHLP-TC becomes stable, the optimal value of UHLP-D draws closer to that of UHLP-TC as

seen in Figure 2.3 for the CAB instance. In the AP and *Set I* instances, Figures 2.4a and 2.4b show that the optimal value of UHLP-D surpasses that of UHLP-TC.

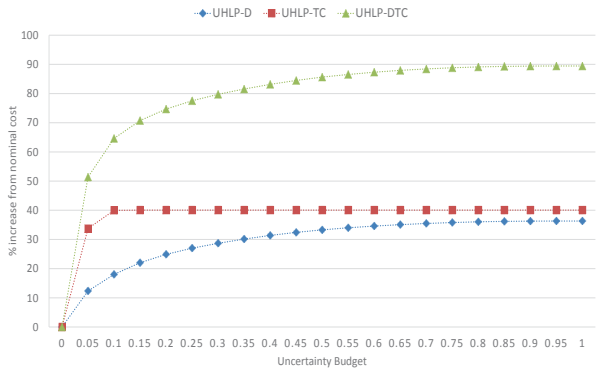
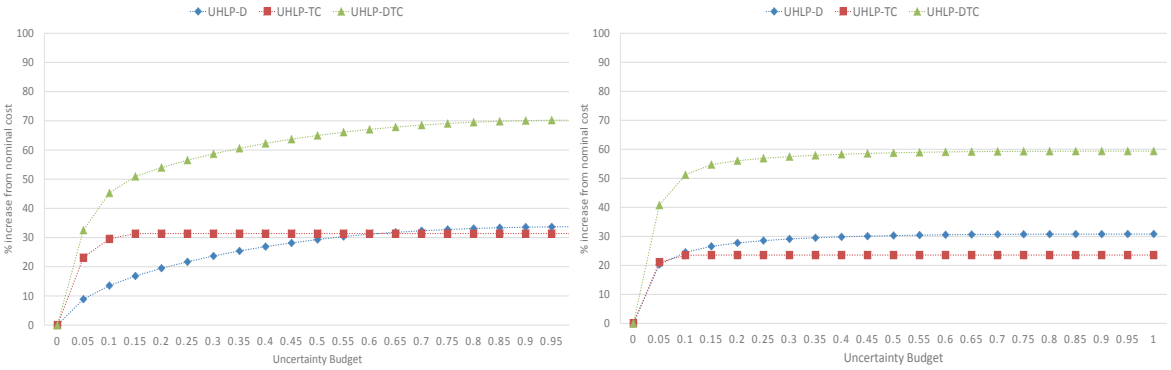



Figure 2.3: Effect of budget of uncertainty for CAB 25-node instance




(a) AP 20-node instance

(b) *Set I* 20-node instance

Figure 2.4: Effect of uncertainty budget

As for the optimal hub configurations, in the case of the CAB 25-node instances these remained unchanged for all budgets strictly greater than 0. These are $\{13, 19, 25\}$ for UHLP-D and UHLP-TC which coincide with the deterministic case, and $\{5, 13, 19, 25\}$ for the UHLP-DTC. This reaffirms the stability (or robustness) of the hub configurations for this particular dataset. In Table 2.1, we present the sets of open hubs for each of the robust counterparts for the AP and *Set I* 20-node instances. Each row corresponds to a set of values of the uncertainty budget for which the optimal solution is the same.

| Budget | UHLP-D | UHLP-TC | UHLP-DTC |
|---|---|---|---|
| 0.00 | 7 14 | 7 14 | 7 14 |
| 0.05 | 7 14 | 1 7 14 | 1 7 14 |
| [0.1, 0.55] | 7 14 | 1 7 14 | 2 7 8 14 |
| [0.6, 1.0] | 6 11 14 | 1 7 14 | 2 7 8 14 |

(a) AP 20-node instance

| Budget | UHLP-D | UHLP-TC | UHLP-DTC |
|---|---|---|---|
| 0.00 | 2 20 | 2 20 | 2 20 |
| [0.05, 0.10] | 2 4 6 11 | 2 4 13 | 2 4 6 11 |
| [0.15, 1.00] | 2 4 6 11 | 2 4 13 | 2 4 6 9 20 |

(b) *Set I* 20-node instance

Table 2.1: Optimal hub configurations for different budgets of uncertainty

We note that the budget of uncertainty does not lead to a wide variety of different hub configurations. This supports the hypothesis that the impact caused by an increase in uncertainty budgets for UHLPs becomes negligible after a relatively small value, in particular for the robust counterpart of uncertain transportation costs. It is also interesting to see how early the optimal set of open hubs of the deterministic solution is no longer optimal for the UHLP-TC and the UHLP-DTC for both the AP and *Set I* instances. For these cases, the set of open hubs begins to differ from that of the deterministic solution from as early as a 5% uncertainty budget. For the UHLP-TC, these robust hub configurations remain unchanged for increased uncertainty budgets while for UHLP-DTC only an additional hub configuration is obtained at 10 and 15% for the AP and *Set I* instances, respectively. Finally, we remark the fact that each of the robust counterparts converges to a different optimal solution when assigned a 100% uncertainty budget, thereby showing the uniqueness of each model.

### 2.4.3 Computational Performance of MIP Formulations

For this set of experiments we use the AP and Set I instances with up to 50 nodes as well as the CAB 25-node instances. We consider $\alpha \in \{0.2, 0.5, 0.8\}$, and $\Omega \in \{0.5, 1\}$. Table 2.2 summarizes the results obtained by CPLEX when solving the MIP formulations of UHLP-D and UHLP-TC. The first two columns give the information associated with the instance size $|N|$ and interval of uncertainty $\Omega$. The columns under the heading *Solved* give the number of optimal solutions found within 345,600

seconds (four days) of CPU time. The columns under the heading %LP provide the average optimality gap relative to the linear programming relaxation bound of the MIP formulations of each model. The columns under the heading *Time (sec)* give the average CPU time in seconds needed to obtain the optimal solution of each group if instances. The columns under the heading *Nodes* provide the number of explored nodes in the branch and bound.

| | | UHLP-D | | | | UHLP-TC | | | |
|---|---|---|---|---|---|---|---|---|---|
| $\|N\|$ | $\Omega$ | Solved | LP% | Time (secs) | Nodes | Solved | LP% | Time (secs) | Nodes |
| 10 | 0.5 | 6/6 | 0.06 | 0.07 | 0.00 | 6/6 | 1.11 | 0.59 | 4.67 |
| | 1 | 6/6 | 0.13 | 0.08 | 0.00 | 6/6 | 2.06 | 0.93 | 5.50 |
| 20 | 0.5 | 6/6 | 0.00 | 0.59 | 0.00 | 6/6 | 1.41 | 24.55 | 7.83 |
| | 1 | 6/6 | 0.00 | 0.67 | 0.00 | 6/6 | 4.86 | 102.47 | 44.33 |
| 25 | 0.5 | 6/6 | 0.00 | 1.52 | 0.00 | 6/6 | 1.62 | 143.93 | 10.33 |
| | 1 | 6/6 | 0.00 | 1.80 | 0.00 | 6/6 | 5.78 | 1,017.75 | 86.50 |
| 30 | 0.5 | 3/3 | 0.00 | 7.37 | 0.00 | 3/3 | 0.93 | 871.52 | 19.33 |
| | 1 | 3/3 | 0.00 | 10.14 | 0.00 | 3/3 | 3.72 | 3,996.24 | 47.33 |
| 40 | 0.5 | 6/6 | 0.01 | 52.47 | 0.50 | 6/6 | 1.70 | 5,254.59 | 21.00 |
| | 1 | 6/6 | 0.06 | 82.72 | 1.33 | 6/6 | 4.69 | 32,597.16 | 73.50 |
| 50 | 0.5 | 6/6 | 0.00 | 289.46 | 0.00 | 6/6 | 2.02 | 69,444.89 | 35.50 |
| | 1 | 6/6 | 0.01 | 474.61 | 0.00 | 1/6 | 7.04 | 185,957.97 | 107.00 |

Table 2.2: Summary results of UHLP-D and UHLP-TC

From Table 2.2 we note a significant difference in solution times between these models, in particular for the larger instances. For example, the average CPU time for 50-node instances with $\Omega = 0.5$ is under five minutes for UHLP-D while for UHLP-TC it is over 19 hours. This behaviour can be partially explained by the fact that the additional *robust* constraints (2.11) used in the MIP formulation of UHLP-TC are much more dense than the analogue *robust* constraints (2.8) of UHLP-D. Another important difference that contributes to this increased difficulty is that the LP gap for UHLP-D is significantly tighter than that of UHLP-TC, which leads to the need for further exploration in the enumeration tree for the latter as seen in Table 2.2. Another interesting observation is the increase in computation time required when

solving instances with $\Omega = 1$ when compared to $\Omega = 0.5$. This characteristic is seen for both models and shows how instances with larger uncertainty sets are more difficult to solve. Finally, we point out that 5 of the 50 node instances were not solved within the time limit for the UHLP-TC model while for the UHLP-D model 20 minutes was the longest computation time required.

The detailed results of the branch-and-cut algorithm for the UHLP-DTC are given in Table 2.3. The columns under the headings *Solved*, *%LP*, *Nodes*, and *Time (sec)*, have the same interpretation as in the previous table. The columns under the headings *Int cuts* and *Frac cuts* provide the average number of cuts added at integer and fractional solutions, respectively. The last column under the heading *% Sep time* gives the average percentage of the total time spent solving the separation problem SP.

| $\|N\|$ | $\Omega$ | Solved | %LP | Int cuts | Frac cuts | Nodes | Time (sec) | % Sep time |
|---|---|---|---|---|---|---|---|---|
| 10 | 0.5 | 6/6 | 0.89 | 97.83 | 7.17 | 5.83 | 4.20 | 42.54 |
| | 1 | 6/6 | 1.93 | 231.83 | 10.83 | 17.17 | 10.61 | 38.37 |
| 20 | 0.5 | 6/6 | 1.68 | 58.17 | 14.33 | 22.33 | 164.45 | 10.93 |
| | 1 | 6/6 | 4.80 | 437.33 | 53.83 | 197.33 | 1,997.09 | 8.11 |
| 25 | 0.5 | 6/6 | 1.67 | 65.50 | 16.67 | 32.50 | 848.30 | 7.63 |
| | 1 | 6/6 | 6.21 | 611.67 | 151.50 | 621.50 | 37,273.18 | 3.51 |
| 30 | 0.5 | 3/3 | 0.74 | 134.33 | 24.33 | 38.67 | 5,660.83 | 2.97 |
| | 1 | 2/3 | 1.40 | 153.00 | 37.00 | 73.00 | 12,268.55 | 1.92 |
| 40 | 0.5 | 6/6 | 1.46 | 117.17 | 25.17 | 54.00 | 35,489.97 | 1.45 |
| | 1 | 2/6 | 1.73 | 223.00 | 50.00 | 157.00 | 218,011.02 | 0.53 |
| 50 | 0.5 | 4/6 | N/A | 147.75 | 28.25 | 55.25 | 247,633.79 | 0.82 |
| | 1 | 1/6 | N/A | 222.00 | 22.00 | 3.00 | 322,896.84 | 0.80 |

Table 2.3: Summary results of UHLP-DTC

From Table 2.3, we observe that UHLP-DTC requires the longest solution time of all three robust counterparts. Our algorithm is capable of consistently solving instances with up to 30 nodes. After that, it is capable of solving 10 out of 12 instances with 40 and 50 nodes with $\Omega = 0.5$ but is only capable of solving 3 out of

12 instances with $\Omega = 1$ in four days of CPU time. The long solution times come as a result of the need for the algorithm to explore more nodes in the enumeration tree to prove optimality and having to add many dense cuts to a large model. Note that although many cuts are added, very little time is spent solving the separation problem, showing that despite the theoretical NP-hardness of SP, CPLEX is still able to solve it efficiently. Finally, we point out the relatively small %LP gaps of this formulation.

### 2.4.4 A Comparison of Deterministic, Stochastic and Robust Solutions

In this final set of computational experiments, we compare the solutions obtained from the deterministic UHLP, a two-stage stochastic variant of the UHLP presented in Contreras et al. [50], and our proposed robust counterpart UHLP-TC. Our analyses only consider the hub configurations and disregard the routing decisions obtained from each of the approaches. We point out that this is not equivalent to using an adaptive robust approach as those in Bertsimas et al. [27] and Zeng and Zhao [180]. For our model, both location and routing decisions must be made in a single stage prior to any revelation of the uncertain parameters unlike two stage stochastic and adaptive robust optimization approaches. While one expects that this difference leads to a poor performance of our solutions in a two-stage setting, we show that their performance is close to that of solutions obtained from the two-stage stochastic approach.

To evaluate these solutions as fairly as possible we define $W_k(\xi)$ and $d_{ij}(\omega)$ as independent random variables following both uniform distributions over the intervals $W_k(\xi) \sim U[W_k^L - W_k^\Delta, W_k^L + W_k^\Delta]$ and $d_{ij}(\omega) \sim U[d_{ij}^L - d_{ij}^\Delta, d_{ij}^L + d_{ij}^\Delta]$ and normal distributions defined by $W_k(\xi) \sim N\left(W_k^L, \sqrt{(2W_k^\Delta)^2/12}\right)$ and $d_{ij}(\omega) \sim N\left(d_{ij}^L, \sqrt{(2d_{ij}^\Delta)^2/12}\right)$, respectively.

As mentioned in Section 2.2, Contreras et al. [50] show that when considering de-

mand uncertainty, the two-stage stochastic model of UHLP is actually equivalent to its associated expected value problem in which the uncertain demand is replaced by its expectation. However, when considering both demand and independent transportation cost uncertainty this situation does not hold. For the sake of completeness, we next present the stochastic model with demand and transportation cost uncertainty.

We assume that the random parameters $d_{ij}(\omega)$ and $W_k(\xi)$ are independent and use the notation $C_{ak}(\omega) = \chi d_{o(k)i}(\omega) + \alpha d_{ij}(\omega) + \delta d_{jd(k)}(\omega)$ where $a = (i, j)$. We define $z_i$ as first stage variables and $x_{ak}(\omega, \xi)$ as second stage variables that adjust depending on the realization of the uncertainty parameter. The *uncapacitated hub location problem with stochastic demands and transportation costs* (UHLP-SDTC) can be stated as follows:

$$\text{minimize} \quad \sum_{i \in N} f_i z_i + E_{\omega,\xi}[\sum_{k \in K} \sum_{a \in A} C_{ak}(\omega) W_k(\xi) x_{ak}(\omega, \xi)] \tag{2.17}$$

$$\text{subject to} \quad \sum_{a \in A} x_{ak}(\omega, \xi) = 1 \qquad \forall k \in K, \omega \in \Omega, \xi \in \Xi \tag{2.18}$$

$$\sum_{a \in A: i \in a} x_{ak}(\omega, \xi) \leq z_i \qquad \forall k \in K, i \in N, \omega \in \Omega, \xi \in \Xi \tag{2.19}$$

$$z_i \in \{0, 1\} \qquad \forall i \in H \tag{2.20}$$

$$0 \leq x_{ak}(\omega, \xi) \leq 1 \qquad \forall a \in A, k \in K, \omega \in \Omega, \xi \in \Xi. \tag{2.21}$$

Since both parameters are independently distributed then $E_{\omega,\xi} = E_\omega E_\xi$. Also, note that for a first-stage vector $z$ and a fixed realization of distances $\omega$, the optimal route $x_{ak}(\omega, \xi)$ is the same independent of the realization of $\xi$, i.e., $x_{ak}(\omega, \xi) =$

$x_{ak}(z,\omega)$. Therefore, given $(z,\omega)$ we have

$$E_\omega E_{\xi|\omega}\left[\sum_{k,a} C_{ak}(\omega)W_k(\xi)x_{ak}(\omega,\xi)\right] = E_\omega E_{\xi|\omega}\left[\sum_{k,a} C_{ak}(\omega)W_k(\xi)x_{ak}(z,\omega)\right] \quad (2.22)$$

$$= E_\omega\left[\sum_{k,a} E_\xi\left[C_{ak}(\omega)W_k(\xi)x_{ak}(z,\omega)\right]\right] \quad (2.23)$$

$$= E_\omega\left[\sum_{k,a} C_{ak}(\omega)W_k^L x_{ak}(z,\omega)\right], \quad (2.24)$$

where (2.22) comes from the independence of $x_{ak}(\omega,\xi)$ of the realization of $\xi$, (2.23) comes from the independence of $\omega$ and $\xi$, and (2.24) is from our assumptions on the uncertainty distributions of the $W_k(\xi)$. As a consequence, UHLP-SDTC reduces to the uncapacitated hub location problem with stochastic independent transportation costs presented in Contreras et al. [50]. Hence, we use the sample average approximation algorithm proposed in Contreras et al. [50] to obtain $\epsilon$-optimal solutions for the stochastic counterparts. In our experiments, we take 20 samples of size 1,000 to estimate a lower bound and a larger sample size of 100,000 to estimate an upper bound.

Our comparison methodology is comprised of two steps. The first is to note when the solutions obtained from the stochastic, robust and deterministic models coincide with respect to the open hubs. The second is to study instances for which these differ and compare each proposed solution in both a risk-neutral and worst-case scenario. This analysis is done with the AP and *Set I* 20-node instances as well as the CAB 25-node instance for values of $\Omega \in \{0.5, 1.0, 1.5, 2.0\}$ and $\alpha \in \{0.2, 0.5, 0.8\}$. To obtain the corresponding robust solutions, we solve UHLP-TC at budget values of $\{0.1, 0.2, 0.3, ..., 1\}|A|$ and consider all hub configurations obtained for these. Tables 2.4, 2.5 and 2.6 show the optimal hub configurations obtained by each approach for the CAB, AP and *Set I* instances, respectively. Note that for the robust approach, if more than one hub configuration is obtained throughout the budget values tested,

then all distinct configurations will be taken into account in the table.

| $\Omega$ | $\alpha$ | Det | Stoch U(A,B) | Stoch N($\mu, \sigma$) | Robust |
|---|---|---|---|---|---|
| 0.5 | 0.2, 0.5, 0.8 | 13 19 25 | 13 19 25 | 13 19 25 | 13 19 25 |
| 1.0 | 0.2, 0.5, 0.8 | 13 19 25 | 13 19 25 | 13 19 25 | 13 19 25 |
| 1.5 | 0.2, 0.5, 0.8 | 13 19 25 | 13 19 25 | 13 19 25 | 5 13 19 25 |
| 2.0 | 0.2, 0.5, 0.8 | 13 19 25 | 5 19 25 | 5 19 25 | 5 13 19 25 |

Table 2.4: Optimal hubs for each approach with CAB instances

| $\Omega$ | $\alpha$ | Det | Stoch U(A,B) | Stoch N($\mu, \sigma$) | Robust |
|---|---|---|---|---|---|
| 0.5 | 0.2, 0.5, 0.8 | 7 14 | 7 14 | 7 14 | 7 14 |
| 1 | 0.2 | 7 14 | 7 14 | 7 14 | 1 7 14 |
| 1 | 0.5, 0.8 | 7 14 | 7 14 | 7 14 | 7 14 |
| 1.5 | 0.2 | 7 14 | 7 14 | 7 14 | 1 7 8 14 |
| 1.5 | 0.5 | 7 14 | 7 14 | 7 14 | 1 8 11 14<br>1 7 14 |
| 1.5 | 0.8 | 7 14 | 7 14 | 7 14 | 6 8 14 |
| 2 | 0.2 | 7 14 | 7 11 14 | 7 11 14 | 1 7 8 14 |
| 2 | 0.5 | 7 14 | 7 11 14 | 7 14 | 1 8 11 14<br>1 7 8 14 |
| 2 | 0.8 | 7 14 | 7 11 14 | 7 11 14 | 1 8 11 14<br>1 7 8 14<br>7 8 14 |

Table 2.5: Optimal hubs for AP instances

| $\Omega$ | $\alpha$ | Det | Stoch U(A,B) | Stoch N($\mu, \sigma$) | Robust |
|---|---|---|---|---|---|
| 0.5 | 0.2, 0.5 | 2 20 | 2 20 | 2 20 | 2 4 13 |
| 0.5 | 0.8 | 2 20 | 2 20 | 2 20 | 2 20 |
| 1 | 0.2, 0.5 | 2 20 | 2 20 | 2 20 | 2 4 13 |
| 1 | 0.8 | 2 20 | 2 20 | 2 20 | 2 6 11<br>2 13 |
| 1.5 | 0.2 | 2 20 | 2 11 20 | 2 4 20 | 2 4 13 |
| 1.5 | 0.5 | 2 20 | 2 11 20 | 2 11 20 | 2 4 6 11<br>2 4 13 |
| 1.5 | 0.8 | 2 20 | 2 11 20 | 2 20 | 2 6 11 19<br>2 4 6 |
| 2 | 0.2 | 2 20 | 2 11 20 | 2 4 13 | 2 4 6 9 20 |
| 2 | 0.5 | 2 20 | 2 11 20 | 2 19 20 | 2 4 6 11<br>2 4 13 |
| 2 | 0.8 | 2 20 | 2 11 20 | 2 19 20 | 2 6 11 19<br>2 4 6 11<br>2 13 19 |

Table 2.6: Optimal hubs for *Set I* instances

From Table 2.4 we note that for the CAB data set, the open hubs of all three approaches coincide for all instances with $\Omega \in \{0.5, 1.0\}$. This shows the stability of the open hubs for this dataset requiring an increase of 150% of the nominal cost for the hub configuration to change. Even at a 150% level of uncertainty, the stochastic approach still keeps the same hub configuration as the deterministic case while the robust counterparts open only one additional hub. Table 2.5 shows that for the case of the AP instances, for $\Omega \in \{0.5, 1.0\}$ and $\alpha \in \{0.5, 0.8\}$, all three approaches give the same optimal hub configuration. At all other values, the robust solutions differ from those obtained from the other approaches while the stochastic approach opens the same hubs as the deterministic model for all values except for $\Omega = 2.0$. For the *Set I* instances, Table 2.6 shows that all three approaches give the same hub configuration only when $\Omega = 0.5$ and $\alpha = 0.8$. The robust approach differs from the other approaches for all other tested values while the stochastic approach obtains different hub configurations when $\Omega \in \{1.5, 2.0\}$.

For the next step in our analysis, we ignore the two-stage stochastic model with uncertain parameters following a normal distribution. The reason is that, in our view, the uniform distribution is a more fair comparison to the interval uncertainty that we assume. From each dataset, we arbitrarily select a value of $\Omega$ and $\alpha$ that yield different hub configurations for each approach to measure their performance in a worst-case and risk-neutral setting. To evaluate in a worst-case framework, we fix the hub variables $z_i$ from UHLP-TC to the hub configurations obtained from each approach and solve the remaining model for budgets of $\{0.1, 0.2, 0.3, ..., 1.0\}$, while in the risk-neutral setting we use Monte Carlo simulation to evaluate the average performance of all hub configurations assuming realizations follow the previously mentioned normal distribution.

A summary of the performance of optimal hub configurations obtained from each approach (deterministic, stochastic, and robust) in a worst-case setting for the CAB, AP and *Set I* instances is given in Figures 2.5, 2.6a and 2.6b, respectively. In particular, these figures give the percentage increase in the optimal solution value with respect to the deterministic (nominal) cost for different values of the budget of uncertainty. Table 2.7 shows a similar analysis for a risk-neutral setting.

For the case of the *Set I* instance, note that in a worst-case scenario, the relative increase in optimal value is almost 30% greater for the hub configuration from the stochastic model than that from the robust model. Also, note that the hub configuration of the deterministic model has the worst performance of the three in this setting. However, observe that in the case of the risk-neutral setting the average performance of the hub configuration of the robust model is only 5.56% worse than that of the stochastic model while that of the deterministic model is 3.38% worse.

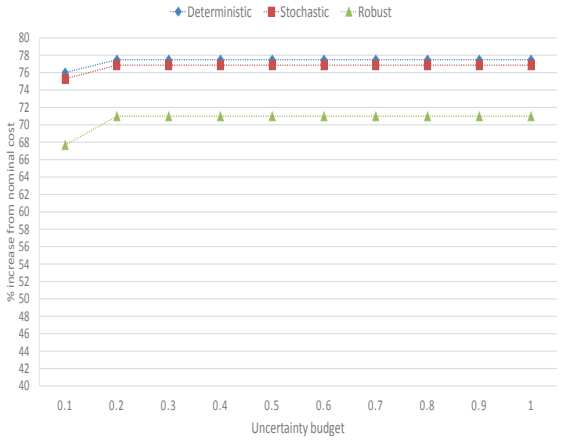
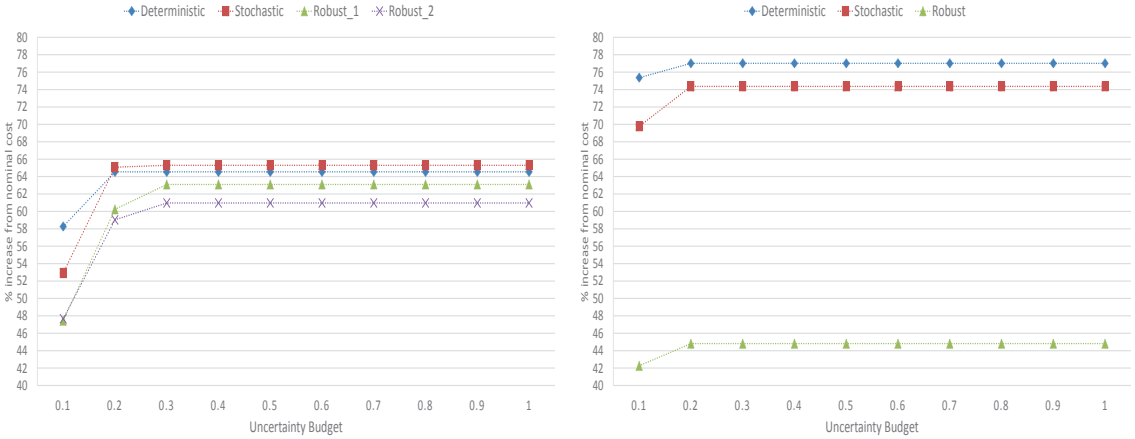



Figure 2.5: Solution performance in worst-case setting for CAB instance with $\Omega = 2$ and $\alpha = 0.2$




(a) AP instance with $\Omega = 2$ and $\alpha = 0.2$ (b) *Set I* instance with $\Omega = 2$ and $\alpha = 0.5$

Figure 2.6: Solution performance in worst-case setting

| Instance | Approach | Open Hubs | Exp. Cost | $\sigma$ | %Increase |
|---|---|---|---|---|---|
| CAB | Stochastic | 5 19 25 | 1,421,798.65 | 824.57 | |
| | Deterministic | 13 19 25 | 1,424,828.17 | 866.89 | 0.21 |
| | Robust | 5 13 19 25 | 1,433,177.30 | 758.16 | 0.80 |
| AP | Stochastic | 7 11 14 | 199,390.29 | 95.38 | |
| | Deterministic | 7 14 | 200,442.11 | 106.96 | 0.53 |
| | Robust1 | 1 8 11 14 | 214,087.97 | 78.69 | 7.37 |
| | Robust2 | 1 7 8 14 | 216,080.27 | 74.48 | 8.37 |
| *Set I* | Stochastic | 2 11 20 | 40,576.19 | 31.96 | |
| | Deterministic | 2 20 | 41,946.05 | 38.78 | 3.38 |
| | Robust | 2 4 6 9 20 | 42,832.30 | 17.07 | 5.56 |

Table 2.7: Solution performance in a risk-neutral setting

This shows how the hub configuration of our robust approach significantly outperforms those obtained from the stochastic and deterministic approaches in a worst-case setting and is only slightly worse in a risk-neutral setting. A somehow similar behaviour is noted for the CAB instance as seen in Figure 2.5 and Table 2.7, although in a risk-neutral setting both deterministic and robust models have an increase that does not exceed 1%. For the AP instance the hub configuration of the stochastic approach is the worst performing in a worst-case setting as seen in Figure 2.6a while the hub configurations of the robust approach are on average under 8.4% more costly than the stochastic solution in a risk-neutral setting as seen in Table 2.7.

## 2.5 Conclusion

In this paper we introduced robust counterparts of the well-known uncapacitated hub location problem with multiple assignments for the cases considering uncertain demands, transportation costs, and both simultaneously. We considered an interval of uncertainty for demands and transportation costs and used a budget of uncertainty to control the level of conservatism in solution networks. We presented mixed integer linear programming formulations for each of the considered robust counterparts.

We performed extensive computational experiments to evaluate the performance of our formulations and to study the impact of the intervals of uncertainty and of the budgets of uncertainty in the optimal hub configurations. Finally, we compared the performance between solutions obtained from deterministic, stochastic and robust models in both worst-case and risk-neutral settings.

The proposed robust counterparts have provided important insights on the ideal topology for uncapacitated hub networks when the demand quantity, transportation costs and both simultaneously are uncertain. In particular, we note that when transportation costs are uncertain, commodities are routed through multiple paths; a behaviour mostly seen in capacitated models. An interesting question to explore is the adaptation of these robust counterparts for capacitated hub location models.

In terms of algorithmic refinements, there are possibly three lines of research that could lead to an improved solution method, in particular for larger models. The first deals with cut density. As seen in our computational experiments, cut density is one of the main contributing factors to the long computation times. Methods that substitute these dense constraints for a family of equivalent or possibly weaker but sparser constraints could lead to a faster method. Second, the large number of variables in these models also leads to longer solution times. Methods such as variable elimination and on the fly variable generation seem to be promising lines to circumvent this and speed up the solution time of the LP subproblems. Finally, along with having a large number of variables, these models also contain a large number of constraints. To reduce this, the relaxation and on-the-fly generation of some of the families of constraints seems to be a promising approach.

# Chapter 3

# Exact Algorithms for Multicommmodity Uncapacitated Fixed-charge Network Design


## Abstract

This paper presents two exact algorithms based on Benders decomposition for solving the multicommodity uncapacitated fixed-charge network design problem. The first is a branch-and-cut algorithm based on a Benders reformulation enhanced with an in-tree matheuristic to obtain improved feasible solutions and valid inequalities in the projected master problem space to close the linear programming gap. In addition, implementation details crucial to the algorithm's efficiency such as cut and core point selection criteria are addressed. The second exact algorithm exploits the problem's structure to combine a cut-and-solve strategy with Benders decomposition. Extensive computational experiments show both exact algorithms provide a speedup of between one and two orders of magnitude compared to a state-of-the-art general-purpose MIP solver.

## 3.1 Introduction

*Network design problems* (NDPs) lie at the heart of designing and operating efficient systems in several sectors such as personnel scheduling [13, 19], service network design [8, 60, 61], logistics network design [54, 91, 162], and transportation [128, 138]. They are able to capture the system-wide interactions between strategic and operational decisions, namely arc activation and routing, to ensure cost-effective paths among a selected set of nodes. NDPs can be classified into *single* and *multicommodity* variants depending on the characteristics of the demand. In single-commodity problems, the demand at each node can be satisfied by any of the other nodes' supply since they all route the same commodity. In multicommodity problems, demand is expressed as origin-destination (OD) pairs where a destination node's demand must be satisfied by a corresponding origin node's supply.

In this paper we focus on a fundamental NDP: the *multicommodity uncapacitated fixed-charge network design problem* (MUFND). The problem is defined on a directed graph and considers a set of commodities modeled by OD pairs, each with an origin node, a destination node, and a demand quantity. The objective is to install a subset of arcs to route all commodities from their origins to their destinations at minimal cost. The MUFND is $\mathcal{NP}$-hard [109] and generalizes a large class of well-known problems such as the *traveling salesman problem*, the *uncapacitated lot-sizing problem*, and the *Steiner network design problem* [146].

At the same time, the MUFND is generalized by the multicommodity capacitated fixed-charge network design problem. The latter has been extensively studied from different research directions. Some authors have approached this problem from a polyhedral perspective, proposing new families of valid inequalities that strengthen well-known mixed integer formulations [9, 10, 29, 98, 153]. Recently, Chouman et al. [43] provided insight on the computational impact that commodity representations have on the efficiency of five families of these valid inequalities. Others have focused

on the development of decomposition methods [62, 64, 85–87, 156] that exploit the problem structure to decompose the model into smaller subproblems. Despite the significant contributions presented in these papers, solving the capacitated network design problem to proven optimality in reasonable time still remains an open problem.

Another line of research for solving the capacitated variant is the use of heuristic algorithms to obtain high quality solutions. Among these are the slope scaling heuristics [63, 110, 112, 113]; cycle-based and other neighbourhood searches [93, 94, 151, 175]; and matheuristics that exploit available mathematical programming software [90, 103, 139, 160].

In the case of the MUFND, the first proposed solution algorithm is an add-drop heuristic by Billheimer and Gray [31]. Other heuristics are those of Dionne and Florian [69], Boffey and Hinxman [35], Los and Lardinois [124], and Kratica et al. [115]. Lamar et al. [119] proposed a novel form of iteratively obtaining strengthened dual bounds from a weaker formulation by adjusting artificial capacity constraints. Balakrishnan et al. [12] presented a dual ascent algorithm and a primal heuristic to obtain solutions for large-scale instances with up to 600 arcs and 1,560 commodities. Their method obtains solutions that are between 1% and 4% away from optimality in less than 150 seconds of computing time. With respect to exact methods, Magnanti et al. [126] developed a tailored Benders decomposition for the variant with undirected design decisions of the MUFND. They were able to solve instances with up to 130 arcs and 58 commodities to proven optimality. Holmberg and Hellstrand [105] used a Lagrangean branch-and-bound algorithm to solve directed instances with up to 1,000 arcs and 600 commodities. Recently, Fragkos et al. [84] used Benders decomposition to solve a multi-period extension of the MUFND. They experimented with the use of Pareto-optimal cuts and with the unified cut approach of Fischetti et al. [78], obtaining significant computational gains with the latter on instances with up to 318 arcs, 100 commodities and 108 time periods. To the best of our knowledge, these are

the current state-of-the-art exact methods for the undirected and directed variants of the MUFND.

Benders decomposition has been an effective tool for solving several classes of network design problems with various applications [58]. Some fields in which it has recently been applied are closed loop supply chains [107], hazardous material transportation [81], and health services [179]. It has also recently been proven effective in solving fundamental network design problems such as the optimum communication spanning tree problem [182] and extensions such as the multi-layer [82], hop-constrained [37], and multi-period [84] network design problems. In the last few years, it has also been applied to NDPs with parameter uncertainty as in Keyvanshokooh et al. [111], Lee et al. [121] and Rahmaniani et al. [155]. Other applications of Benders decomposition to fixed-charge NDPs can be found in Costa [58] while Rahmaniani et al. [154] provides a survey on the algorithm and its use in optimization problems.

This paper revisits the use of Benders decomposition proposed by Magnanti et al. [126] to solve the undirected design variant of the MUFND. As in Fischetti et al. [79], our purpose is to redesign this once discarded approach for solving the MUFND to exploit the state-of-the-art of algorithmic and computational resources. The resulting Benders algorithms use branch-and-cut [148], local branching [76], and cut-and-solve [45] procedures implemented within the *cut callback* framework available in today's general purpose mixed integer programming solvers. We present, in detail, the nuances of adopting these tools and propose novel refinements to reduce the computation time required to solve the MUFND with directed arc design decisions.

We present two exact algorithms based on the Benders reformulation of a well-known mixed integer programming model of the directed MUFND. Both algorithms solve the linear relaxation of the Benders reformulation with a cutting-plane procedure to obtain Pareto-optimal cuts and cutset inequalities at each node of the enumeration

tree. To accelerate the algorithms' convergence we introduce new valid inequalities referred to as *Benders lift-and-project cuts* to improve the linear programming relaxation and an in-tree matheuristic that finds better feasible solutions by using path information generated while exploring the branch-and-bound tree.

The first algorithm, referred to as a branch-and-Benders-cut algorithm, solves the Benders reformulation in one enumeration tree. We address critical implementation details that should be considered when separating cuts at fractional and integer points such as core point selection for Pareto optimal cuts and propose a tailored core point selection criterion that provides a significant speed-up for solving the MUFND.

The second method is based on the combination of a modified cut-and-solve scheme [45] and our branch-and-Benders-cut algorithm. The method iteratively restricts the potential design arcs to solve smaller problems that produce a sequence of feasible solutions with non-increasing objective function value. The algorithm also allows for the recycling of Benders cuts generated in previous iterations thereby saving computational effort.

We report computational experience on several sets of benchmark instances to assess the performance of our algorithms. The proposed exact methods are up to three orders of magnitude faster than the state-of-the-art MIP solver CPLEX 12.7.1 and solve instances of larger size than those previously presented. This computational contribution is accompanied by methodological insights such as the simultaneous use of a path and arc-based formulation for the MUFND, the introduction of *Benders lift-and-project cuts* to reduce the linear programming gap, and the hybridization of two well-known mixed integer programming tools.

The remainder of the paper is organized as follows. Section 3.2 provides a formal definition of the MUFND and presents the arc-based formulation. Section 3.3 describes the Benders reformulation for the MUFND while Section 3.4 details the enhancements implemented in our branch-and-cut algorithm. In Section 3.5, we present

our second algorithmic framework, a hybrid cut-and-solve Benders algorithm. Summarized results of our computational experiments are given in Section 3.6, while Section 3.7 presents concluding remarks and future lines of research.

## 3.2 Problem definition

The MUFND is defined on a directed graph $G = (N, A)$ where $N$ is a set of nodes, $A$ is a set of arcs and $K$ is a set of commodities each defined by the tuple $(o_k, d_k, W_k)$ representing the origin, destination, and demand quantity of a commodity $k \in K$, respectively. The key feature of this problem is its use in evaluating the trade-off between infrastructure investment and operational costs. The former is modeled by the fixed cost paid for using an arc $f_{ij}$ joining node $i$ to node $j$. The latter is modeled by a linear transportation cost $c_{ij}^k$ paid per unit of commodity $k$ routed on arc $(i, j)$. The goal is to route all commodities from origins to destinations at minimal cost.

Two well-known mixed integer models for this problem are the formulations with *aggregated* and *disaggregated* constraints. Both use a set of binary variables $y_{ij}$ to model whether arc $(i, j)$ is installed or not and a set of continuous variables $x_{ij}^k$ to represent the fraction of commodity $k$'s demand routed on arc $(i, j)$. In this study, we use the disaggregated formulation since its tighter linear programming (LP) relaxation is preferred when applying Benders decomposition [127]. The MUFND can be formulated as follows:

$$\text{(P)} \quad \text{minimize} \sum_{(i,j)\in A} f_{ij}y_{ij} + \sum_{k\in K}\sum_{(i,j)\in A} W^k c_{ij}^k x_{ij}^k \tag{3.1}$$

$$\text{subject to} \quad \sum_{j\in N} x_{ji}^k - \sum_{j\in N} x_{ij}^k = \begin{cases} -1 & \text{if } i = o_k \\ 1 & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, k \in K \tag{3.2}$$

$$x_{ij}^k \leq y_{ij} \quad \forall (i,j) \in A, k \in K \tag{3.3}$$

$$x_{ij}^k \geq 0 \qquad \forall (i,j) \in A, k \in K \tag{3.4}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{3.5}$$

The objective function (4.2) is the total cost of the network including both the installation and routing costs for all arcs and commodities. Flow conservation constraints (3.2) ensure that the demand for all commodities is routed from origin to destination. Constraint set (3.3) assures that no flow is sent through an arc that has not been installed, while (3.4) and (3.5) are the non-negativity and integrality conditions on $x$ and $y$, respectively.

Note that depending on the instance data, P is a valid formulation for the Steiner tree problem (all commodities share the same origin and no transportation costs exist) or the travelling salesman problem (all arcs have the same fixed cost, the underlying graph is complete, commodities are sent between every pair of nodes, and transportation costs are commodity independent) [105]. This shows the wide range of special cases generalized by the MUFND and as such the inherent difficulty in developing an efficient exact algorithm for this general model.

## 3.3 Benders decomposition for the MUFND

Benders decomposition is a well-known solution method for mixed integer programming problems [22]. It splits large formulations into two problems, an integer master problem and a linear subproblem. The principle behind Benders decomposition is the projection of a large problem into a smaller subspace, namely the space of the integer constrained variables. As a consequence, the projected model contains an exponential number of constraints known as Benders cuts, indexed by the extreme points and extreme rays of a special linear programming problem known as the dual subproblem (DSP) or slave problem. Noting that not all Benders cuts are necessary to obtain the optimal solution, Benders [22] proposed to relax these and iteratively solve the integer master problem to obtain a lower bound on the integral optimal solution value and then substitute the solution into the dual subproblem thereby obtaining an upper bound and a Benders cut to be added to the master problem. This is to be repeated until the upper and lower bounds are within a given optimality tolerance $\epsilon$. In this section, we present the derivation of the Benders reformulation of P and the use of cutset inequalities to replace the classic Benders feasibility cuts.

### 3.3.1 Benders reformulation

The following steps describe the process of applying Benders decomposition to formulation P of the MUFND. Note that by fixing $y = \bar{y}$, where $\bar{y} \in Y$ and $Y = \mathbb{B}^{|A|}$ denotes the set of binary vectors associated with the $y_{ij}$ variables, we obtain a linear program in $x$ that is easily solved. This new linear program will be denoted as the primal subproblem (PSP) and has the following form:

$$\text{(PSP)} \quad \text{minimize} \sum_{k \in K} \sum_{(i,j) \in A} W^k c_{ij}^k x_{ij}^k$$

$$\text{subject to} \quad \sum_{j \in N} x_{ji}^k - \sum_{j \in N} x_{ij}^k = \begin{cases} -1 & \text{if } i = o_k \\ 1 & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, k \in K \tag{3.6}$$

$$x_{ij}^k \leq \bar{y}_{ij} \qquad \forall (i,j) \in A, k \in K \tag{3.7}$$

$$x_{ij}^k \geq 0 \qquad \forall (i,j) \in A, k \in K.$$

Note that PSP can be split into $|K|$ subproblems $\text{PSP}_k$, one for each commodity. Let $\lambda$ and $\mu$ denote the dual variables of constraints (3.6) and (3.7), respectively. From strong duality, each $\text{PSP}_k$ can be substituted by its linear programming dual, denoted as $\text{DSP}_k$, of the form:

$$\text{(DSP}_k\text{)} \quad \text{maximize } \lambda_{d_k}^k - \lambda_{o_k}^k - \sum_{(i,j) \in A} \mu_{ij}^k \bar{y}_{ij} \tag{3.8}$$

$$\text{subject to} \quad \lambda_j^k - \lambda_i^k - \mu_{ij}^k \leq W^k c_{ij}^k \qquad \forall (i,j) \in A$$

$$\mu_{ij}^k \geq 0 \qquad \forall (i,j) \in A$$

$$\lambda_i^k \in \mathbb{R} \qquad \forall i \in N.$$

From Farkas' Lemma, we know that for a given $k \in K$, $\text{PSP}_k$ is feasible if and only if

$$\bar{y} \in R_k = \left\{ y \in Y |\ 0 \geq \lambda_{d_k}^k - \lambda_{o_k}^k - \sum_{(i,j) \in A} \mu_{ij}^k \bar{y}_{ij},\ \forall (\lambda^k, \mu^k) \in \Theta \right\},$$

where $\Theta$ is the recession cone of $\text{DSP}_k$. The inequalities that define $R_k$ are known as Benders feasibility cuts and, although by Farkas' Lemma there exists an infinite

number of them, only those associated with the (finite) set of extreme rays are necessary. Therefore, we use the representation of each polyhedron associated with each $\text{DSP}_k$ in terms of its extreme points and extreme rays to determine whether PSP is infeasible or feasible and bounded.

Let $\text{Ext}(\text{DSP}_k)$ and $\text{Opt}(\text{DSP}_k)$ denote the sets of extreme rays and extreme points of $\text{DSP}_k$, respectively. If, for a given $y \in Y$, there exists at least one $k \in K$ and one extreme ray $(\lambda, \mu) \in \text{Ext}(\text{DSP}_k)$ for which

$$0 < \lambda^k_{d_k} - \lambda^k_{o_k} - \sum_{(i,j)\in A} \mu^k_{ij} \bar{y}_{ij},$$

then $\text{DSP}_k$ is unbounded and PSP is infeasible. However, if

$$0 \geq \lambda^k_{d_k} - \lambda^k_{o_k} - \sum_{(i,j)\in A} \mu^k_{ij} \bar{y}_{ij},$$

for each $k \in K$ and each extreme ray $(\lambda, \mu) \in \text{Ext}(\text{DSP}_k)$, then all $\text{DSP}_k$ are bounded and the PSP is feasible. The optimal value of each $\text{DSP}_k$ is then equal to

$$\max_{(\lambda,\mu)\in \text{Opt}(\text{DSP}_k)} \lambda^k_{d_k} - \lambda^k_{o_k} - \sum_{(i,j)\in A} \mu^k_{ij} \bar{y}_{ij}.$$

Using continuous variables $z_k$ for the transportation cost of each commodity $k \in K$, the *Benders reformulation* of P is

$$(\mathrm{MP}_0) \quad \text{minimize} \sum_{(i,j)\in A} f_{ij}y_{ij} + \sum_{k\in K} z_k \tag{3.9}$$

$$\text{subject to} \quad z_k \geq \lambda^k_{d_k} - \lambda^k_{o_k} - \sum_{(i,j)\in A} \mu^k_{ij} y_{ij} \qquad \forall k \in K, (\lambda,\mu)_k \in \mathrm{Opt}(\mathrm{DSP}_k) \tag{3.10}$$

$$0 \geq \bar{\lambda}^k_{d_k} - \bar{\lambda}^k_{o_k} - \sum_{(i,j)\in A} \bar{\mu}^k_{ij} y_{ij} \qquad \forall k \in K, (\bar{\lambda},\bar{\mu})_k \in \mathrm{Ext}(\mathrm{DSP}_k) \tag{3.11}$$

$$z \in \mathbb{R}^{|K|}$$

$$y \in \{0,1\}^{|A|}.$$

$\mathrm{MP}_0$, also known as the Benders master problem, exploits the decomposability of the subproblems by disaggregating the feasibility and optimality cuts per commodity. This type of *multi-cut* reformulation leads to a better approximation of the transportation costs at each iteration, which has been empirically shown to improve solution times [49, 126].

Exploiting the structure of the MUFND, we replace the Benders feasibility cuts (3.11) with cutset inequalities which are sufficient to guarantee the feasibility of PSP [59]. The advantage of using cutset inequalities is that they can be efficiently separated by solving a minimum cut problem over an auxiliary network. Algorithms such as the Edmonds-Karp algorithm [70] and breadth first search are efficient in separating cutset inequalities for fractional and integer solutions, respectively. With this in mind, we substitute the use of Benders feasibility cuts with cutset inequalities yielding the following final Benders reformulation:

$$\text{(MP)} \quad \text{minimize} \sum_{(i,j)\in A} f_{ij}y_{ij} + \sum_{k\in K} z_k \tag{3.12}$$

$$\text{subject to} \quad z_k \geq \lambda^k_{d_k} - \lambda^k_{o_k} - \sum_{(i,j)\in A} \mu^k_{ij} y_{ij} \qquad \forall k \in K, (\lambda,\mu)_k \in \text{Opt}(\text{DSP}_k) \tag{3.13}$$

$$\sum_{(i,j)\in\delta(S)} y_{ij} \geq 1 \qquad \forall S \in \Delta_K \tag{3.14}$$

$$z \in \mathbb{R}^{|K|}$$

$$y \in \{0,1\}^{|A|},$$

where $\delta(S) = \{(i,j) \in A | i \in S, j \in N \backslash S\}$ and $\Delta_K$ is the set of subsets $S \subset N$ such that $o_k \in S$ and $d_k \notin S$ for some $k \in K$.

## 3.4 A branch-and-Benders-cut algorithm for the MUFND

Since its introduction, Benders decomposition has been successfully used to solve several difficult problems such as airline scheduling [53, 150], facility location [79, 91, 147], hub network design [49], and fixed-charge network design [58]. Although the initially proposed algorithm suffered from slow convergence, through the years researchers have devised enhancements to significantly increase its speed. Recent implementations of Benders decomposition incorporate additional strategies such as the generation of strong cuts, cut selection, stabilization, lower bound reinforcing, and solving one enumeration tree [3, 33, 37, 79, 140, 147, 154, 171]. Choosing the best enhancements for a given problem is not a trivial task since each improves the performance in a different manner.

Our branch-and-Benders-cut algorithm employs the following algorithmic features:

a preprocessing routine to solve the linear relaxation, the generation of Pareto-optimal cuts, a core point selection criterion, lower bound strengthening via lift-and-project cuts, an in-tree matheuristic, and fine-tuning of cut parameters. In the following sections we explain each of the aforementioned enhancements.

### 3.4.1 Preprocessing

Since MP is a Benders reformulation of the original formulation P, by relaxing the integrality constraints and adding all Benders cuts to MP, we would obtain the LP relaxation solution of P. This is particularly important to note when implementing Benders decomposition in a single enumeration tree. One of the recent common practices is to solve MP as a linear program with a cutting plane algorithm and use the Benders cuts generated as part of the problem definition declared to the MIP solver. General-purpose solvers use this information to increase lower bounds, infer good branching rules, and fix variables in their preprocessing routine to reduce the underlying linear program's size.

In our algorithm, we solve MP as a linear program before declaring it within the MIP framework. However, instead of defining the problem with all Benders cuts generated so far, we only include the Benders cuts that are binding at the optimal LP solution as in Fischetti et al. [79] and in Bodur and Luedtke [33]. This guarantees that we obtain the LP optimal value before attempting to separate Benders cuts but pass on only the essential information to the general-purpose solver and avoid declaring an excessively large problem. This step, while obvious, significantly helps the solution process.

### 3.4.2 Pareto-optimal cut separation

Since the seminal paper on cut selection for Benders decomposition by Magnanti and Wong [127], Pareto-optimal cuts have become a standard practice. The approach ap-

plies to problems for which there is an infinite number of alternative optimal solutions to $\text{DSP}_k$ and therefore Benders optimality cuts. This is particularly the case in network design problems known for their primal degeneracy. For a minimization problem, the authors define cut dominance as follows. Given two cuts defined by dual solutions $u$ and $u^1$ of the form $z \geq f(u) + yg(u)$ and $z \geq f(u^1) + yg(u^1)$, respectively, the cut defined by $u$ dominates that defined by $u^1$ if and only if $f(u)+yg(u) \geq f(u^1)+yg(u^1)$ with strict inequality holding for some feasible $y$ of $MP$. If a cut defined by $u$ is not dominated by any other optimality cut, then this cut is said to be a Pareto-optimal Benders cut.

In general, to obtain Pareto-optimal Benders cuts an additional linear program must be solved at each iteration. This additional linear program is the same as the dual subproblem with two exceptions. The first is that a point $y^0$ in the relative interior of the master problem space, known as a core point, replaces the master problem solution $\bar{y}$ in the objective function (3.8). The second is that an equality constraint is added to ensure that the obtained solution belongs to the set of alternative optimal solutions of $\text{DSP}_k$ for the current master problem solution $\bar{y}$. In most cases, these modifications break the structure of the dual subproblem exploitable by an efficient combinatorial algorithm. This leads to having to solve an additional linear program. Papadakos [149] addresses this issue and presents a modified procedure that does not require solving an additional linear program. The modified dual subproblem uses a point that must satisfy characteristics that are more relaxed than Magnanti and Wong's conditions.

In our algorithm, we use the "tailored" subproblem for the MUFND as presented in Magnanti et al. [126]. The authors point out that the additional linear program for each commodity $k \in K$ in the classic Pareto-optimal approach is equivalent to solving the following *parametric minimum cost flow problem*:

$$(\text{MCF}_k) \quad \text{minimize} \sum_{(i,j)\in A} W^k c_{ij}^k x_{ij}^k - DSP_k(\bar{y})x_0 \tag{3.15}$$

$$\text{subject to} \quad \sum_{j\in N} x_{ji}^k - \sum_{j\in N} x_{ij}^k = \begin{cases} -(1+x_0) & \text{if } i = o_k \\ 1+x_0 & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N \tag{3.16}$$

$$x_{ij}^k \leq y_{ij}^0 + x_0 \bar{y}_{ij} \qquad \forall (i,j) \in A \tag{3.17}$$

$$x_{ij}^k \geq 0 \qquad \forall (i,j) \in A$$

$$x_0 \in \mathbb{R}.$$

The problem can be interpreted as that in which a rebate of $\text{DSP}_k(\bar{y})$ is given for each additional unit of the commodity routed on the network with demand and capacities defined by (3.16) and (3.17), respectively [126]. The authors show that any fixed value $x_0 \geq \sum_{(i,j)\in A} y_{ij}^0$ is optimal for $\text{MCF}_k$, leaving a minimum cost flow problem to be solved for each commodity $k \in K$.

As a result of fixing $x_0$, it is no longer necessary to solve $\text{DSP}_k(\bar{y})$ since it is now multiplied by a constant in $\text{MCF}_k$. This observation allows us to save computational time by solving $\text{MCF}_k$ directly as the separation problem rather than as a complementary problem for Pareto-optimal Benders cuts.

Magnanti and Wong [127] note that the selection of different core points $y^0$ leads to varied Pareto-optimal cuts. To the best of our knowledge, the question of selecting an adequate $y^0$ has not been addressed before in the literature. We provide some guidelines and computationally test different strategies for this in Section 3.6.2.

### 3.4.3 Benders lift-and-project cuts

With the current trend of implementing Benders decomposition within a branch-and-cut framework, the need for problem formulations to have a strong LP relaxation has become more important. Unfortunately, several problems do not satisfy this property. Recently, Bodur and Luedtke [33] and Bodur et al. [34] proposed the use of general mixed integer cuts such as mixed integer rounding and split cuts, respectively, within a branch-and-Benders-cut algorithm to improve the LP relaxation. Their results show a significant decrease in the LP gap leading to faster solution times.

Although Hellstrand et al. [102] show that the polytope defined by P is quasi-integral, the use of modified pivots for integral basic solutions is impractical due to P's degeneracy [15]. In addition, modified integral pivots require the complete formulation whereas our Benders reformulation is a projection into the smaller subspace of the integer variables. As a result, we must find a way to close the LP gap within the Benders decomposition framework. To do this, we adopt the lift-and-project cuts proposed by Balas et al. [17] to strengthen the master problem LP relaxation.

Lift-and-project cuts, a result of disjunctive programming theory [14], were initially proposed as a cutting plane algorithm by Balas et al. [17] but were later extended to the branch-and-cut framework [18] by proving the ability to find globally valid cuts at nodes within the enumeration tree by means of a closed form lifting procedure. The framework is as follows.

Given $P = \{x \in \mathbb{R}^n | \tilde{A}x \geq \tilde{b}\}$ with inequalities of the form $1 \geq x \geq 0$ included in $\tilde{A}x \geq \tilde{b}$ and $P_D = conv\{x \in P | x_j \in \{0,1\}, \forall j = 1...p\}$ where $p < n$, lift-and-project cuts can be obtained by:

1 Selecting an index $\hat{\jmath} \in \{1...p\}$. Multiplying $\tilde{A}x \geq \tilde{b}$ by $(1 - x_{\hat{\jmath}})$ and $x_{\hat{\jmath}}$.

2 Linearizing the obtained system by substituting $y_i = x_{\hat{\jmath}} x_i$ and $x_i = x_i x_i$.

3 Projecting the system back into the original space by means of a cone projection.

Balas [14] shows that it is possible to obtain the "deepest" lift-and-project cut of the form $\sum_{i \in I} \alpha_i x_i \geq \beta$ that cuts off the LP optimum $\bar{x}$ for a given $\hat{\jmath} \in \{1...p\}$ by solving the following linear program:

$$
\begin{aligned}
(\text{CGLP}_{\hat{\jmath}}) \quad & \text{minimize} \sum_{i \in I} \alpha_i \bar{x}_i - \beta \\
\text{subject to} \quad & \alpha - uA + u^0 e_{\hat{\jmath}} \geq 0 \\
& \alpha - vA - v^0 e_{\hat{\jmath}} \geq 0 \\
& -\beta + ub = 0 \\
& -\beta + vb + v_0 = 0 \\
& u, v \geq 0,
\end{aligned}
$$

where $e_{\hat{\jmath}}$ is the vector of all 0s except for a 1 in the $\hat{\jmath}$-th component.

The feasible space of $\text{CGLP}_{\hat{\jmath}}$ is a convex cone. Therefore, a normalization constraint must be added to ensure a finite optimal solution. It has been shown [16] that varied normalizations lead to significantly different cuts. In our implementation, we use the following normalization constraint:

$$\sum_i u_i + u^0 + \sum_i v_i + v^0 = 1.$$

The resulting cut can be strengthened by using a closed formula derived by imposing integrality constraints of other $\{0, 1\}$ variables [16]. We adopt this strengthening procedure in our algorithm. For a given variable $x_k$, its coefficient $\alpha_k$ in the lift-and-project cut can be replaced by $\alpha'_k = \min\{ua_k + u^0 \lceil m_k \rceil, va_k - v^0 \lfloor m_k \rfloor\}$, where

$$m_k = \frac{va_k - ua_k}{u^0 + v^0}.$$

While solving our Benders reformulation we do not have the complete polyhedral

description of P since there would be exponentially many constraints. We resort to defining $\tilde{A}$ as the feasibility and optimality cuts that are binding at the LP relaxation and the $1 \geq x \geq 0$ constraints. Although this is known to give a weaker lift-and-project cut, we also highlight its role in significantly reducing the size of $CGLP_{\hat{\jmath}}$ leading to times of less than 0.02 seconds to obtain a cut. A similar strategy is used by Balas and Perregaard [16] outside the context of Benders decomposition. Another important factor in the lift-and-project process is selecting the index $\hat{\jmath}$. In our implementation we choose the fractional variable of the LP solution with highest fixed cost.

During our experiments we noted that these lift-and-project cuts are very dense and often with small, numerically unstable coefficients. This led to numerical issues when too many were added to the master problem at the same time. To circumvent this, we ensure that at most seven lift-and-project cuts are added. We use an additional stopping criterion of increase in the LP optimal value. In other words, if the effect of adding a Benders lift-and-project cut is negligible, we then stop generating them. Note that the effect of the lift-and-project cut is highly dependent on the variable $\hat{\jmath}$ chosen. However, we found this criterion to be an effective rule to prevent numerical instabilities in our algorithm.

### 3.4.4 An in-tree matheuristic

An important factor in solving difficult optimization problems, in particular when using branch-and-bound methods, is obtaining high quality feasible solutions. Finding these early in the enumeration process often leads to smaller search trees since they provide better bounds for pruning and a guide for selecting variables to branch on. If found in a preprocessing stage, they can be used to perform variable elimination tests as in Contreras et al. [48, 49]. Preliminary tests showed that the latter approach eliminated few variables from the problem even if the optimal solution value was used

for these variable elimination tests. We therefore propose an in-tree matheuristic that exploits the information generated during the enumeration process. Our algorithm uses the paths obtained while solving the Benders subproblems as variables in a path-based formulation of the MUFND.

Let $\Theta_k^\mu$ denote a binary variable whose value is 1 if path $\mu$ is used for commodity $k$ and 0 otherwise, while $y_{ij}$ denotes the network design variables as in P. Define parameter $v_k^\mu(i,j) = 1$ if arc $(i,j)$ belongs to path $\mu$ for commodity $k$, and 0 otherwise. Finally, let $\Omega_k$ denote the set of paths from $o(k)$ to $d(k)$ and $\Omega$ represent the union of these sets over $K$. With this notation we have the following path-based formulation for the MUFND:

$$(P_{Heur}) \quad \text{minimize} \sum_{(i,j)\in A} f_{ij}y_{ij} + \sum_{k\in K}\sum_{\mu\in\Omega_k}[W^k \sum_{(i,j)\in A} c_{ij}^k v_k^\mu(i,j)]\Theta_k^\mu \tag{3.18}$$

$$\text{subject to} \quad \sum_{\mu\in\Omega_k} \Theta_k^\mu = 1 \qquad \forall k \in K \tag{3.19}$$

$$\sum_{\mu\in\Omega_k} v_k^\mu(i,j)\Theta_k^\mu \leq y_{ij} \qquad \forall (i,j) \in A, k \in K \tag{3.20}$$

$$\Theta_k^\mu \in \{0,1\} \qquad \forall k \in K, \mu \in \Omega_k. \tag{3.21}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{3.22}$$

The objective function (3.18) represents the total cost of routing the commodities on the network, including design and transportation costs. Constraints (3.19) ensure that each commodity is routed by exactly one path while (3.20) force the design variables of the arcs used in a path to take value 1 if the path is used to route a commodity. We note that (3.20) can be disaggregated. However, preliminary tests showed it to have a negative effect on the computation time.

Given the exponential number of paths for a given commodity, $P_{Heur}$ is usually solved using column generation and branch-and-price when used to represent the

MUFND. In our algorithm, however, we will only use this formulation to solve for improved MUFND solutions obtained from paths generated while solving the Benders subproblems. This avoids having to solve several rounds of pricing problems at each branch-and-price node. The only added computational effort comes from the fact that primal solutions to $\text{MCF}_k$, the Benders subproblem to obtain Pareto optimal cuts, sends different amounts of flow through several paths from origin to destination. To obtain single paths to be used in $\text{P}_{Heur}$, we solve a shortest path problem over two networks derived from the primal solution of $\text{MCF}_k$. The first network contains the arcs that send any flow greater than 0.1 in the solution while the second contains arcs that send more than 1 unit of flow. This provides us with the potential to generate two different paths at a low computational cost every time the Benders subproblem is solved.

Finally, note that the branching over design variables during the enumeration process of our Benders algorithm forces the generation of a varied set of paths for the Benders subproblems and hence variables for $\text{P}_{Heur}$. The integration of our in-tree matheuristic into our branch-and-Benders-cut algorithm provides a means of exploiting two formulations of the same problem simultaneously as in Hewitt et al. [103] for capacitated multicommodity network design with the difference that our algorithm solves the problem to optimality whereas theirs finds feasible solutions with a quality certificate.

### 3.4.5 Implementation details

We begin our solution process by solving the LP relaxation of MP using our cutset separating routines and $\text{MCF}_k$ as our separation oracle to obtain Pareto-optimal Benders cuts with core points defined as will be described in Section 3.6.2. Upon confirming that no more violated Benders cuts exist, we use those that are binding at the optimal solution to obtain a violated Benders lift-and-project cut. This cut

is added to the MP relaxation and we resume separating violated Benders cuts. We repeat this process until our stopping criteria are satisfied.

We then define the MIP problem in CPLEX with the active Benders and lift-and-project cuts as lazy and user constraints. This prevents defining an excessively large initial problem. Another of the important aspects to consider when implementing this method is the separation and cut adding frequency. Adding too few cuts leads to an underestimation of the lower bounds of nodes in the enumeration tree, while adding too many cuts leads to large LPs that require a longer computation time to solve.

Several cutting frequencies were tested in preliminary experiments. The best of the tested strategies was separating Benders cuts at all nodes in the first five levels of the enumeration tree and then separating at every 100th node. For all these, only one round of violated Benders cuts are added. For fractional solutions, a minimum violation of 0.01 was required to add the cut to the constraint pool of the node's linear problem.

Lastly, to prevent executing our in-tree matheuristic too frequently, we limit its use to only one node at each depth of the enumeration tree greater than ten that is divisible by five. In addition, to ensure it has the potential to find an improved solution, it is only called if at least $N$ new paths have been generated since its last execution. Finally, to avoid spending excessive time in this heuristic process, a time limit of thirty seconds was set for each execution.

## 3.5 A cut-and-solve algorithm for the MUFND

Introduced by Climer and Zhang [45] in the artificial intelligence community, cut-and-solve has been used to solve well-known combinatorial optimization problems such as the *travelling salesman problem* and the *single-source capacitated facility location*

*problem* [89, 177]. The cut-and-solve framework is closely related to local branching [76] in the sense that at each level of the enumeration tree only two child nodes exist, one corresponding to a smaller "sparse" problem and the other as its complement known as the "dense" problem. However, while in local branching one begins with a feasible solution and defines the subproblems based on the Hamming distance, cut-and-solve allows for more generic problem definitions and does not require an initial feasible solution. Since our proposed framework is more closely aligned with the latter, we adopt the cut-and-solve terminology and notation for the rest of the paper. We next provide a brief description of the cut-and-solve procedure as presented by Climer and Zhang [45].

The "sparse" and "dense" problems are defined by constraints over a set of variables. These constraints, known as "piercing" cuts, are of the form $\sum_{i \in I} x_i \leq \sigma$ and $\sum_{i \in I} x_i \geq \sigma + 1$ where $I \subset N$ is a subset of the problem's binary variables and $\sigma \in \mathbb{Z}$.

Upon branching, the "sparse" problem is solved to optimality by means of branch-and-bound or any exact method to obtain a primal bound ($UB_{sparse}$) on the original problem. This highlights the need to define sparse problems that are easily solved. Next, the linear relaxation of the dense problem is solved to obtain a lower bound ($LB_{dense}$) on the remaining solution space of the original problem. If $LB_{dense} \geq UB_{sparse}$ then $UB_{sparse}$ is optimal for the complete problem. Otherwise, another piercing cut is defined over the dense problem and the procedure is repeated.

We propose the use of our branch-and-Benders-cut algorithm as the black box MIP solver within the cut-and-solve algorithm and a tailored rule for selecting the variables to consider in the "sparse" problems. Two important advantages of using our Benders algorithm within the cut-and-solve framework are the reduced problem size and the re-usability of the Benders cuts generated in previous sparse problems. On the other hand, some advantages to using cut-and-solve over Benders is that piercing cuts significantly reduce the solution space and the optimal values of previous sparse

problems are useful for pruning branches in the enumeration tree.

Our cut-and-solve algorithm begins by considering the union of shortest paths, denoted as $\cup_{k\in K} P_k$, of each commodity using only their transportation costs. The resulting set contains on average approximately 90% of the arcs open in an optimal solution and is thus an ideal candidate to define our first "sparse" problem.

For ease of exposition we introduce the following notation. Let $(\bar{y}, \bar{z})(t)$ represent the solution of the $t$-th sparse problem, $I(t)$ denote the set of indices of arc variables whose value is 1 in $(\bar{y}, \bar{z})(t)$ and $\chi(\bar{y}, \bar{z})(t))$ be its objective function value. In particular, $(\bar{y}, \bar{z})(0)$ refers to the objective function value of activating and routing on the arcs of the union of shortest paths.

At a given $t \geq 1$, we define the following sparse problem:

$$(\text{MP}_{sparse}(t)) \quad \text{minimize} \sum_{(i,j)\in A} f_{ij} y_{ij} + \sum_{k\in K} z_k \tag{3.23}$$

$$\text{subject to} \quad z_k \geq \lambda^k_{d_k} - \lambda^k_{o_k} - \sum_{(i,j)\in A} \mu^k_{ij} y_{ij} \qquad \forall (\lambda, \mu)_k \in Opt(DSP_k), k \in K \tag{3.24}$$

$$\sum_{(i,j)\in\delta(S)} y_{ij} \geq 1 \qquad \forall S \in \Delta_K \tag{3.25}$$

$$z \in \mathbb{R}^{|K|} \tag{3.26}$$

$$y \in \{0,1\}^{|A|} \tag{3.27}$$

$$\sum_{(i,j)\notin I(t-1)} y_{ij} \leq t \tag{3.28}$$

$$\sum_{(i,j)\notin I(s)} y_{ij} \geq s+2 \qquad \forall s = 0, ..., t-1 \tag{3.29}$$

$$\sum_{(i,j)\in A} f_{ij} y_{ij} + \sum_{k\in K} z_k \leq \chi((\bar{y}, \bar{z})(t-1)), \tag{3.30}$$

where $\Delta_k = \{S \subset N | \exists k \in K \text{ where } o_k \in S, d_k \notin S\}$. The constraints (3.23)-(3.27) are the Benders master problem reformulation of the MUFND. Constraint (3.28)

is the piercing cut that allows at most $t$ variables not in the previous solution to take the value of 1 while constraints (3.29) are the negations of (3.28) from previous iterations. The latter ensure that previously searched areas of the feasible space are not considered in the new sparse problem. Finally, constraint (3.30) imposes that the optimal solution of the current sparse problem has objective value of at most the optimal value of the previous one. This constraint ensures that the obtained solutions do not worsen after each iteration and saves computation time since its value is used as a pruning criterion for the enumeration tree.

If $\text{MP}_{sparse}(t)$ is feasible, we define $\text{MP}_{sparse}(t+1)$ without solving the LP relaxation of the corresponding dense problem. In this respect, our algorithm bears resemblance to local branching as proposed in Fischetti and Lodi [76]. This is done until two successive optimal solutions to $\text{MP}_{sparse}(t)$ are the same or until $\text{MP}_{sparse}(t)$ is infeasible, due to (3.30). If either occurs, the dense problem, $\text{MP}_{dense}$, is defined. The dense problem is similar to $\text{MP}_{sparse}(t)$ with the exception that (3.28) is replaced by $\sum_{(i,j)\notin I(t-1)} y_{ij} \geq t+1$ and $(\bar{y}, \bar{z})(t-1)$ in (3.30) is replaced by the best solution found so far. We then solve $\text{MP}_{dense}(t)$ which will either be feasible and hence provide the true optimal solution or will be infeasible meaning that the best solution found so far is indeed optimal.

The presented framework provides a novel research direction, different from Rei et al. [158], for combining cut-and-solve and Benders decomposition. Below we present the pseudocode of our cut-and-solve Benders algorithm.

**Algorithm 2** Cut-and-solve Benders algorithm for the MUFND

**Require: 0**: Initialization
  $(\bar{y}, \bar{z})(0) = \cup_{k \in K} P_k$, $t = 1$, $best = (\bar{y}, \bar{z})(0)$
  **Step 1**: Define and solve $\text{MP}_{sparse}(t)$
  **if** ($\text{MP}_{sparse}(t)$ is feasible $\wedge$ $(\bar{y}, \bar{z})(t-1) \neq (\bar{y}, \bar{z})(t)$) **then**
    $best = (\bar{y}, \bar{z})(t)$
    $t = t + 1$;
    `Goto` **Step 1**
  **else**
    `Goto` **Step 2**
  **end if**
  **Step 2**: Define and solve $\text{MP}_{dense}$
  **if** $MP_{dense}$ is feasible **then**
    Update *best*
  **end if**
  Return *best*.

## 3.6 Computational experiments

We perform extensive computational experiments to evaluate the efficiency of our proposed methods and the effect of the enhancements implemented. Our analyses focus on: the LP gap closed by adding Benders lift-and-project cuts to MP's linear relaxation, adequate core point selection, and the efficiency of our proposed solution methods versus the state-of-the-art general-purpose MIP solver CPLEX 12.7.1.

We use the well-known "Canad" multicommodity capacitated network design instances [62] as our testbed. This dataset consists of 205 instances with arc capacities. Ignoring the capacity constraints leaves a total of 93 distinct instances for our experiments. The testbed can be divided into three classes. The first are the 31 "C" instances with many commodities compared to nodes while the second are eight "C+" instances with few commodities compared to nodes. Finally, Class III is divided into two subgroups. Class III-A and III-B are each comprised of 27 "R" instances on small and medium sized graphs, respectively.

We generate eight large-scale instances, denoted as Class IV, on which we test

our algorithms with a 24-hour time limit. These were generated using the Mulgen generator [62] available at *http://pages.di.unipi.it/frangio/* with sizes of up to 1,500 arcs and 1,500 commodities. To the best of our knowledge these are the largest instances of the MUNDP to be solved by an exact algorithm.

Another characteristic of our testbed is the existence of instances that have an LP gap strictly greater than 0. This is important in our analysis as we need to test our algorithm's ability to quickly explore the enumeration tree and the efficiency of our proposed lift-and-project Benders cuts. The "Canad" testbed contains several instances with this property. Table 3.1 details the number of instances for each LP gap range (%) in our testbed.

Table 3.1: Distribution of "Canad" instances' LP gaps (%)

| **Class** | **0** | $(0,1]$ | $(1,2]$ | $(2,3]$ | $(3,4]$ | $[4,7.2]$ | **Total** |
|---|---|---|---|---|---|---|---|
| Class I | 10 | 7 | 2 | 4 | 2 | 6 | 31 |
| Class II | 7 | 1 | | | | | 8 |
| Class III-A | 26 | | 1 | | | | 27 |
| Class III-B | 9 | 2 | | 2 | 5 | 9 | 27 |
| Class IV | 0 | 3 | 3 | 2 | | | 8 |
| **Total** | **52** | **13** | **6** | **8** | **7** | **15** | **101** |

All algorithms were coded in C using the callable library for CPLEX 12.7.1. The separation and addition of cutset inequalities and Benders optimality cuts is implemented via lazy cut callbacks and user cut callbacks. For a fair comparison, all use of CPLEX was limited to one thread and the traditional MIP search strategy. Experiments were executed on an Intel Xeon E5 2687W V3 processor at 3.10 GHz under Linux environment.

### 3.6.1 Impact of lift-and-project cuts on LP gap

As shown in Table 3.1, Class I, III-B, and IV contain most of the instances with higher LP gap. Preliminary tests showed these to be the most difficult to solve, in

particular when it came to proving optimality. It is in this spirit that we proposed using lift-and-project cuts to improve the LP bound at the root node.

Table 3.2 shows the percentage of the LP gap ($LP_{imp}\%$) closed by applying at most seven lift-and-project cuts to the linear relaxation of our Benders master problem. This percentage is calculated as $LP_{imp} = 100 \times \frac{(LP_{MPLP} - LP_{MP})}{Opt - LP_{MP}}$ where $LP_{MPLP}$ is the optimal value of the linear relaxation of the master problem with the additional lift-and-project cuts, $LP_{MP}$ is the optimal value of the linear relaxation of the master problem, and $Opt$ is the optimal value of the problem.

Table 3.2: LP gap (%) closed

| **Class** | **No. of instances** | $LP_{imp}$ |
|---|---|---|
| Class I | 21 | 6.61 |
| Class II | 1 | 9.07 |
| Class III-A | 1 | 20.45 |
| Class III-B | 18 | 3.65 |
| Class IV | 8 | 2.99 |
| **Total** | **49** | **5.26** |

The average improvement of the LP gap is of 5.26% over the 50 solved instances with an LP gap. There are many factors that contribute to this behavior. The first is that lift-and-project cuts as proposed by Balas et al. [17] require the complete formulation of the problem. In our implementation, we use a relaxation comprised of only the Benders cuts that are binding at the LP solution. In Balas et al. [18], this relaxation is shown to generate weaker cuts. Second, our stopping criterion for lift-and-project cuts is conservative, avoiding generating too many of them at the beginning due to their numerical instability. Finally, our simplified variable selection rule also contributes to this performance.

### 3.6.2 Impact of core point selection

Despite the use of Pareto optimal Benders cuts being now common practice, little computational experimentation with core point selection strategies has been done. We next show how core point selection influences the solution time and that a tailored core point selection strategy can lead to significant time savings. We test three strategies for core point selection. The first is the most common practice in the literature while the second is a novel strategy that can be applied to any Benders reformulation of a mixed binary program. Finally, the third is a strategy tailor-made for the MUFND and is based on the union of shortest paths of the commodities $k \in K$, as in the definition of our initial "sparse" problem. Let $y^0$ and $\bar{y}$ denote the current core point and master problem solution, respectively. The details of the three core point selection strategies are as follows:

1 Initialize $y^0$=$\{1\}^{|A|}$ and dynamically update the core point as $y^0 = 0.5y^0 + 0.5\bar{y}$ as in Papadakos [149] and similar to Fischetti et al. [79].

2 Initialize a stabilizer point $\hat{y}$ as $\hat{y} = \{1\}^{|A|}$ which will then be updated as better incumbent solutions are found during the enumeration process. Dynamically update the core point as $y^0 = 0.5\hat{y} + 0.5\bar{y}$.

3 Fix the core point throughout the entire process based on the arcs that are present in at least one of the commodities' shortest paths, denoted as $\cup_{k\in K} P_k$. The fixed core point is defined as $y^0_{ij} = 0.7$ if $(i,j) \in \cup_{k\in K} P_k$ and $y^0_{ij} = 0.2$ if $(i,j) \notin \cup_{k\in K} P_k$. These values were chosen after running preliminary experiments with the values of $\{0.5, 0.6, 0.7, 0.8, 0.9\}$ for arcs in the routing solution and $\{0.1, 0.2, 0.3, 0.4, 0.5\}$ for arcs not in the routing solution.

Note that in all three cases, the proposed core point is in the interior of the $\{0,1\}^{|A|}$ hypercube. However, to solve MCF$_k$, $y^0$ must not only lie in the interior

of the $\{0,1\}^{|A|}$ hypercube but must also define a network through which one unit of demand can be sent from $o_k$ to $d_k$, $\forall k \in K$. Failure to do so could lead to $\text{MCF}_k$ being infeasible despite $\bar{y}$ being a feasible solution for the MUFND. This was observed empirically to have a particularly pernicious effect on the overall computation time.

To remedy this we solve a minimum cut for each $k \in K$ to check for feasibility when defining the fixed core point of strategy 3. If there exists a minimum cut $\delta(S)_k = \{(i,j) \in A | o_k \in S \text{ and } d_k \in N \backslash S\}$ for a commodity $k \in K$ with $\sum_{(i,j)\in\delta(S)_k} y^0_{ij} < 1$, we then increase the value of each arc in $\delta(S)_k$ by $[1/|\delta(S)_k|]+0.01$ and check again for a violated cutset. This is repeated until no such cutset exists. We place a cap on the value of $y^0_{ij}$ to be at most 0.9999 to ensure $y^0$ remains an interior point. Given that the core point is fixed throughout the solution process, this verification is only done once at the beginning. Since the other strategies constantly update the core point, running this procedure every time proved to be time consuming. To circumvent this we run this procedure only when $\text{MCF}_k$ becomes infeasible. The solution times, in seconds, of these strategies implemented in our branch-and-Benders-cut algorithm are shown in Table 3.3.

Table 3.3: Impact of core point selection- time in seconds

| **Class** | **Nb** | **Strategy 1** | **Strategy 2** | **Strategy 3** |
|---|---|---|---|---|
| Class I | 31 | 119.46 | 286.42 | 73.74 |
| Class II | 8 | 12.96 | 6.50 | 6.91 |
| Class III-A | 27 | 0.03 | 0.04 | 0.04 |
| Class III-B | 27 | 227.52 | 451.77 | 187.15 |
| Class IV | 6 | 112.17 | 204.02 | 67.25 |
| **Total** | **99** | **107.31** | **225.80** | **78.78** |

The best performing is our tailored core point selection strategy (strategy 3) which saves over a quarter of the average computation time of the second best performing strategy, the well-known dynamic mid-point update (strategy 1). The worst is the incumbent stabilizer update (strategy 2). These results show the added value of using core point selection strategies that exploit problem structure.

### 3.6.3 Computation time

We now compare the computation time of each of our proposed algorithms. We begin by focusing on our branch-and-Benders-cut algorithm since we use the best performing as the black box solver in our cut-and-solve/local branching algorithm. To show the impact of each enhancement, we present four versions of our branch-and-Benders-cut algorithm. The first is without using our in-tree matheuristic nor our lift-and-project cuts ($B_0$). The second is the same, with the addition of the in-tree heuristic ($B_1$). $B_2$ is the branch-and-Benders-cut algorithm with lift-and-project cuts added at the root node and the final version ($B_3$) combines them all. A time limit of 24 hours is set for all algorithms.

The results are presented in Table 3.4 with the exception of the instances of classes II and III-A which were all solved in less than a second by our four algorithms and CPLEX. The first three columns describe the problem class, instance sizes ($|N|, |A|, |K|$), and number of instances in each instance group respectively. For each version of the algorithm, two columns are displayed,"Seconds" which denotes the average solution time in seconds and "Nodes" which refers to the average number of nodes explored. Finally, we point out that the averages of class IV and the total test bed are taken only over the instances with comparable solution times to avoid the averages being skewed by large numbers, i.e., instance groups 50,1500,1000 and 50,1500,1500 are omitted.

Table 3.4: Computational performance of branch-and-Benders-cut algorithm

| | | | $B_0$ | | $B_1$ | | $B_2$ | | $B_3$ | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Class** | $(\|N\|,\|A\|,\|K\|)$ | **Nb** | **Seconds** | **Nodes** | **Seconds** | **Nodes** | **Seconds** | **Nodes** | **Seconds** | **Nodes** |
| | 20,230,40 | 3 | 0.22 | 0 | 0.21 | 0 | 0.21 | 0 | 0.21 | 0 |
| | 20,230,200 | 4 | 15.17 | 274.50 | 15.65 | 271.00 | 21.58 | 395.00 | 26.15 | 496.00 |
| | 20,300,40 | 4 | 0.24 | 0.25 | 0.24 | 0.25 | 0.28 | 0.75 | 0.28 | 0.75 |
| | 20,300,200 | 4 | 12.11 | 200.50 | 12.80 | 159.00 | 25.94 | 371.50 | 34.67 | 384.25 |
| **I** | 30,520,100 | 4 | 305.08 | 3,335.25 | 157.50 | 2,456.50 | 286.05 | 2,694.75 | 233.35 | 4,439.25 |
| | 30,520,400 | 4 | 9.14 | 35.25 | 9.06 | 35.25 | 12.21 | 36.75 | 12.34 | 36.25 |
| | 30,700,100 | 4 | 8.86 | 188.25 | 7.92 | 162.00 | 9.42 | 209.25 | 8.55 | 256.25 |
| | 30,700,400 | 4 | 332.35 | 4,613.25 | 293.22 | 5,322.25 | 254.17 | 3,800.50 | 276.19 | 3,716.75 |
| | **Sub-Total** | **31** | **88.14** | **1,115.77** | **64.07** | **1,084.68** | **78.68** | **968.84** | **76.35** | **1,203.81** |
| | 20,120,40 | 3 | 0.11 | 0 | 0.10 | 0 | 0.10 | 0 | 0.11 | 0 |
| | 20,120,100 | 3 | 1.97 | 70.67 | 1.98 | 70.67 | 3.42 | 88.67 | 3.50 | 93.00 |
| | 20,120,200 | 3 | 214.59 | 1,808.00 | 255.64 | 1,921.00 | 166.83 | 1,062.33 | 178.16 | 1,521.67 |
| | 20,220,40 | 3 | 2.00 | 100.67 | 1.95 | 99.00 | 2.08 | 111.67 | 2.34 | 99.33 |
| **III-B** | 20,220,100 | 3 | 39.81 | 688.67 | 49.54 | 839.33 | 177.31 | 2,548.33 | 124.55 | 2,232.33 |
| | 20,220,200 | 3 | 775.17 | 3,644.67 | 929.44 | 3,285.33 | 1,320.94 | 3,583.33 | 1,460.67 | 3,738.00 |
| | 20,320,40 | 3 | 11.44 | 876.67 | 15.14 | 849.67 | 12.47 | 872.00 | 12.69 | 832.67 |
| | 20,320,100 | 3 | 6.87 | 104.33 | 7.17 | 104.33 | 12.50 | 223.67 | 11.81 | 199.33 |
| | 20,320,200 | 3 | 625.82 | 1,723.00 | 669.79 | 2,273.67 | 1,012.85 | 2,476.67 | 1,087.30 | 3,176.00 |
| | **Sub-Total** | **27** | **186.42** | **1,001.85** | **214.53** | **1,049.22** | **300.94** | **1,218.52** | **320.13** | **1,321.37** |
| | 40,1200,400 | 1 | 9.19 | 6.00 | 8.97 | 6.00 | 15.56 | 7.00 | 15.68 | 7.00 |
| | 40,1200,800 | 1 | 53.00 | 537.00 | 55.98 | 669.00 | 68.69 | 715.00 | 63.87 | 711.00 |
| **IV** | 40,1200,1200 | 1 | 57.07 | 61.00 | 57.34 | 61.00 | 82.95 | 74.00 | 86.69 | 74.00 |
| | 50,1400,400 | 1 | 29.94 | 649.00 | 31.70 | 580.00 | 34.82 | 601.00 | 36.54 | 597.00 |
| | 50,1400,800 | 1 | 117.31 | 1,938.00 | 127.06 | 1,886.00 | 177.01 | 3,860.00 | 196.47 | 3,215.00 |
| | 50,1400,1200 | 1 | 201.33 | 2,162.00 | 198.49 | 2,257.00 | 183.17 | 1,655.00 | 208.59 | 1,929.00 |
| | 50,1500,1000 | 1 | 54,074.52 | 419,563.00 | 54,680.74 | 420,298.00 | 44,367.11 | 389,246.00 | 57,487.39 | 555,136.00 |
| | 50,1500,1500 | 1 | 69,637.47 | 262,007.00 | 69,055.11 | 262,007.00 | time | 483,149.00 | time | 438,179.00 |
| | **Sub-Total** | **6** | **77.97** | **892.17** | **79.92** | **909.83** | **93.70** | **1,152.00** | **101.31** | **1,088.83** |
| | **Total** | **64** | **128.65** | **1,046.75** | **129.03** | **1,053.33** | **173.86** | **1,091.34** | **181.53** | **1,242.63** |

We note that with respect to computation time, implementing Benders without including Benders lift-and-project cuts performs on average the fastest. Between the two versions that exclude it, $B_0$ is on average marginally better than $B_1$ over the instances of class III-B and IV but 30% slower on average over the instances of class I. This shows that although in most cases using the in-tree heuristic will increase the solution time, there are instances for which it pays off significantly, for example instance groups 30,520,100 and 30,700,400 of class I.

While incorporating Benders lift-and-project cuts has a better solution time when compared to $B_0$ for the instances in class I, it produces on average over a 30% increase

in solution time over the complete testbed. $B_2$ provides the fastest solution time for instance groups 30,700,400 of class I; 20,120,200 of class III-B; and 50,1500,1000 of class IV; however, for other instance groups, it can double the solution time despite exploring fewer nodes (see instance group 20,220,200 of class III-B). An explanation for this is the density and numerical instability of the Benders lift-and-project cuts. As mentioned before, these cuts have several non-zero coefficients close to zero. This leads to more time required to solve the underlying linear programs and in some instances numerical instability that prevents CPLEX from constructing an advanced basis for nodes in the tree. We also note that their use rarely leads to a reduction in the size of the enumeration tree as both $B_3$ and $B_4$ show a larger average number of nodes explored than $B_0$ and $B_1$. This indicates that the addition of these cuts negatively influences the branching within the enumeration tree.

Incorporating both Benders lift-and-project cuts and our in-tree heuristic simultaneously is the version that requires the most computation time on average over the entire test bed. In fact, its solution time is worse than the versions that incorporate them individually. One of the main factors contributing to this is the increase that comes from incorporating Benders lift-and-project cuts which as we have seen, also negatively influences the branching within the enumeration tree. The rest of the solution time increase can be explained by the additional time the in-tree heuristic required to solve $\mathrm{P}_{Heur}$.

Finally, we note that both $B_0$ and $B_1$ are able to solve the two largest instances that required significantly more time. In particular, incorporating our in-tree matheuristic proved marginally beneficial for the largest instance. On the other hand, including Benders lift-and-project cuts rendered a time saving of one-fifth of the computation time required by $B_0$ to solve the second largest instance. This again shows the unpredictable effect of lift-and-project cuts in our branch-and-Benders-cut algorithm.

Considering these results, we choose $B_0$ as the black box solver for our cut-and-solve algorithm. Figure 3.1 is the performance profile of our branch-and-Benders-cut algorithm ($BB\&C$), our cut-and-solve ($CS/LB$) and solving $P$ with CPLEX 12.7.1's branch-and-cut algorithm $CPX$. We do not compare with CPLEX's black box Benders implementation since preliminary results showed it to perform significantly worse than CPLEX's branch-and-cut. Figure 3.1 plots the number of instances solved by each algorithm within a given number of seconds.

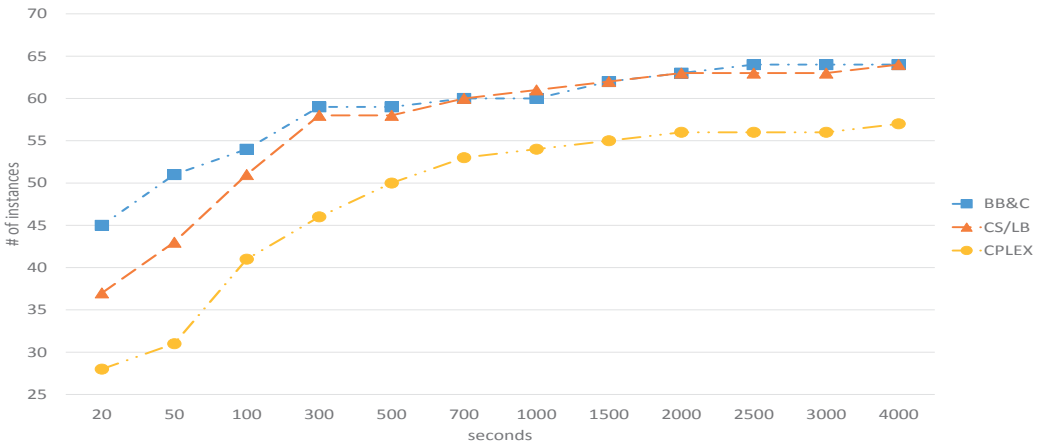



Figure 3.1: Number of instances solved in a given time limit

We note that both our proposed methodologies are able to solve more instances in less time than CPLEX, having solved close to 83% of the instances in less than 100 seconds and solving all except the two largest instances within 45 minutes. CPLEX on the other hand manages to solve only 41 instances within 100 seconds; less than the number solved by our branch-and-Benders-cut algorithm in 20 seconds. CPLEX runs into trouble proving optimality for 10 instances requiring over an hour for the least troublesome and over half a day for the most burdensome. In addition, it is unable to solve the two largest instances within the one-day time limit while our branch-and-Benders-cut algorithm solves all instances within twenty hours of CPU time.

When comparing the performance between ($CS/LB$) and ($BB\&C$), we see from

Figure 3.1 that the latter is faster at solving instances. However, when given additional time, both algorithms have similar behavior. To make a more precise comparison, Table 3.5 contains the average times, in seconds, required for each class and description of our testbed.

Our results show that on average both $BB\&C$ and $CS/LB$ are an order of magnitude faster than CPLEX for instances all three are able to solve. This speedup is even more significant when limiting our analysis to the large-scale instances. For these, our branch-and-Benders-cut algorithm is up to three orders of magnitude faster than CPLEX. The instances of Class III-B also show a significant saving in computation time in favor of our Benders decomposition-based algorithms. The savings obtained with $BB\&C$ can be largely attributed to solving smaller underlying linear programs in the enumeration tree and exploring the nodes in significantly less time.

Table 3.5: Comparison of computation times in seconds

| Class | $(\|N\|, \|A\|, \|K\|)$ | Nb | *CPX* | *CS/LB* | *BB&C* |
|---|---|---|---|---|---|
| I | 20,230,40 | 3 | 0.07 | 0.21 | 0.22 |
| | 20,230,200 | 4 | 252.95 | 34.56 | 15.17 |
| | 20,300,40 | 4 | 0.17 | 0.33 | 0.24 |
| | 20,300,200 | 4 | 303.24 | 30.35 | 12.11 |
| | 30,520,100 | 4 | 3,181.33 | 172.15 | 305.08 |
| | 30,520,400 | 4 | 95.46 | 20.61 | 9.14 |
| | 30,700,100 | 4 | 71.61 | 19.56 | 8.86 |
| | 30,700,400 | 4 | 10,479.58 | 550.03 | 332.35 |
| | **Sub-Total** | **31** | **1,856.05** | **106.80** | **88.14** |
| III-B | 20,120,40 | 3 | 0.05 | 0.10 | 0.11 |
| | 20,120,100 | 3 | 13.42 | 5.56 | 1.97 |
| | 20,120,200 | 3 | 361.23 | 245.47 | 214.59 |
| | 20,220,40 | 3 | 6.91 | 6.81 | 2.00 |
| | 20,220,100 | 3 | 153.86 | 54.32 | 39.81 |
| | 20,220,200 | 3 | 1,615.31 | 1,396.17 | 775.17 |
| | 20,320,40 | 3 | 27.79 | 36.06 | 11.44 |
| | 20,320,100 | 3 | 69.25 | 29.36 | 6.87 |
| | 20,320,200 | 3 | 2,592.58 | 435.30 | 625.82 |
| | **Sub-Total** | **27** | **537.82** | **245.46** | **186.42** |
| IV | 40,1200,400 | 1 | 59.82 | 19.95 | 9.19 |
| | 40,1200,800 | 1 | 4,483.75 | 141.31 | 53.00 |
| | 40,1200,1200 | 1 | 1,664.10 | 110.68 | 57.07 |
| | 50,1400,400 | 1 | 575.91 | 68.06 | 29.94 |
| | 50,1400,800 | 1 | 39,051.12 | 296.61 | 117.31 |
| | 50,1400,1200 | 1 | 57071.73 | 500.18 | 201.33 |
| | 50,1500,1000 | 1 | time | time | 54,074.52 |
| | 50,1500,1500 | 1 | time | time | 69,637.47 |
| | **Sub-Total** | **6** | **17,151.07** | **189.47** | **77.97** |
| | **Total** | **64** | **2,733.83** | **173.05** | **128.65** |

This is surprising because unlike solving $P$ with CPLEX, our branch-and-Benders-cut algorithm does not explicitly have a complete description of the problem's polytope but must instead estimate it on the fly by generating Benders cuts. This leads to the possibility of underestimating the solution of the underlying linear programs at each node of the enumeration tree, leading to weak dual bounds. This is more likely to occur in $BB\&C$ since we only allow for one round of Benders cuts to be added at non-root nodes of our enumeration tree. However, due to the enhancements proposed, we leave the root node with a linear program that captures most of the important characteristics in a smaller problem.

On the other hand, our modified cut-and-solve algorithm's performance is also two orders of magnitude faster than CPLEX for large-scale instances while for Class III-B, it saves over 50% of the solution time. On average, $CS/LB$ solves five sparse problems before proving optimality of its obtained solution. Each of these sparse problems are solved up to three orders of magnitude faster than solving the complete problem with CPLEX and sometimes in half the time than if solved with branch-and-Benders-cut algorithm. It is because of these time savings that it outperforms CPLEX in all instances and our branch-and-Benders-cut algorithm in instance group 20,320,200 of class III-B. The advantage of this method is that it finds the optimal solution early on and spends the rest of the time proving optimality by solving another sparse problem followed by the remaining dense problem.

Finally, we point out that while dimensionality does play a role in the computation time required to solve these instances, there exist other factors that contribute to the difficulty of these problems. This can be seen in the difference in solution time between the instance group 30,520,100 and 30,520,400 of class I where the group with four times more commodities is solved in significantly less computing time. The same is seen when comparing differences in number of arcs. Instance group 30,700,100 (class I) requires significantly less computing time than the instance group 30,520,100 (class

I) which has less arcs. Identifying the other factors that make some network design instances particularly difficult for mixed integer programs would allow researchers to devise algorithms with an improved, more stable performance.

## 3.7 Conclusion

We have presented two exact solution algorithms for the multicommodity uncapacitated fixed-charge network design problem that significantly outperform the state-of-the-art general-purpose MIP solver CPLEX. The first exact algorithm is based on implementing Benders decomposition within a branch-and-cut framework using Pareto-optimal cuts, appropriate core point selection, and an in-tree matheuristic. These additional refinements also serve as general guidelines for implementing this algorithm for other mixed integer problems.

We present a novel strategy for improving the LP bound of our Benders reformulation by means of Benders lift-and-project cuts applied to the master problem's feasibility and optimality cuts. These are obtained using a modified cut generating linear program that takes less than 0.02 seconds to solve. This procedure extends beyond the MUFND and can be applied to all problems that allow a mixed integer programming formulation and corresponding Benders reformulation.

Finally, we present a strategy that combines ideas from cut-and-solve/local branching and our proposed branch-and-Benders-cut algorithm. The advantages of this method are: breaking down the problem into a few sparse MIPs which make it easier to obtain high quality feasible solutions, the non-increasing optimal values obtained from the sparse problems, the reduced size of the sparse problem solution space, and the re-usability of Benders cuts generated in previous iterations. The results of our implementation show this fusion to be a promising method for solving large-scale MIPs.

# Chapter 4

# Profit-oriented Fixed-charge Network Design with Elastic Demand


## Abstract

This paper extends classic fixed charge multicommodity network design by explicitly considering demand elasticity with respect to routing cost in a profit maximization context with service commitments. Demand quantity is determined by a spatial interaction model that accounts for routing costs thus capturing the trade-off between infrastructure investment, efficient routing, and increased revenue. A numerical example is presented to demonstrate the added value of incorporating demand elasticity in profit-oriented network design problems. An arc-based and a path-based formulation, both with the flexibility of incorporating O/D pair selection by means of network and data transformations, are presented. The arc-based formulation is solved using state-of-the-art global optimization software while the path-based formulation serves as the basis for a hybrid matheuristic that combines a slope scaling metaheuristic and

column generation. Computational experience shows the hybrid matheuristic to be superior in terms of solution quality and computation time.

## 4.1 Introduction

Fixed charge multicommodity network design (FMND) is a fundamental optimization problem arising in industries such as transportation and communications to capture the trade-off between strategic investment and operational efficiency. The problem consists of selecting a subset of potential arcs to be installed and to route the demand of commodities from several origin/destination (O/D) pairs using only the installed arcs. A fixed cost is incurred upon installing an arc and a unit transportation cost is paid for each unit of commodity transported through each arc. It generalizes a large class of well-known combinatorial problems such as the *shortest path problem*, the *traveling salesman problem*, the *uncapacitated lot-sizing problem*, and the *Steiner network design problem* [105, 181], and models a variety of problems in communications and transportation [128, 138].

In this paper, we extend the FMND by incorporating demand elasticity with respect to routing cost in a profit maximization context where a predetermined amount of revenue is received for each unit of demand routed. The problem considers the same decisions as the classic model: selecting a subset of arcs to install and routing the demand of O/D pairs using them. In addition, the decision maker selects which O/D pairs to serve subject to a service commitment constraint. Since demand quantity is a function that depends on route distance which in turn is determined by the solution of the model, the proposed framework endogenously captures the feedback loop between the network design decisions and demand quantities, leading to equilibrium-like conditions at its optimal solutions.

In the network optimization literature, demand quantities are exogenously esti-

mated via historical information and given as fixed parameters to the problem. This makes the resulting optimization model highly dependent on the quality of the initial demand estimate. To circumvent this, researchers have developed models for FMND that account for demand uncertainty via robust optimization as in Lee et al. [121] and Keyvanshokooh et al. [111] or stochastic programming as in Rahmaniani et al. [155]. The former assumes demand realizations are within a predefined set and finds the best solution considering the worst possible demand realization occurs (risk-averse). The latter assumes that demand follows a given probability distribution and seeks the solution with the best expected value (risk-neutral).

Both approaches assume demand behaves statically in the sense that it does not depend on the network's configuration, i.e., demand is inelastic. This does not hold in many of the applications of FMND. An example is the construction of the U.S.A's interstate highway system which led to a significant increase in transportation of commodities among cities in particular within the "Sunbelt states" (*https://www.fhwa.dot.gov*). Another example is the shift in travel patterns when shorter flight routes are offered between cities [36]. The myopic perspective of ignoring demand elasticity compromises the applicability of FMND at the strategic level where decisions have long-lasting repercussions, a context in which its use is ubiquitous.

In the literature, there are three areas of network design in which elastic demand has been considered: transit network design, service network design, and network pricing. Each models demand elasticity according to the level of detail needed for the problem. In transit network design, elastic demand is incorporated by means of traffic assignment problems (TAP), where demand information is obtained at the link level. Service network design and network pricing problems on the other hand use a distance decay function to estimate demand between O/D pairs.

Traffic assignment problems on congested networks were the earliest to incorporate

elastic demand into network problems. The purpose of these models is to calculate the use of each link (road) on a network considering: 1) users are selfish and will use paths that minimize their travel time; 2) travel time over a link is inversely proportional to the number of users on it; and 3) the number of users going between two points in the network is a function of the travel time. Beckmann et al. [20] presented a non-linear formulation whose optimal solution also solves the TAP, satisfying what later became termed as a "user-equilibrium". Solution algorithms that converge to this user-equilibrium for the fixed demand case were first proposed by Dafermos [65] and LeBlanc et al. [120], while Florian and Nguyen [80] and Evans [74] devised the first efficient algorithms for the elastic demand case. Numerous extensions have been proposed for both the static TAP [135] and the dynamic TAP [152].

As mentioned before, the TAP is a subproblem of transit network design problems in which higher level decisions such as added road capacity, vehicle outgoing frequency, or vehicle sizes must be determined [141]. These problems are posed as bilevel programs in which the upper level seeks to maximize social benefit and the lower level corresponds to a TAP. Due to the difficulty of solving the TAP, in particular when considering elastic demand, and the added challenge of bilevel programming, most solution methods for these problems have been heuristic in nature [44].

Demand elasticity has also been incorporated into location models by means of distance decay functions, spatial interaction models, or user-equilibrum constraints to characterize demand loss due to travel time/cost, congestion and decline in utility. Two families of location problems have considered demand elasticity. The first is Competitive Facility Location in which the decision maker seeks to maximize the market share captured or minimize lost demand by strategically locating facilities for customers whose willingness to patronize it is sensitive to travel and waiting time. Solution algorithms and extensions of these can be found in Aboolian et al. [1], Berman et al. [23], Marianov et al. [131] and Marianov et al. [132]. Other works

with a specific application to healthcare service network design are Zhang et al. [183] and Zhang et al. [184] in which the former uses queueing theory to determine expected demand and the latter models demand elasticity via user-equilibrium conditions to account for willingness to participate in preventive healthcare.

The second family of location problems to consider demand elasticity is the profit maximizing service network design problem presented in Aboolian et al. [2]. The model seeks to determine the optimal facility locations and their corresponding service levels so as to maximize the profit obtained as the difference between the revenue received from the captured demand and the investment in infrastructure. The model accounts for sensitivity to both travel and waiting time where the latter is incorporated as constraints derived from well-known queueing theory results. It is solved to optimality by a successive improvement algorithm that removes non-optimal feasible solutions at each iteration. However, the solution time shows to be sensitive to the allowed minimum number of workers per facility.

The most recent area of network design to incorporate elastic demand is that of network pricing introduced by Labbé et al. [118]. This problem seeks to maximize the revenue raised from tolls placed on a network that must transport multiple commodities. Kuiteing et al. [116] extended the original version to include elastic demand by means of a linearly decreasing function. The resulting problem is posed as a bilevel program and reformulated into a mixed integer quadratic program which is solved by a general-purpose solver. Kuiteing et al. [117] later extended the model to include non-linear demand decay functions and proposed an exact method based on piecewise linear approximations of the demand function that asymptotically converges to the optimal solution.

These studies demonstrate the importance and impact of accounting for elastic demand in strategic network design problems both in "directed choice" models where a central decision maker establishes the O/D routes and in "user-choice" models where

routing is determined by user-equilibria. Recently, Daganzo [66] presented conditions under which demand estimation and system design can be done separately in public infrastructure network design through user-choice models. These problems seek to maximize social benefit by deciding on the system design, including its layout and control, and the prices to be charged for the service. We note that this result does not apply to the problem presented in this paper as it is a directed choice model placed in a more abstract context to allow its use in applications beyond public infrastructure.

Among the classes of problems reviewed, the most closely related to the proposed framework is that of Aboolian et al. [2]. It also captures the trade-off between additional investment and increased revenue. However, the presented model differs in that it assumes a maximum threshold of possible demand, and that elasticity is modeled by both a distance and congestion decay function. In addition, the inherent difference between location and network design problems makes the corresponding approaches significantly different. In location problems, locating a facility directly impacts the travel costs of nearby patrons. On the other hand, the effect on routing costs of installing an arc in a network is dependent on the other arcs that determine the shortest paths of commodities.

With respect to modeling demand elasticity, the methods used in the reviewed literature do not fit well with the assumptions and level of detail of FMND. Compared to the TAP, we do not require traffic levels per link nor does the model assume having the parameters necessary to formulate the equilibrium model. On the other hand, distance decay functions are useful when modeling lost demand whereas the goal of the proposed framework is to also capture the possibility of increasing demand quantity based on network decisions.

In this paper we propose the use of a gravity model to incorporate demand elasticity to routing cost into a profit-oriented variant of FMND. One of the key advantages of the use of the gravity model is that its simplest version allows for the incorporation

of demand elasticity by using an O/D demand matrix as in the classic FMND. On the other hand, more sophisticated gravity models that consider other determining features for demand prediction can be easily incorporated by doing the necessary calibration. This allows for a wide spectrum of possible gravity models to predict demand being incorporated into FMND. The gravity model also generalizes distance decay functions as it does not assume an estimated maximum possible demand. Instead, the maximum possible demand is implicitly determined by the underlying network's shortest distances. In addition, we allow the decision maker to choose which O/D pairs will be served subject to a service commitment constraint that enforces a minimum number of them to be routed. We present two non-linear mixed integer programming formulations obtained by incorporating a general form of the gravity model and demonstrate the added value of incorporating demand elasticity by comparing solutions obtained from the proposed model and from its inelastic version. Both formulations are able to model O/D pair selection by means of simple network transformations. Finally, we present solution algorithms and a computational comparison of their performance with respect to solution quality and computation time.

The rest of the paper is organized as follows. In Section 4.2, we provide some preliminaries on the use of gravity models, present the notation, formally describe the problem, and give a numerical example that demonstrates the value of incorporating demand elasticity. Section 4.3 presents an arc-based and a path-based formulation and the transformations necessary to incorporate O/D pair selection and service commitment constraints. Section 4.4 details the components of the hybrid matheuristic used to solve the path-based formulation while Section 4.5 compares its performance with that of solving the arc-based formulation with a state-of-the-art global optimization solver. Finally, Section 4.6 provides conclusions and future lines of research.

## 4.2 Problem definition

As the proposed framework is based on the FMND, we adopt the same notation. The problem is defined on a directed graph $G = (N, A)$ with node set $N$ and arc set $A$. We assume the existence of a set of O/D pairs, which we denote by $K$, between which demand must be routed on a single path. Each O/D pair will also be referred to as a commodity. Each arc has a corresponding fixed installation cost $f_{ij} \geq 0$ and a unit transportation cost $c_{ij}^k > 0$ for each commodity $k \in K$. A revenue of $\alpha_k \geq 0$ is received for every unit of commodity $k \in K$ that is routed.

There is an added value in allowing the decision maker to select which commodities to route in a profit-oriented problem. A commodity may be left unserved if the resulting installation and operational costs do not compensate the obtained revenue in the overall network design. This additional decision level is incorporated in the problem along with a service commitment constraint that enforces that at least $\Gamma$ O/D pairs are routed, where $\Gamma \in \mathbb{Z}^+$. Considering all previously mentioned characteristics, the proposed problem consists of finding the network configuration that maximizes the total profit obtained from routing at least $\Gamma$ of a set of given O/D pairs. The flexibility of the resulting problem allows it to be used in both regulated industries where service commitments are imposed and unregulated industries where the service provider has complete freedom to selfishly pursue profit.

Given that the total revenue depends on the demand quantities of the commodities routed, the role of demand elasticity to routing cost directly impacts our objective. The proposed problem implicitly seeks an equilibrium between spending on network infrastructure to increase demand quantities of commodities while ensuring maximum profitability in the overall endeavor. We next introduce the preliminaries of gravity models, the tool used to model demand elasticity to routing cost.

### 4.2.1 Gravity models

Based on Newton's law of universal gravitation [142], gravity models have spread to other fields in the social sciences [101]. Since the late 19th century, ideas based on gravity models have been used to explain principles of social science [42], to define laws of migration [157], to explain retail gravitation [159], to model consumer behavior [106], and to analyze traffic patterns [172]. In the optimization community, Huff-like models have been used in competitive facility locations [1, 71, 75] with both market share capture and profit maximization objectives.

The original idea behind gravity models was to measure the interaction between two locations as being directly proportional to their size and inversely proportional to the distance between them. The model's simplified form led to skepticism about its prediction capabilities [108]. As a response, theoretical refinements were made to improve its reliability as a prediction tool. Wilson [173] presented gravity models that considered information restrictions and provided a corresponding taxonomy for the resulting families of gravity models: *unconstrained*, *production constrained*, *attraction constrained*, and *production-attraction constrained*. Later, Senior [163] provided a means of extracting information at a disaggregated level based on entropy maximization. Throughout the years, other refinements have been proposed that allow for the inclusion of industry-specific features into the gravity model [83]. Today, refined and meticulously calibrated gravity models such as those presented in Grosche et al. [97], Hodgson [104], Lesage and Polasek [122], Zhong et al. [185] and Boonekamp et al. [36] provide reliable estimates of interactions between locations.

In general, gravity models assume the interactions $W_{ij}$ from city $i$ to $j$ can be estimated as $W_{ij} = f(\bar{U}_i, \bar{V}_j, \bar{D}_{ij})$ where $\bar{U}_i$ is a vector of origin features, $\bar{V}_j$ is a vector of destination features, $\bar{D}_{ij}$ represents a set of separation attributes, one of which is the travel cost, and $f(.)$ is a real-valued function [83]. To ease notation, we assume

the following simplified version of the gravity model:

$$W_{ij} = \frac{P_i P_j}{(d_{ij})^r}, \tag{4.1}$$

where $P_i$ is a weight attributed to the population size at location $i$, $d_{ij}$ represents the distance between them, and $r \geq 0$ is an exponent that models the sensitivity to distance. As seen in Black [32], depending on the context in which the gravity model is used, different values of $r$ have been shown to better approximate demand patterns. In practice, its value often varies between 0.5 and 2.0 [83].

The formulations and solution algorithms presented in this paper are also compatible with more sophisticated gravity models. Note that the components of $\bar{U}_i$ and $\bar{V}_j$ are not affected by the decisions taken in network design problems. In fact, the only feature affected by network design decisions is the distance between the locations. Therefore, the addition of features would appear in the proposed optimization model as constants. On the other hand the variability of the exponent $r$ is accounted for and, as will be seen in Section 4.4, the presented solution algorithm easily adapts to any value of $r$.

### 4.2.2 Profit-oriented network design with elastic demand

We next present the objective function of the problem resulting from incorporating (4.1) into FMND where distances considered are based on routing costs. Let the variable $x_{ij}^k \in \{0, 1\}$ represent whether arc $(i, j) \in A$ is used in the route of commodity $k \in K$ while $y_{ij}$ represents the activation of arc $(i, j) \in A$. To model revenue, we convert the demand quantity into a monetary value by multiplying it by the per unit revenue $\alpha_k \geq 0$ received for each unit of commodity $k \in K$ routed. We assume that the routing costs used in the gravity model are directly proportional to the transportation costs incurred $c_{ij}^k$, i.e., $d_{ij} = \tau c_{ij}^k$, where $\tau > 0$ is a transportation

cost scaling factor. Substituting the gravity model form (4.1) for the fixed demand parameter $W_k$ of the FMND, we obtain the following objective function for the profit-oriented network design problem with elastic demand (POFMND-E):

$$(OF) \quad \sum_{k\in K} \frac{\alpha_k P_{o(k)} P_{d(k)}}{(\tau \sum\limits_{(i,j)\in A} c_{ij}^k x_{ij}^k)^r} - \sum_{k\in K} \frac{P_{o(k)} P_{d(k)}}{(\tau \sum\limits_{(i,j)\in A} c_{ij}^k x_{ij}^k)^r} \sum_{(i,j)\in A} c_{ij}^k x_{ij}^k - \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{4.2}$$

$$= \sum_{k\in K} P_{o(k)} P_{d(k)} \frac{[\alpha_k - \sum\limits_{(i,j)\in A} c_{ij}^k x_{ij}^k]}{(\tau \sum\limits_{(i,j)\in A} c_{ij}^k x_{ij}^k)^r} - \sum_{(i,j)\in A} f_{ij} y_{ij} \tag{4.3}$$

$$= \sum_{k\in K} PP_k(x) - \sum_{(i,j)\in A} f_{ij} y_{ij}. \tag{4.4}$$

The first, second and third terms of (4.2) correspond to the total revenue, transportation cost and investment cost, respectively. Note that unlike classic multicommodity network design problems, in $OF$ the demand quantities between O/D pairs will depend on the cost of the routes used. This models the effect the decision maker's choice of routes has on demand quantities, i.e., demand elasticity to routing cost.

Simplifying, we obtain (4.3) from which we can more clearly see the profit maximization and demand elasticity characteristics of the non-linear objective function. Note that each addend of the first term, rewritten as $PP_k(x)$ in (4.4), lends itself to the interpretation of "partial profit" obtained from serving commodity $k \in K$. For each O/D pair, it calculates the difference between the per unit revenue and transportation cost, multiplied by the demand quantity obtained according to the gravity model.

Since no assumptions, except non-negativity, are imposed on the value of the per unit revenue $\alpha_k$ of a commodity, the model is capable of handling cases in which for a given commodity $k$, $PP_k(x) < 0$. In other words a loss is incurred. One such case is when the per unit revenue of a commodity is strictly less than the cheapest route over the complete underlying network, $p_s^k$.

An important characteristic of $PP_k(x)$ is that it is the composition of two functions $d_k(x) : \mathbb{R}^n \to \mathbb{R}$ and $PP_k(d_k) : \mathbb{R} \to \mathbb{R}$, where $d_k(x) = \sum_{(i,j)\in A} c_{ij}^k x_{ij}^k$ and $PP_k(d_k) = P_{o(k)}P_{d(k)}\frac{[\alpha_k - d_k]}{(\tau d_k)^r}$. The following proposition uses this to provide insights on the shape of the function $PP_k(x)$.

**Proposition 1.** *For $r > 1$,*

*a) $d_k(x) = \frac{r\alpha_k}{r-1}$ is a minimum of $PP_k(d_k(x))$,*

*b) $PP_k(x)$ is convex for $0 < d_k(x) \leq \frac{(r+1)\alpha_k}{(r-1)}$,*

*c) $PP_k(x)$ is concave for $d_k(x) \geq \frac{(r+1)\alpha_k}{(r-1)}$.*

*Proof.* Given that $PP_k(d_k) : \mathbb{R} \to \mathbb{R}$, i.e., the partial profit of commodity $k$ as a function of distance $d_k > 0$, is a twice differentiable univariate function, we can calculate its first and second order derivatives. We denote $f'(x)$ and $f''(x)$ as the first and second order derivatives of $f(x)$, respectively. Using univariate calculus we obtain the following differential information for $PP_k(d_k)$:

$$PP_k'(d_k) = P_{o(k)}P_{d(k)}\frac{(d_k(r-1) - r\alpha_k)}{\tau^r d^{r+1}}, \tag{4.5}$$

$$PP_k''(d_k) = P_{o(k)}P_{d(k)}r\frac{((r+1)\alpha_k - (r-1)d_k)}{\tau^r d^{r+2}}. \tag{4.6}$$

From (4.5) and (4.6) we observe that for any $r > 1$:

- $PP_k''(d_k) \geq 0$ when $d_k \leq \frac{(r+1)\alpha_k}{(r-1)} \implies PP_k(d_k)$ is convex for $d_k \in (0, \frac{(r+1)\alpha_k}{(r-1)}]$ .
- $PP_k''(d_k) < 0$ when $d_k > \frac{(r+1)\alpha_k}{(r-1)} \implies PP_k(d_k)$ is concave for $d_k \in (\frac{(r+1)\alpha_k}{(r-1)}, \infty)$ .
- $PP_k(d_k)$ obtains a minimum of $\frac{-P_{o(k)}P_{d(k)}\alpha_k(r-1)^{(r-1)}}{(r\tau\alpha_k)^r}$ at $d_k = \frac{r\alpha_k}{r-1}$ since $PP_k'(\frac{r\alpha_k}{r-1}) = 0$ and $PP_k''(\frac{r\alpha_k}{r-1}) > 0$.

Given that $d_k(x)$ is an affine transformation and $PP_k(x)$ is the composition of $PP_k(d_k)$ and $d_k(x)$, we conclude that $PP_k(d_k)$ is convex when $0 < d_k(x) \leq \frac{(r+1)\alpha_k}{(r-1)}$

and concave when $d_k(x) \geq \frac{(r+1)\alpha_k}{(r-1)}$ [38, section 3.2.2] and is therefore a non-convex function throughout the domain $x \in \mathbb{R}^{|A|}$. $\square$

The case of $0 \leq r \leq 1$ merits special attention because the function $PP_k(x)$ possesses properties that can be exploited from a mathematical programming perspective. Although the scope of this study is to present a general solution methodology adaptable to any $r \geq 0$, we present these special characteristics for completeness.

Substituting $0 \leq r \leq 1$ into (4.6) we note that $PP_k''(d_k) \geq 0$ for all $d_k \in \mathbb{R}$. This implies that $PP_k(d_k)$ is a convex function throughout the entire domain of $x \in \mathbb{R}$. In fact, as $r \mapsto 0$, $PP_k(d_k)$ tends to become more linear, achieving linearity at $r = 0$. In addition, not only is $PP_k(d_k)$ convex throughout its domain, it also does not attain its minimum. This can be seen by substituting $0 \leq r \leq 1$ into the expression $(4.5) = 0$ whose solutions give the stationary points of $PP_k(d_k)$. We note that a key assumption of these results is that $c_{ij}^k > 0$ for all $k \in K$ and $(i, j) \in A$, since routes with a cost of zero are undefined in the presented definition of partial profit. The advantage of convexity lies in the fact that there exist efficient ways of dealing with $OF$ by using subgradients to under-approximate each $PP_k(x)$.

Figure 4.1 shows the shape of $PP_k(d_k)$ with respect to per unit routing cost $d_k$ for three parameter values of $r$. For a given $k \in K$, note that the partial profit $PP_k(d_k)$ may take negative values, i.e., incur a loss, when the per unit routing cost $d_k$ is greater than $\alpha_k$. After attaining its minimum, which as seen from our previous analysis will always be a maximum loss, if the demand sensitivity parameter $r > 1$ then an increase in per unit routing cost decreases the loss. Therefore in this particular case, after a certain point, there is an incentive for the decision maker to route commodities on a longer path so as to dissuade demand of a particular commodity whose efficient routing does not globally compensate the revenue obtained.

On the other hand, if $r \in [0, 1]$ then the partial profit $PP_k(d_k)$ is a convex function that keeps decreasing as the routing cost increases therefore there would be no incen-

tive to route commodities through a longer route. This flexibility in modeling demand behavior and its effect on the partial profit is one of the most important reasons for using the gravity model. It generalizes schemes such as demand decay functions and is able to capture phenomena such as the effect of hyper-sensitivity to routing costs ($r > 1$) on partial profit that may be missed from more conventional demand functions. Finally, we point out that in the context of our problem, the values $d_k(x)$ are bounded above and below by the most expensive ($p_l^k$) and the cheapest ($p_s^k$) possible routes, respectively, in the complete underlying graph. Therefore, depending on their values, the corresponding problem may consider only the convex part of $PP_k$ when $r > 1$. However, in general, the problem to be solved is a non-linear optimization problem for all $r \geq 0$.

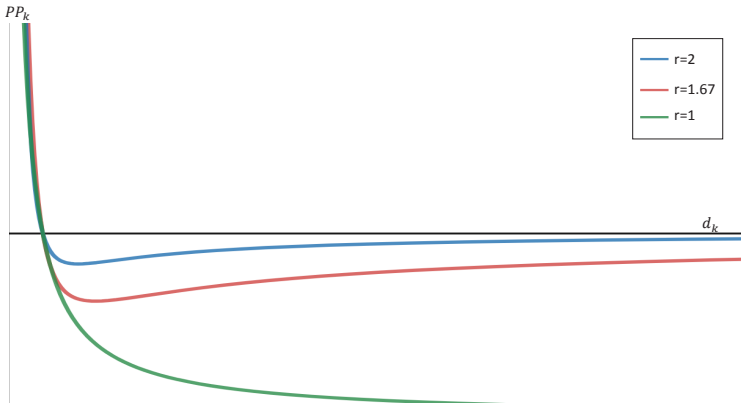



Figure 4.1: Shape of $\mathrm{PP}_k(\mathrm{d}_k)$

These nuances capture other levels of trade-off inherent to profit maximization where taking losses or offering worse service levels for some commodities compensates the overall profitability of an enterprise's operation. The proposed framework can also be used as a means of finding how to create commodity groups with better operation synergy, i.e., considering them together leads to decisions that do not dissuade demand by offering poorer service.

Comparing the shape of the partial profit for varying values of $r$, we note that the minimum value of partial profit is larger for greater values of $r$. This comes from

the fact that larger values of $r$ represent greater demand sensitivity to routing cost leading to fewer units of the commodity being routed at the minimum value of partial profit and therefore a smaller loss.

## 4.3 Formulations

We next propose two non-linear mixed integer programming formulations for the *profit-oriented network design problem with elastic demand* (POFMND-E) in which all commodities must be served. These formulations can be easily adapted to incorporate O/D pair selection decisions and service commitments. Both formulations fall in the domain of global optimization problems. Given the limited resources to efficiently solve these types of problems, we use one of the formulations as a base to develop a hybrid matheuristic that is applicable to any $r \geq 0$. It exploits known results of similar network design problems and converges in a reasonable amount of computation time. The other formulation serves as a benchmark for solving the POFMND-E with O/D pair selection and service commitments using a state-of-the-art general purpose global optimization software.

### 4.3.1 Arc-based formulation

Our arc-based formulation is based on the well-known strong formulation for the uncapacitated FMND with the difference that the routing variables $x_{ij}^k$ are binary instead of continuous. In the case of POFMND-E, when the per unit revenue $\alpha_k$ of a commodity $k$ is less than the cheapest possible route $p_s^k$, the model then seeks to make the loss resulting from $PP_k(x)$ as small as possible by increasing the length of the route as much as possible. Since flow conservation constraints alone do not prohibit circuits, formulations based only on the constraints of the classic uncapacitated FMND formulation lead to solutions having routes with sub-circuits within the path

and isolated from it.

Given the inability of most global optimization software to allow for cut callbacks, we implement the subtour elimination constraints (SEC) via a modified version of the well-known, but weak, Miller-Tucker-Zemlin (MTZ) subtour elimination constraints [137]. These require an additional set of variables $u_i^k$ for each $i \in N, k \in K$ and that $x_{ij}^k$ be binary leading to the following arc-based formulation. The variables $u$ represent the order in which all nodes except the depot are visited.

$$(P_1) \quad \text{maximize} \sum_{k \in K} P_{o(k)} P_{d(k)} \frac{[\alpha_k - \sum\limits_{(i,j) \in A} c_{ij}^k x_{ij}^k]}{(\tau \sum\limits_{(i,j) \in A} c_{ij}^k x_{ij}^k)^r} - \sum_{(i,j) \in A} f_{ij} y_{ij} \tag{4.7}$$

$$\text{subject to} \quad \sum_{j \in N:(j,i) \in A} x_{ji}^k - \sum_{j \in N:(i,j) \in A} x_{ij}^k = \begin{cases} -1 & \text{if } i = o_k \\ 1 & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, \forall k \in K \tag{4.8}$$

$$x_{ij}^k \leq y_{ij} \quad \forall (i,j) \in A, k \in K \tag{4.9}$$

$$u_{o_k}^k = 1 \quad \forall k \in K \tag{4.10}$$

$$u_i^k - u_j^k + 1 \leq (N-1)(1 - x_{ij}^k) \quad \forall (i,j) \in A, k \in K \tag{4.11}$$

$$2 \leq u_i^k \leq N \qquad \forall i \in N \backslash \{o_k\}, k \in K \tag{4.12}$$

$$x_{ij}^k \in \{0,1\} \qquad \forall (i,j) \in A, k \in K \tag{4.13}$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j) \in A. \tag{4.14}$$

The objective function (4.7) seeks to maximize profit, while constraints (4.8) and (4.9) are the flow conservation constraints and binding constraints from the classic uncapacitated FMND, respectively. Finally, constraints (4.10) and (4.11) are the MTZ subtour elimination constraints, while (4.12)-(4.14) are the variable definitions.

As a required input of the algorithms used in global optimization software, we provide the lower and upper bounds for the values of each $PP_k(x)$ as detailed in Section 4.2.2. This model is solved with the general-purpose global optimization solver Baron 18.8.23 [114, 161, 170] accessed through the AMPL modeling language.

### 4.3.2 Path-based formulation and pricing problem

We next present a path-based formulation of the problem. Let $\Theta_k^\mu$ denote a binary variable whose value is equal to 1 if path $\mu$ is used for commodity $k$, and define the parameter $v_k^\mu(i,j)$=1 if arc $(i,j)$ belongs to path $\mu$ for commodity $k$, 0 otherwise. Finally, let $\Omega_k$ denote the set of simple paths from $o(k)$ to $d(k)$ and $\Omega$ represent the union of these over $k \in K$. With this notation we have the following path-based formulation for the POFMND-E.

$$(P_2) \qquad \text{maximize} \sum_{k\in K}\sum_{\mu\in\Omega_k}[P_{o(k)}P_{d(k)}\frac{[\alpha_k - \sum\limits_{(i,j)\in A} c_{ij}^k v_k^\mu(i,j)]}{(\tau \sum\limits_{(i,j)\in A} c_{ij}^k v_k^\mu(i,j))^r}]\Theta_k^\mu - \sum_{(i,j)\in A} f_{ij}y_{ij} \quad (4.15)$$

$$\text{subject to } [\lambda] \quad \sum_{\mu\in\Omega_k} v_k^\mu(i,j)\Theta_k^\mu \le y_{ij} \qquad \forall (i,j)\in A, \forall k \in K \quad (4.16)$$

$$[\mu] \quad \sum_{\mu\in\Omega_k} \Theta_k^\mu = 1 \qquad \forall k \in K \quad (4.17)$$

$$\Theta_k^\mu \in \{0,1\} \qquad \forall (i,j)\in A. \quad (4.18)$$

$$y_{ij} \in \{0,1\} \qquad \forall (i,j)\in A. \quad (4.19)$$

The objective function (4.15) calculates the total profit obtained from the selected network configuration. Constraint set (4.16) ensures that all design variables of the arcs used in the routing take value one while constraints (4.17) ensure that each commodity is routed through one path. Finally, $\lambda_{ij}^k \ge 0$, $\forall (i,j)\in A$, $\forall k \in K$ and $\mu_k \in \mathbb{R}$, $\forall k \in K$ are the dual variables corresponding to (4.16) and (4.17), respectively.

We point out that, unlike $P_1$, the presented path-based formulation is a binary linear formulation.

Given the exponential number of potential paths that can be used to route each commodity, traditionally these are added on the fly to $P_2$ by means of a column generation algorithm which adds columns based on the solution of a pricing problem that determines whether new columns are needed to calculate the linear relaxation of $P_2$. In this case, the pricing problems of POFMND-E decompose to one for each commodity $k \in K$. These have the following form for each $k \in K$:

$$(Pr_2^k) \quad \text{maximize} \quad [P_{o(k)}P_{d(k)}\frac{[\alpha_k - \sum_{(i,j)\in A} c_{ij}^k v_k^\mu(i,j)]}{(\tau \sum_{(i,j)\in A} c_{ij}^k v_k^\mu(i,j))^r}] - \sum_{(i,j)\in A} \lambda_{ij} v_k^\mu(i,j) \quad (4.20)$$

$$\text{subject to} \quad \sum_{j\in N:(j,i)\in A} v_k^\mu(j,i) - \sum_{j\in N:(i,j)\in A} v_k^\mu(i,j) = \begin{cases} -1 & \text{if } i = o_k \\ 1 & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N \quad (4.21)$$

$$u_{o_k}^k = 1 \quad \forall k \in K \quad (4.22)$$

$$u_i^k - u_j^k + 1 \leq (N-1)(1 - v_k^\mu(i,j)) \quad \forall (i,j) \in A \quad (4.23)$$

$$2 \leq u_i^k \leq N \qquad \forall i \in N \backslash \{o_k\} \quad (4.24)$$

$$v_k^\mu(i,j) \in \{0,1\} \qquad \forall (i,j) \in A. \quad (4.25)$$

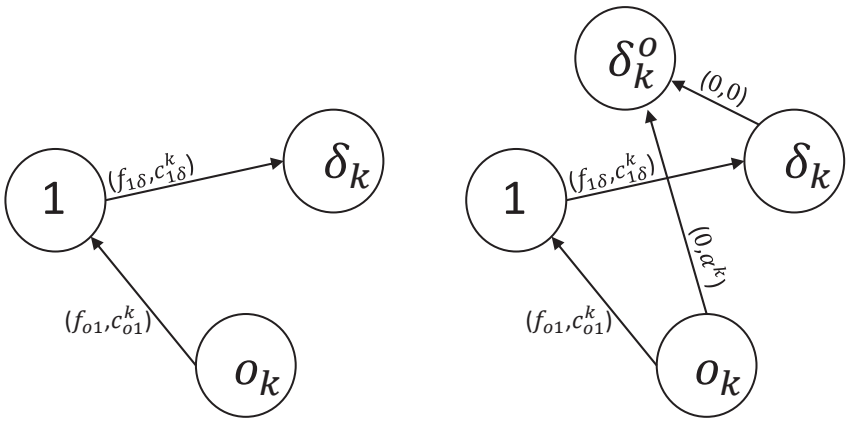
The objective function (4.20) seeks to maximize the reduced cost of the column to be added while constraints (4.21) are the classic flow conservation constraints. Constraints (4.22) and (4.23) are the MTZ subtour elimination constraints. Note that $\text{Pr}_2^k$ is similar to $P_1$ in that it is a non-linear formulation, with binary variables and MTZ subtour elimination constraints. However, $\text{Pr}_2^k$ is significantly smaller since it is defined for each $k \in K$ and the fixed costs are replaced by the dual variables of

(4.16). It is a modified shortest path problem with a non-linear objective function. The difficult non-convexity present in $P_1$ is transferred to the pricing problem $Pr_2^k$. Our efforts will therefore be placed in finding alternative ways of adding new columns to $P_2$ by heuristically solving an easier problem based on $Pr_2^k$.

### 4.3.3 Incorporating O/D pair selection and service commitments

Formulations $P_1$ and $P_2$ do not account for O/D pair selection and service commitment. Both formulations allow for the convenient incorporation of both considerations by either performing simple transformations on the network or defining artificial variables, and adding a knapsack-type constraint, respectively. For $P_1$ to allow O/D pair selection, the only modification required is done to the underlying graph used in the formulation. The following network transformation incorporates O/D pair selection into $P_1$.

Let $\Delta = \{i \in N | \exists (o,d) \in K \text{ such that } i = d\}$ denote the set of nodes that are destinations of some commodity $k \in K$. For each node $\delta \in \Delta$ create an artificial node $\delta^o$ and an arc $(\delta, \delta^o) \in A$ with $f_{\delta\delta^o} = 0$ and $c^k_{\delta\delta^o} = 0$ for all $k \in K$. In addition, redefine all commodities $(o, \delta)$ as $(o, \delta^o) \in K$ and add another artificial arc $(o, \delta^o) \in A$ with $f_{\delta\delta^o} = 0$ and $c^k_{\delta\delta^o} = \alpha_k$ for all $k \in K$. Figure 4.2 illustrates the proposed transformation.




(a) Original network (b) Transformed network

Figure 4.2: Network transformation to allow O/D pair selection

Note that by carrying out this transformation and using the resulting network in $P_1$, its solution accounts for the decision maker selecting which commodities to route. If a commodity $\bar{k} \in K$ used the arc $(o, \delta^o) \in A$, it obtains no revenue and incurs no fixed cost. In other words, the model has selected not to route commodity $\bar{k} \in K$. On the other hand, if a commodity uses the arc $(\delta, \delta^o) \in A$, its routing, fixed cost, and corresponding revenue remain the same as if it was routed from $(o, \delta)$ in the original network.

Finally, to impose a service commitment constraint in $P_1$ stating that at least $\Gamma$ commodities are to be routed, the following knapsack type constraint on the arc variables of the transformed network should be added:

$$\sum_{(o,\delta^o)\in K} x^{o\delta^o}_{o\delta^o} \leq |K| - \Gamma. \tag{4.26}$$

Incorporating O/D pair selection in $P_2$ requires significantly less work. The addition of an empty route $\gamma_{\bar{k}}$, with $PP_{\bar{k}(x)} = 0$ for each commodity $\bar{k} \in K$ into the pool of routes $\Omega_k$ is enough to account for O/D pair selection. The associated binary vari-

ables for these empty routes will be denoted as $\Theta_{\bar{k}}^{\mu^0}$. If in a solution, the associated $\Theta_{\bar{k}}^{\mu^0}$ is equal to 1, then commodity $\bar{k}$ is not routed.

Imposing a service commitment in $P_2$ where at least $\Gamma$ commodities are to be routed is also done by incorporating a knapsack-type constraint. In this case, the inequality is defined over the artificial variables $\Theta_{\bar{k}}^{\mu^0}$ of the empty routes and has the following form:

$$\sum_{k \in K} \Theta_k^{\mu^0} \leq |K| - \Gamma. \tag{4.27}$$

The network transformations or empty routes can be added as a preprocessing step leaving the incorporation of either inequality (4.26) or (4.27) as the only modification of $P_1$ and $P_2$, respectively, to incorporate O/D pair selection and service commitment. These transformations allow us to formulate the POFMND-E with commodity selection and a service commitment of $\Gamma$, denoted as POFMND-E($\Gamma$). $P_1(\Gamma)$ is defined as $P_1$ with the addition of constraint (4.26) while $P_2(\Gamma)$ is defined as $P_2$ with the addition of constraint (4.27).

In the interest of brevity, particular attention will be placed on the two extreme cases $\Gamma = |K|$ and $\Gamma = 0$ referred to as variants I and II, respectively. Note that for both cases, one can omit the service commitment constraints (4.26) or (4.27). In variant I, it suffices to not carry out the network transformation or not define the empty routes in $P_1$ or $P_2$, respectively. In variant II, the service commitment constraints become redundant as the right hand side value $|K|$ is the maximum possible value that can be taken.

We next present a numerical example to show the value of incorporating demand elasticity in profit-oriented network design. Although the example presents a comparison for variant I, a similar conclusion can be derived for the general POFMND-E($\Gamma$).

### 4.3.4 The value of considering demand elasticity

To demonstrate the value of considering demand elasticity within POFMND, we solve variant I for a small example of a network with 10 nodes numbered from one to ten, 35 arcs and ten commodities. Table 4.1 details the parameters of the ten commodities: origin, destination, population at origin, population at destination, per unit revenue and cheapest route over the complete network, respectively.

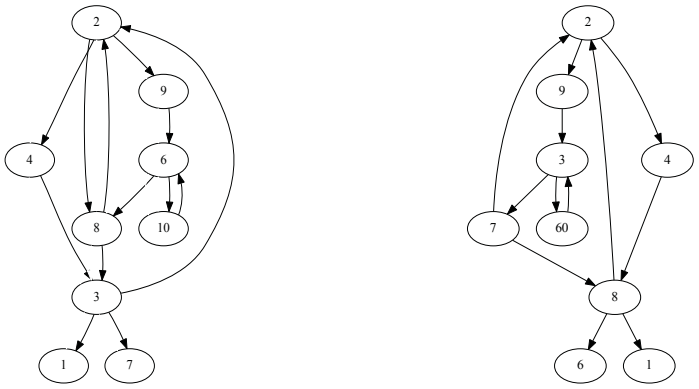
Table 4.1: Commodity parameters of example

| $o_k$ | $d_k$ | $P_o$ | $P_d$ | $\alpha_k$ | $p_s^k$ |
|---|---|---|---|---|---|
| 2 | 8 | 524 | 1,076 | 93 | 93 |
| 3 | 1 | 228 | 744 | 141 | 76 |
| 4 | 2 | 792 | 524 | 210 | 122 |
| 4 | 7 | 792 | 160 | 189 | 106 |
| 6 | 1 | 640 | 744 | 245 | 141 |
| 6 | 10 | 640 | 448 | 53 | 41 |
| 8 | 1 | 1,076 | 744 | 220 | 117 |
| 8 | 6 | 1,076 | 640 | 157 | 93 |
| 9 | 8 | 292 | 1,076 | 215 | 123 |
| 10 | 4 | 448 | 792 | 304 | 164 |

As mentioned in Section 4.2, POFMND-E obtains solutions with equilibrium-like characteristics between demand quantity and routing cost. Therefore, to formulate an equivalent profit-oriented FMND with inelastic demand (POFMND-I), we must assume a fixed demand quantity for each commodity. A demand quantity value that exploits all information in the instance data is to assign the demand quantity between nodes $i$ and $j$ as $W_{ij} = \frac{P_i P_j}{(p_s^k)^r}$. This is an optimistic value of demand quantity as it assumes all commodities are routed along the path with least travel cost over the complete underlying network. Note that by fixing demand quantity, the model becomes a binary linear program which can then be solved by any of a plethora of available integer linear programming tools.

Figure 4.3a is the optimal solution POFMND-I while Figure 4.3b is the optimal

solution of POFMND-E.




(a) POFMND-I (b) POFMND-E

Figure 4.3: Optimal solutions POFMND where all commodities are served

We note that at a design level, the optimal solution POFMND-I has one additional arc from node $2 \rightarrow 8$ as a result of routing commodity (2,8) directly from origin to destination. POFMND-E instead routes this commodity along nodes $2 \rightarrow 9 \rightarrow 6 \rightarrow 8$ with a routing cost of 136. POFMND-I routes all commodities except (6,1) along their cheapest possible path, $p_s^k$, therefore their corresponding demand quantity values coincided with that obtained from the presented gravity model. To ensure the solutions of both models are comparable, we substitute the solution obtained from POFMND-I into the objective function of POFMND-E. The objective function value obtained from POFMND-I is 9.55% worse than that of POFMND-E. This comes as a result of POFMND-I not having the flexibility to identify the overall benefit of routing commodity (2,8) along a more costly, unprofitable route with less demand. We next present a hybrid matheuristic algorithm for solving $P_2$.

## 4.4 Solving the path-based formulation

To exploit existing algorithms for classic network design problems, we replace the use of $Pr_2^k$ with the classic uncapacitated FMND formulation to generate new paths. By

doing this, we are no longer able to obtain dual bounds to assess the solution quality. However, given that our purpose is to develop a fast heuristic adaptable to any value of $r \geq 0$, the loss of dual bounds does not hinder our purpose. On the other hand, the classic uncapacitated FMND is itself solved by a metaheuristic since its main purpose is to generate a varied set of paths. As we will see later on, the solutions obtained from our proposed heuristic are optimal for most cases for which the general purpose global optimization solver was able to prove optimality.

### 4.4.1 A hybrid matheuristic for the path-based formulation

We begin by presenting the logic behind the tools used for the matheuristic part of the proposed solution algorithm. For each $k \in K$ let $W_k = \frac{P_{o(k)}P_{d(k)}}{(\tau \sum\limits_{(i,j)\in A} c_{ij}^k v_k^\mu(i,j))^r}$ then:

$$\text{maximize} \sum_{k\in K} P_{o(k)}P_{d(k)} \frac{[\alpha_k - \sum\limits_{(i,j)\in A} c_{ij}^k x_{ij}^k]}{(\tau \sum\limits_{(i,j)\in A} c_{ij}^k x_{ij}^k)^r} - \sum_{(i,j)\in A} f_{ij}y_{ij} \tag{4.28}$$

$$\iff \text{maximize} \sum_{k\in K} W_k[\alpha_k - \sum_{(i,j)\in A} c_{ij}^k x_{ij}^k] - \sum_{(i,j)\in A} f_{ij}y_{ij} \tag{4.29}$$

$$\iff \text{minimize} \sum_{(i,j)\in A} f_{ij}y_{ij} + \sum_{k\in K}\sum_{(i,j)\in A} W_k c_{ij}^k x_{ij}^k \tag{4.30}$$

where the equivalence between (4.29) and (4.30) comes from the fact that $W_k\alpha_k$ is now a constant. Expression (4.30) is in fact the objective function of the uncapacitated FMND and methods for this problem can thus be applied to obtain new routes, update $W_k$ based on the best routes found, and then repeat.

Given that the uncapacitated FMND instance obtained assumes a particular value for $W_k$, it is not in our interest to solve it exactly. A more suitable strategy is to use a metaheuristic that generates diverse routes for each commodity. In this spirit, we use a slope-scaling metaheuristic (SS) based on that presented in Crainic et al. [63]

for the capacitated FMND. This heuristic will be detailed later on in this section.

We next outline the algorithm which embeds the slope-scaling metaheuristic within two loops, the first updates the assumed $W_k$ while the second solves a relaxation of the path-based formulation and updates the transportation costs of the network used for the slope-scaling.

We begin by initializing $W_k$ based on the maximum possible demand, $\nabla_k$, calculated by substituting the cheapest route $p_s^k$ into (4.1). If the per unit revenue $\alpha_k$ of commodity $k$ is less than the cheapest route $p_s^k$, we initialize $W_k$ as $0.2\nabla_k$. Otherwise we initialize it as $W_k = \nabla_k$.

The slope-scaling metaheuristic is then called and all routes generated during the process are stored in $\Omega$ with the proper evaluation of $PP_k(x)$. $W_k$ is then updated based on the routes of the best solution found, with respect to the uncapacitated FMND, and the process is then repeated. This loop is terminated when the values of $W_k$ no longer change. We then add all generated paths to the formulation $\mathrm{P}_2(\Gamma)$ and solve its linear relaxation. The values of $\lambda$ in $\mathrm{P}_2(\Gamma)$ are then used to update the transportation costs of each commodity $k \in K$ on each arc $(i,j) \in A$ as $c_{ij}^k = c_{ij}^k + \lambda_{ij}^k$ and the entire process is repeated.

The purpose of this algorithm is to keep generating new routes based on modified demand quantities and transportation costs. Greater diversification leads to a richer pool $\Omega$ of routes in the master problem $\mathrm{P}_2(\Gamma)$. We denote this problem as $\mathrm{P}_2(\Gamma, \Omega(t))$. The final loop is terminated when no new paths have been generated. After this step, we then proceed to solve the restricted master problem $\mathrm{P}_2(\Gamma, \Omega(t))$ as an integer program to obtain a heuristic solution. Algorithm 3 summarizes the proposed hybrid matheuristic.

**Algorithm 3** Hybrid matheuristic for POFMND-E

**Initialization:** $t = 0$; $\Omega(t) = \emptyset$; $W_k(0) = \nabla_k$ if $\alpha_k > p_s^k$, $W_k(0) = 0.2\nabla_k$ otherwise;
**do**
  **do**
    Execute SS; add all generated routes $\mu_k$ to $\Omega_k(t)$; best solution=$(\bar{y}, \bar{x})$.
    Update $W_k(t+1) = (P_{o(k)}P_{d(k)})/(\tau \sum_{(i,j)\in A} c_{ij}^k \bar{x}_{ij}^k)^r$
    t=t+1
  **while** ($\exists k$ such that $|W_k(t+1) - W_k(t)| > 0$)
  Solve LP of restricted master problem $\mathrm{P}_2(\Gamma, \Omega(t))$; obtain dual variables $\lambda$.
  Update $c_{ij}^k = c_{ij}^k + \lambda_{ij}^k$
**while** ($|\Omega(t)| > |\Omega(t-1)|$)
Solve restricted master problem $\mathrm{P}_2(\Gamma, \Omega(t))$ with integrality constraints.

We point out that in general the metaheuristic SS can be substituted by any metaheuristic that generates a rich variety of paths. Other metaheuristics such as local and neighborhood searches can also prove effective if embedded within this hybrid matheuristic. The key to the complete procedure is to keep information at the path level therefore allowing $\mathrm{P}_2$ the freedom to choose paths that were previously not considered together during the metaheuristic phase. A similar logic is used in Zetina et al. [181].

### 4.4.2 A Slope Scaling Metaheuristic

Slope scaling was first presented in Yaged [174] and later in Kim and Pardalos [112, 113] as a heuristic to solve network optimization problems. Crainic et al. [63] improved it by adding Lagrangean perturbation and long term memory to help in diversifying and intensifying the search. The method is based on the idea that there exists a linear program of the form

$$\text{(SS)} \quad \min \sum_{k \in K} \sum_{(i,j) \in A} \hat{c}_{ij}^k x_{ij}^k \tag{4.31}$$

$$\text{s.t.} \quad \sum_{j \in N} x_{ji}^k - \sum_{j \in N} x_{ij}^k = \begin{cases} -W^k & \text{if } i = o_k \\ W^k & \text{if } i = d_k \\ 0 & \text{otherwise} \end{cases} \quad \forall i \in N, \forall k \in K \tag{4.32}$$

$$0 \leq x_{ij}^k \leq W^k \qquad \forall (i,j) \in A, k \in K, \tag{4.33}$$

that obtains the same optimal solution as the uncapacitated FMND. The algorithm attempts to estimate the $\hat{c}_{ij}^k$ for which this equivalence holds by defining it as $\hat{c}_{ij}^k = c_{ij}^k + \rho_{ij}^k$, where $\rho_{ij}^k$ is a slope scaling factor that estimates the contribution of the fixed costs. An initial $\rho_0$ is chosen to begin the algorithm. At each iteration $t$, SS($\rho_t$) is solved and its solution is used to obtain $\rho_{t+1}$.

For our implementation, SS is split into $|K|$ shortest path problems with arc lengths of $\hat{c}_{ij}^k$. We use a multi-start method with different initial values of $\rho$ based on the fixed cost and demand quantity. Let $\bar{W}^k = \sum_{k \in K} W^k / |K|$, i.e., the average demand quantity of the commodities. The initial values for $\rho$ are

- $f_{ij} / \sum_{k \in K} W^k$: the fixed cost divided by the total demand quantity;
- $f_{ij} / \max_{k \in K} W^k$: the fixed cost divided by the largest demand quantity;
- $f_{ij} / \bar{W}^k$: the fixed cost divided by the average demand quantity;
- $f_{ij} / [(\max_{k \in K} W^k - \bar{W}^k)/2]$: the fixed cost divided by the mid-point between the average and maximum demand quantities;
- $f_{ij} / [(\sum_{k \in K} W^k - \bar{W}^k)/2]$: the fixed cost divided by the mid-point between the average and total demand quantities;

- $f_{ij}/[(\sum_{k\in K} W^k - \max_{k\in K} W^k)/2]$: the fixed cost divided by the mid-point between the total and maximum demand quantities.

Upon obtaining the optimal solution $\tilde{x}$ of SS($\rho(i)$), the slope scaling factor is updated as

$$\rho_{ij}^k(i+1) = \begin{cases} \frac{f_{ij}}{\sum_{k\in K} \tilde{x}_{ij}^k} & \text{if } \sum_{k\in K} \tilde{x}_{ij}^k > 0 \\ \rho_{ij}^k(i) & \text{otherwise.} \end{cases} \tag{4.34}$$

This process is continued until a given number of iterations ($T_{SS}$) have been performed. Note that upon solving SS, a feasible solution can be constructed for FMND by fixing to 1 the arcs through which some flow has been sent and solving $|K|$ shortest path problems over this subgraph. To improve the quality of the solution, we then remove any arcs of the subgraph that have not been used in the shortest path of at least one of the commodities.

When two successive iterations obtain the same solution $\tilde{x}$, then the procedure will not produce any new distinct solutions. This may occur before having performed the $T_{SS}$ iterations. To aid in diversifying the metaheuristic's search, we implement a perturbation tool similar to that of Crainic et al. [63]. When two successive iterations of solving SS obtain the same optimal value or a determined number of iterations $T_{pert}$ without an improved solution have passed, we then solve a shortest path problem for each $k \in K$ and use the corresponding dual variables $(\lambda_k, \mu_k)$ of the classic shortest path formulation to update the slope scaling factor as $\rho_{ij}^k = -\lambda_j^k + \lambda_i^k + \mu_{ij}^k$ and continue iterating until a maximum number $T_{SS}$ of SS models have been solved.

Finally, as in Crainic et al. [63], we implement a long term memory mechanism in which we keep statistics throughout the history of the search. These statistics are used to update $\rho$ and restart the process. Based on whether the current round of the algorithm produced an improved best solution, we choose to update $\rho$ in such a way to intensify or diversify the search. The statistics kept for each $(i, j) \in A$ and $k \in K$

up to iteration $T$ are the number of iterations for which $\tilde{x}_{ij}^k > 0$ ($n_{ij}^k(T)$), average number of commodities routed through each arc ($\bar{x}_{ij}(T)$) and maximum number of commodities routed through each arc ($\hat{x}_{ij}^k(T)$).

After performing $T_{SS}$ iterations with the corresponding dual perturbations along the way, we calculate for each $(i,j) \in A$, $v_{ij} = \bar{x}_{ij}(T_{SS})/\hat{x}_{ij}(T_{SS})$ or $v_{ij} = 0$ if $\bar{x}_{ij}(T_{SS}) = 0$. Here, $v_{ij}$ measures the variability of the number of commodities sent through arc $(i,j)$ throughout the last $T_{SS}$ iterations. Hence, $v_{ij} \approx 1$ means the number of commodities sent through this arc has been stable throughout the process, while $v_{ij} \approx 0$ shows high variability or no commodities sent at all. During intensification, variables with stable behavior are favored while the opposite is done when a diversification step is taken.

An intensification update to $\rho$ is done if in the last cycle an improved best solution was obtained. Otherwise, a diversification step is taken. A limit of $div_{max}$ and $int_{max}$ diversification and intensification updates, respectively, are applied throughout the algorithm. The updates for each scheme are presented below where $\bar{n}$ and $S_n$ are the average and standard deviation of $n_{ij}^k$, respectively.

- Normalize $\rho_{ij}^k := \rho_{ij}^k - \min_{(i,j)\in A, k\in K} \rho_{ij}^k$ so $\rho_{ij}^k \geq 0 \ \forall (i,j) \in A, k \in K$.
- To apply the intensification scheme, $\forall (i,j) \in A, k \in K$
  - If $n_{ij}^k \geq \bar{n} + S_n$ then $\rho_{ij}^k := \rho_{ij}^k(1 - v_{ij})$
  - If $n_{ij}^k \leq \bar{n}$ then $\rho_{ij}^k := \rho_{ij}^k(2 - v_{ij})$
  - Else $\rho_{ij}^k := \rho_{ij}^k$.
- To apply the diversification scheme, $\forall (i,j) \in A, k \in K$
  - If $n_{ij}^k \geq \bar{n} + S_n$ then $\rho_{ij}^k := \rho_{ij}^k(1 + v_{ij}^k)$
  - If $n_{ij}^k \leq \bar{n}$ then $\rho_{ij}^k := \rho_{ij}^k(v_{ij}^k)$
  - Else $\rho_{ij}^k := \rho_{ij}^k$.

This metaheuristic is called several times within our hybrid matheuristic to generate new paths for which we evaluate the true value of $PP_k(x)$. For this reason, it is stopped prematurely if two consecutive dual perturbations do not lead to any new paths. Our heuristic is run with the parameter values $T_{SS} = 30, T_{DP} = 5, div_{max} = 10, int_{max} = 2$. Figure 4.4 graphically summarizes the slope scaling with dual perturbation and long term memory algorithm sequentially applied to each of the six initial $\rho$ values.

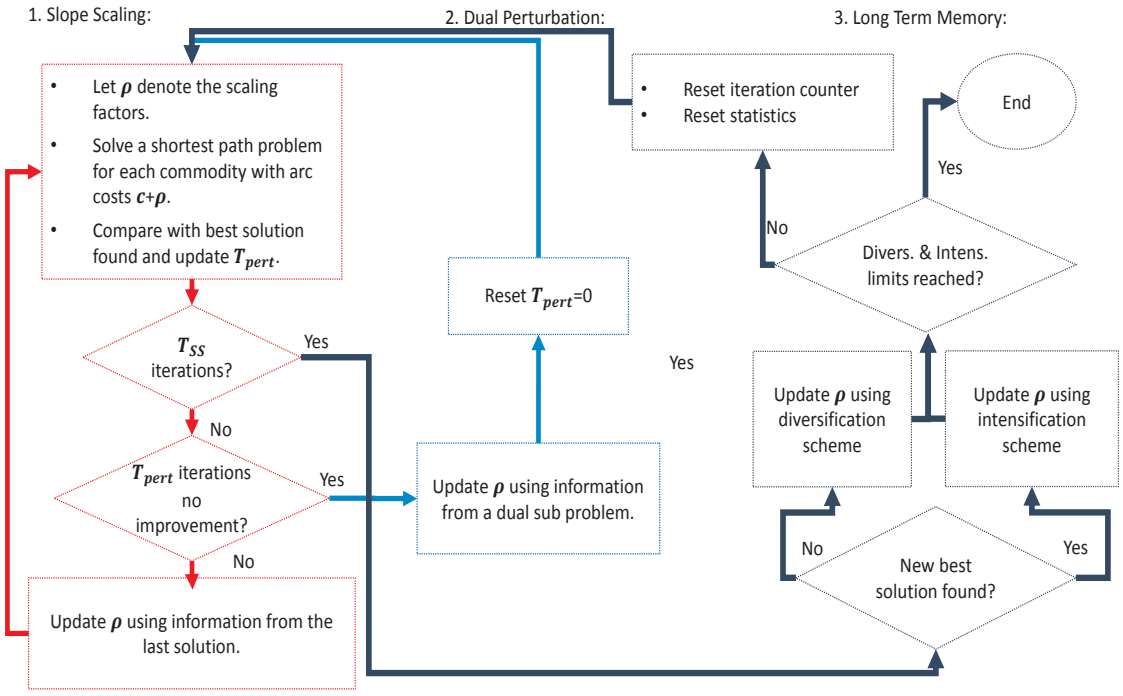



Figure 4.4: Slope scaling heuristic for MUFND

## 4.5 Computational Experiments

We test the computational efficiency of the proposed formulations and solution algorithms using the well-known "Canad" multicommodity capacitated network design testbed [62]. The population weights of each node $P_i$ are calculated as the sum of all inbound and outbound demand in each node multiplied by four. To adjust for nodes $i \in N$ with no inflow or outflow, we assign them population values of $P_{min}$ which

denotes the minimum non-zero population calculated. Per unit revenue of each commodity is $\alpha_k = p_s^k + \sigma p_s^k$ where $\sigma$ is a random number between 0 and 1 and $p_s^k$ is the cheapest possible route, i.e., the shortest path over the complete network. Finally, to adjust for the population factors $P_i$ for $i \in N$, the value of the fixed costs of an arc $(i, j)$ is adjusted based on the original value of the Canad instance $\bar{f_{ij}}$ as follows: $f_{ij} = \frac{\bar{f_{ij}} P_i P_j}{\nu_{ij}}$ where $\nu_{ij} = P_{min} + .75[\max\{P_i, P_j\} - P_{min}]$. In total our testbed is comprised of 85 instances ranging from small to medium scale. The computational experiments will first focus on the two extreme variants of POFMND-E($\Gamma$), where variant I refers to $\Gamma = |K|$ and variant II refers to $\Gamma = 0$. As previously mentioned, in these extreme cases one can omit the service commitment constraint.

The arc-based formulation $P_1$ is solved using the branch-and-reduce algorithm implemented in the general purpose global optimization software Baron 18.8.23 [114, 161, 170] through its AMPL interface. The hybrid matheuristic is coded in C using CPLEX 12.7.0 to solve the restricted master problem and the shortest path subproblems within the slope-scaling metaheuristic. For a fair comparison, all use of CPLEX was limited to one thread and the traditional MIP search strategy. All experiments were executed on an Intel Xeon E5 2687W V3 processor at 3.10 GHz under Linux environment with a time limit of two hours. Finally, we fix the parameters of POFMND-E to $\tau = 1$ and $r = 1.7$. Tables 4.2 and 4.3 compare the performance of $P_1$ solved using Baron and the proposed hybrid matheuristic for variant I and II, respectively.

The first three columns of Table 4.2 contain the instance class (Class), dimensions (N,A,K) representing the number of nodes, arcs and commodities respectively, and the number of instances in each group (Nb.). The next two columns contain the number of instances for which a feasible solution was found by the hybrid matheuristic and its average time to completion in seconds. The following three columns correspond to the number of instances for which solving $P_1$ with Baron found a feasible and optimal

solution, respectively, while the last column details the average time taken.

Table 4.2: Performance comparison-Variant I

| | | | Hybrid Matheuristic | | Baron | | |
|---|---|---|---|---|---|---|---|
| **Class** | **(N,A,K)** | **Nb.** | **Feasible Sols** | **Seconds** | **Feasible Sols** | **Optimal Sols** | **Seconds** |
| I | 20,230,40 | 3 | 3 | 0.47 | 3 | 0 | 7,200.00 |
| | 20,230,200 | 4 | 4 | 1.17 | 0 | 0 | 7,200.00 |
| | 20,300,40 | 4 | 4 | 0.50 | 4 | 0 | 7,200.00 |
| | 20,300,200 | 4 | 4 | 1.62 | 0 | 0 | 7,200.00 |
| | 30,520,100 | 4 | 4 | 2.70 | 0 | 0 | 7,200.00 |
| | 30,520,400 | 4 | 4 | 7.64 | 0 | 0 | 7,200.00 |
| | 30,700,100 | 4 | 4 | 3.09 | 0 | 0 | 7,200.00 |
| | 30,700,400 | 4 | 4 | 9.73 | 0 | 0 | 7,200.00 |
| | **Subtotal** | **31** | **31** | **3.45** | **7** | **0** | **7,200.00** |
| III-A | 10,35,10 | 3 | 3 | 0.04 | 3 | 3 | 99.67 |
| | 10,35,25 | 3 | 3 | 0.08 | 3 | 1 | 4,807.00 |
| | 10,35,50 | 3 | 3 | 0.12 | 3 | 0 | 7,200.00 |
| | 10,60,10 | 3 | 3 | 0.03 | 3 | 3 | 13.00 |
| | 10,60,25 | 3 | 3 | 0.14 | 3 | 1 | 4,862.00 |
| | 10,60,50 | 3 | 3 | 0.23 | 3 | 0 | 7,200.00 |
| | 10,85,10 | 3 | 3 | 0.04 | 3 | 1 | 4,808.00 |
| | 10,85,25 | 3 | 3 | 0.14 | 3 | 0 | 7,200.00 |
| | 10,85,50 | 3 | 3 | 0.29 | 3 | 0 | 7,200.00 |
| | **Subtotal** | **27** | **27** | **0.12** | **27** | **9** | **4,821.81** |
| III-B | 20,120,40 | 3 | 3 | 0.78 | 3 | 0 | 7,200.00 |
| | 20,120,100 | 3 | 3 | 2.36 | 3 | 0 | 7,200.00 |
| | 20,120,200 | 3 | 3 | 4.04 | 2 | 0 | 7,200.00 |
| | 20,220,40 | 3 | 3 | 1.94 | 3 | 0 | 7,200.00 |
| | 20,220,100 | 3 | 3 | 2.42 | 3 | 0 | 7,200.00 |
| | 20,220,200 | 3 | 3 | 5.22 | 0 | 0 | 7,200.00 |
| | 20,320,40 | 3 | 3 | 1.66 | 3 | 0 | 7,200.00 |
| | 20,320,100 | 3 | 3 | 3.25 | 3 | 0 | 7,200.00 |
| | 20,320,200 | 3 | 3 | 5.24 | 0 | 0 | 7,200.00 |
| | **Subtotal** | **27** | **27** | **2.99** | **20** | **0** | **7,200.00** |
| | **Total** | **85** | **85** | **2.25** | **54** | **9** | **6,551.46** |

As seen from Table 4.2 not only is the hybrid matheuristic able to find a feasible solution for all instances compared to only 54/85 for solving $P_1$ with Baron, it also requires significantly less time. In addition, for nine of the small instances of Class III-A, Baron's branch-and-reduce method applied to $P_1$ is able to find and prove an optimal solution. These solutions coincide with that obtained by our hybrid matheuristic for eight of the nine instances while for the remaining instance, the solution found by the hybrid matheuristic is 0.29% away from the optimal. Similar

behavior can be observed when solving variant II of POFMND-E that allows for the decision maker to freely select a subset of commodities to route. Table 4.3 presents a summary of the performance of both algorithms applied to variant II with the same column definitions as Table 4.2.

Table 4.3: Performance comparison-Variant II

| | | **Hybrid Matheuristic** | | **Baron** | | |
|---|---|---|---|---|---|---|
| **Class** | **Nb.** | **Feasible Sols** | **Seconds** | **Feasible Sols** | **Optimal Sols** | **Seconds** |
| Class I | 31 | 31 | 3.29 | 11 | 0 | 6,769.90 |
| Class III-A | 27 | 27 | 0.13 | 27 | 14 | 3,511.56 |
| Class III-B | 27 | 27 | 2.81 | 22 | 0 | 6,654.04 |
| **Total** | **85** | **85** | **2.14** | **60** | **14** | **5,698.09** |

We next analyze the quality of the solutions found. Based on the results seen in Tables 4.2 and 4.3, we note that solving $P_1$ with Baron requires a significant amount of computation time. To assess whether this effort is compensated by better solution quality, we analyze the instances for which both algorithms obtained a feasible solution within the time limit. The study is limited to instances for which optimality of the solution was not proven for the classes I and III-B, i.e., the execution of Baron reached the time limit before proving optimality. Table 4.4 details the objective function value, in millions, of the feasible solutions found in Table 4.2 by the hybrid matheuristic and Baron, respectively. The last column contains the relative difference between them calculated as $\frac{100*|HM-Bar|}{|HM|}$ where $HM$ refers to the value of the solution from the hybrid matheuristic and $Bar$ refers to the value of the solution from Baron.

Table 4.4: Comparison of solution quality-Variant I

| Class | instance | Hybrid Matheuristic Obj (Millions) | Baron Obj Baron (Millions) | Rel. Difference |
|---|---|---|---|---|
| | c33.dat | 229.95 | 9.60 | 95.83 |
| | c35.dat | 106.99 | 26.14 | 75.57 |
| | c36.dat | 132.14 | 12.24 | 90.74 |
| I | c41.dat | 225.16 | (3.92) | 101.74 |
| | c42.dat | 245.59 | (23.44) | 109.54 |
| | c43.dat | 213.82 | (42.58) | 119.92 |
| | c44.dat | 173.11 | (62.10) | 135.88 |
| | r10.1.dow | 1.20 | 0.60 | 50.26 |
| | r10.2.dow | 0.86 | 0.12 | 86.07 |
| | r10.3.dow | 0.52 | (0.24) | 145.67 |
| | r11.1.dow | 10.65 | (1.43) | 113.41 |
| | r11.2.dow | 9.18 | (1.87) | 120.38 |
| | r11.3.dow | 7.61 | (2.38) | 131.21 |
| | r12.2.dow | 89.94 | (15.64) | 117.39 |
| | r12.3.dow | 84.47 | (17.19) | 120.35 |
| | r13.1.dow | 1.56 | (0.21) | 113.50 |
| III-B | r13.2.dow | 1.18 | (0.41) | 134.88 |
| | r13.3.dow | 0.76 | (0.60) | 177.92 |
| | r14.1.dow | 15.66 | (5.06) | 132.33 |
| | r14.2.dow | 14.01 | (5.34) | 138.13 |
| | r14.3.dow | 12.04 | (5.32) | 144.17 |
| | r16.1.dow | 1.50 | (0.48) | 132.16 |
| | r16.2.dow | 1.09 | (0.53) | 148.39 |
| | r16.3.dow | 0.67 | (0.61) | 191.48 |
| | r17.1.dow | 19.32 | (6.22) | 132.19 |
| | r17.2.dow | 17.38 | (5.41) | 131.14 |
| | r17.3.dow | 15.18 | (5.97) | 139.32 |
| **Total** | | **60.43** | **(5.86)** | **109.70** |

Table 4.4 shows that the solutions found by the proposed hybrid matheuristic are of significantly better quality. It finds profitable solutions for all instances in an average of 2.25 seconds as shown in Table 4.2. The results show that not only is it difficult for Baron to find feasible solutions with formulation $P_1$ but also that the solutions found are of poor quality. This comes as a result of the solver not exploiting the network structure of the problem whereas the use of our slope scaling heuristic bypasses the difficulty presented by the non-convexities of $Pr_2^k$ to generate good paths.

The proposed hybrid matheuristic also obtains superior solutions than solving $P_1$ with Baron when solving variant II. Table 4.5 contains the instance class (Class), dimensions (N,A,K) representing the number of nodes, arcs and commodities respectively, and the number of instances considered in each group (Nb.). The following two columns represent the average objective function value of the best found solution for

each instance group while the last column represents the average relative difference between them calculated as in Table 4.4.

Table 4.5: Comparison of solution quality-Variant II

| Class | (N,A,K) | Nb. | Hybrid Matheuristic Obj (Millions) | Baron Obj (Millions) | Rel. Difference |
|---|---|---|---|---|---|
| | 20,230,40 | 2 | 119.57 | 51.15 | 58.96 |
| | 20,230,200 | 1 | 576.78 | 576.83 | 0.01 |
| **I** | 20,300,40 | 2 | 229.72 | 58.79 | 74.25 |
| | 30,520,100 | 2 | 26.25 | 13.91 | 46.18 |
| | **Subtotal** | **7** | **189.69** | **117.79** | **51.25** |
| | 20,120,40 | 2 | 0.82 | 0.77 | 6.00 |
| | 20,120,100 | 2 | 8.49 | 8.27 | 2.46 |
| | 20,120,200 | 1 | 84.69 | 5.57 | 93.42 |
| **III-B** | 20,220,40 | 3 | 1.22 | 0.46 | 61.78 |
| | 20,220,100 | 2 | 14.88 | 7.90 | 49.47 |
| | 20,320,40 | 2 | 0.93 | 0.00 | 99.50 |
| | 20,320,200 | 3 | 159.54 | 52.71 | 67.04 |
| | **Subtotal** | **15** | **41.15** | **13.27** | **52.98** |
| | **Total** | **35** | **88.41** | **46.52** | **52.43** |

As opposed to the previous analysis, the solutions found by Baron are all profitable. This comes from the added flexibility of being able to freely select which commodities to route. In addition, the difference between the best solutions found by both algorithms is less significant than for variant I. Unlike the behavior seen when solving variant I, Baron was able to find a better solution than the proposed hybrid matheuristic for an instance in the 20,230,200 group of class I. This shows the increased difficulty of solving variant I with general purpose global optimization solvers. Being obliged to route all commodities forces the decision maker to consider influencing demand quantity of commodities with lower margins.

We next compare the results and solution process of the hybrid matheuristic for both variants. Table 4.6 details for each instance group, the number of instances, average objective function value of the best solution found in millions (Obj Millions), the average CPU time (Seconds), and average number of master problem iterations (CG Iters) of variant I and II. In addition, the last columns present the average % of commodities not routed in the solution obtained from variant II (% Unserved) and

the average relative increase in profit between the solutions obtained from variants I and II.

Table 4.6: Variant I vs II

| | | | Variant I | | | Variant II | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| **Class** | **(N,A,K)** | **Nb.** | **Obj (Millions)** | **Seconds** | **CG Iters** | **Obj (Millions)** | **Seconds** | **CG Iters** | **% Unserved** | **% Profit Inc.** |
| I | 20,230,40 | 3 | 156.36 | 0.47 | 3.00 | 156.37 | 0.35 | 3.00 | 4.19 | 0.00 |
| | 20,230,200 | 4 | 548.92 | 1.17 | 3.00 | 548.96 | 1.24 | 3.00 | 24.00 | 0.01 |
| | 20,300,40 | 4 | 214.42 | 0.50 | 3.25 | 214.43 | 0.47 | 3.00 | 7.50 | 0.01 |
| | 20,300,200 | 4 | 667.60 | 1.62 | 3.00 | 667.63 | 1.55 | 3.00 | 31.00 | 0.00 |
| | 30,520,100 | 4 | 23.21 | 2.70 | 3.00 | 23.22 | 2.72 | 3.25 | 16.25 | 0.04 |
| | 30,520,400 | 4 | 2,132.69 | 7.64 | 3.00 | 2,132.71 | 6.17 | 3.00 | 24.44 | 0.00 |
| | 30,700,100 | 4 | 29.70 | 3.09 | 3.00 | 29.70 | 2.63 | 3.00 | 16.75 | 0.02 |
| | 30,700,400 | 4 | 2,427.40 | 9.73 | 3.00 | 2,427.43 | 10.51 | 3.00 | 30.19 | 0.00 |
| | **Subtotal** | **31** | **795.00** | **3.45** | **3.03** | **795.01** | **3.29** | **3.03** | **19.78** | **0.01** |
| III-A | 10,35,10 | 3 | 0.05 | 0.04 | 3.33 | 0.06 | 0.02 | 3.00 | 26.67 | 45.65 |
| | 10,35,25 | 3 | 0.68 | 0.08 | 3.00 | 0.70 | 0.09 | 3.33 | 16.00 | 2.65 |
| | 10,35,50 | 3 | 4.53 | 0.12 | 3.00 | 4.55 | 0.14 | 3.33 | 3.33 | 0.40 |
| | 10,60,10 | 3 | 0.10 | 0.03 | 3.67 | 0.11 | 0.03 | 3.00 | 20.00 | 9.35 |
| | 10,60,25 | 3 | 0.75 | 0.14 | 4.00 | 0.77 | 0.17 | 3.00 | 10.67 | 3.07 |
| | 10,60,50 | 3 | 6.90 | 0.23 | 3.33 | 6.90 | 0.31 | 3.33 | 0.00 | 0.00 |
| | 10,85,10 | 3 | 0.06 | 0.04 | 3.67 | 0.07 | 0.03 | 3.33 | 26.67 | 31.71 |
| | 10,85,25 | 3 | 1.05 | 0.14 | 4.00 | 1.05 | 0.12 | 3.00 | 5.33 | 0.51 |
| | 10,85,50 | 3 | 6.19 | 0.29 | 3.00 | 6.23 | 0.25 | 3.00 | 8.67 | 0.71 |
| | **Subtotal** | **27** | **2.26** | **0.12** | **3.44** | **2.27** | **0.13** | **3.15** | **13.04** | **10.45** |
| III-B | 20,120,40 | 3 | 0.86 | 0.78 | 5.00 | 0.95 | 0.69 | 3.33 | 28.33 | 15.02 |
| | 20,120,100 | 3 | 9.15 | 2.36 | 3.33 | 9.22 | 2.39 | 3.00 | 9.33 | 0.83 |
| | 20,120,200 | 3 | 89.67 | 4.04 | 4.00 | 89.76 | 3.72 | 3.33 | 3.33 | 0.11 |
| | 20,220,40 | 3 | 1.17 | 1.94 | 5.00 | 1.22 | 1.98 | 4.33 | 25.00 | 6.68 |
| | 20,220,100 | 3 | 13.90 | 2.42 | 3.33 | 14.04 | 2.39 | 3.00 | 9.67 | 1.10 |
| | 20,220,200 | 3 | 131.42 | 5.22 | 3.33 | 131.53 | 4.71 | 3.67 | 4.50 | 0.09 |
| | 20,320,40 | 3 | 1.08 | 1.66 | 4.67 | 1.12 | 1.11 | 3.33 | 18.33 | 5.05 |
| | 20,320,100 | 3 | 17.29 | 3.25 | 3.33 | 17.45 | 3.78 | 3.00 | 9.00 | 0.99 |
| | 20,320,200 | 3 | 159.19 | 5.24 | 3.00 | 159.54 | 4.51 | 3.00 | 4.33 | 0.23 |
| | **Subtotal** | **27** | **47.08** | **2.99** | **3.89** | **47.20** | **2.81** | **3.33** | **12.43** | **3.34** |
| | **Total** | **85** | **305.61** | **2.25** | **3.44** | **305.66** | **2.14** | **3.16** | **15.30** | **4.39** |

One of the important characteristics of the proposed hybrid matheuristic is its ability to adapt to varying values of $r$ and variants of the problem without it significantly modifying its performance. Table 4.6 confirms the latter to be true. Both require similar computation time and number of master problem iterations. Allowing the decision maker to freely choose which commodities to route (variant II), leads to an average of 4.39% increase in profit obtained by choosing not to route an average of approximately 15% of the commodities.

We point out that this marginal increase in profitability is highly dependent on the instance data. Since in the generated instances there are no commodities whose

revenue is smaller than the shortest possible path, it is not surprising that on average there is not a significant difference between the profit of the best solutions for variants I and II. The same would not be said if the per unit revenue of some commodities was less than their corresponding cheapest route.

We next analyze one of the instances in the testbed with highest percentage of commodities being left unserved (50%) for variant II. The instance consists of ten nodes, 35 arcs and ten commodities. We solve the corresponding POFMND-E($\Gamma$) for each service commitment level $\Gamma \in \{1, 2, .., 10\}$. Figure 4.5 plots the total profit (in thousands) of the optimal solution against the service commitment value $\Gamma$.

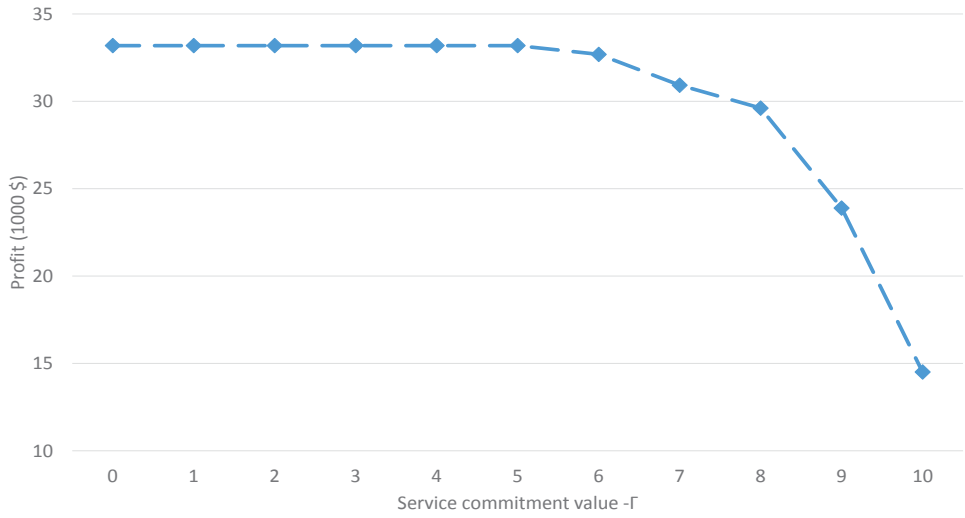



Figure 4.5: The effect of imposing service commitment constraints

Note that imposing a service commitment value of up to 5 commodities has no effect on the profit obtained. This coincides with what was seen from the result of variant II where 50% of the commodities were served despite no service commitment being imposed. As the service commitment value increases, the maximum profit obtained decreases. This decrease becomes more pronounced as the values of $\Gamma$ are closer to $|K|$ with a decrease of 39.3% in profit when increasing $\Gamma$ from $|K| - 1$ to $|K|$. This result suggests that the decision maker should seek to arrange service commitment levels close to that obtained from variant II since these have a marginal

effect on profit.

## 4.6 Conclusion

We have extended the classic fixed charge multicommodity network design problem by incorporating demand elasticity to travel cost in a profit-oriented problem by means of the gravity model. The proposed problem allows the decision maker to choose which O/D pairs will be served subject to a service commitment constraint. The resulting model captures additional levels of trade-off missing in classic fixed charge network design such as the effect of efficient routes on expected demand. We proposed two non-linear mixed integer programming formulations that model the profit-oriented fixed-charge multicommodity network design problem with elastic demand. We showed how both can incorporate O/D pair selection without modifying the formulations by carrying out simple network transformations or adding artificial variables. We also note that the inclusion of a service commitment constraint that requires a minimum number of O/D pairs be served can be done via an additional knapsack-type constraint. We proposed a flexible hybrid matheuristic capable of solving the problem for varying gravity model parameters and service commitment values. Computational results show this algorithm to be superior in terms of both solution time and quality when compared to the use of a general purpose global optimization software. The results also give managerial insights on the establishment of service commitments. The proposed framework allows decision makers to benefit from the added value of incorporating demand elasticity in the optimization of their strategic network design.

# Chapter 5

# Conclusions

This thesis addressed three aspects of network design problems: parameter uncertainty, demand elasticity, and computational efficiency using mixed integer programming tools. It contributed to the current literature by presenting modelling techniques to consider simultaneous parameter uncertainty and a novel framework that integrates concepts from geography into mixed integer programming to account for demand elasticity. These contributions show the capacity of mixed integer programming tools to handle problems in a more realistic setting with imperfect problem information. In addition, new computational tools obtained from the combination of seemingly unrelated mixed integer programming concepts were presented, adding to the repertoire of solution algorithms available to solve mixed integer programs.

In Chapter 2, we addressed demand and transportation cost uncertainty in hub location by means of robust counterparts. Models considering parameter uncertainty individually and simultaneously were presented with the latter leading to a non-linear formulation. This was linearized to a formulation with exponentially many constraints which were incorporated on the fly via a branch-and-cut algorithm. Finally, an analysis of the solution performance in both risk-averse and risk-neutral settings was conducted.

Chapter 3 presented two exact algorithms to solve large scale instances of the well-known multicommodity uncapacitated fixed-charge network design problem. We presented both algorithmic and methodological novelties such as the simultaneous exploitation of two formulations, the use of Lift-and-Project cuts in the Benders reformulation, a tailor-made corepoint selection strategy for Pareto optimal Benders cuts, and a combination of cut-and-solve/local branching enumeration with Benders decomposition. Computational experiments show both exact algorithms to be up to three orders of magnitude faster than a state-of-the-art mixed integer programming solver.

Finally, Chapter 4 presented an extension of the classic fixed-charge network design problem that incorporated elastic demand, profit maximization, commodity selection, and service commitment. Two formulations for the problem were presented, one with a linear number of variables and one with exponentially many. A novel hybrid matheuristic that combines metaheuristics with column generation concepts was presented to solve the formulation with exponentially many variables. This algorithm showed to be significantly better both in solution time and quality than solving the formulation with linearly many variables using a state-of-the-art global optimization solver.

Beyond the contributions mentioned, the research results also serve as first steps toward exploring new lines of thought for the use of mixed integer programming, both convex and non-convex, in real-life applications. The hybridization approaches presented in Chapters 3 and 4 show the added value of tailoring tools such as general mixed integer cuts, branching rules, and metaheuristics to allow for smooth integration into one efficient solution algorithm. On the other hand, the comparison between the robust and stochastic approaches to parameter uncertainty leads to considering a more wholesome approach to these tools by providing a methodology by which fair comparisons can be done. Finally, the incorporation of demand elasticity into opti-

mization models presented in Chapter 4 opens a whole new line of research and breaks the historic setting of assuming parameter uncertainty to be independent of decisions within the optimization process. Researchers could investigate the use of other tools apart from the gravity model to incorporate demand elasticity or propose efficient exact algorithms for the models presented. More importantly, the analysis on the impact of ignoring demand elasticity shows how crucial it is for strategic decisions. This in itself should motivate researchers in the field to find novel ways of addressing this shortcoming.

# Bibliography


[1] R. Aboolian, O. Berman, and D. Krass. Competitive facility location and design problem. *European Journal of Operational Research*, 182(1):40–62, 2007.

[2] R. Aboolian, O. Berman, and D. Krass. Profit maximizing distributed service system design with congestion and elastic demand. *Transportation Science*, 46 (2):247–261, 2012.

[3] Y. Adulyasak, J.-F. Cordeau, and R. Jans. Benders decomposition for production routing under demand uncertainty. *Operations Research*, 63(4):851–867, 2015.

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

[5] S. A. Alumur, B. Y. Kara, and O. E. Karasan. Multimodal hub location and hub network design. *Omega*, 40(6):927 –939, 2012.

[6] S. A. Alumur, S. Nickel, and F. Saldanha-da-Gama. Hub location under uncertainty. *Transportation Research Part B: Methodological*, 46(4):529–543, 2012.

[7] Y. An, Y. Zhang, and B. Zeng. The reliable hub-and-spoke design problem: Models and algorithms. *Transportation Research Part B: Methodological*, 77: 103–122, 2015.

[8] J. Andersen, T. G. Crainic, and M. Christiansen. Service network design with management and coordination of multiple fleets. *European Journal of Operational Research*, 193(2):377–389, 2009.

[9] A. Atamtürk. On capacitated network design cut–set polyhedra. *Mathematical Programming*, 92(3):425–437, 2002.

[10] A. Atamtürk and D. Rajan. On splittable and unsplittable flow capacitated network design arc–set polyhedra. *Mathematical Programming*, 92(2):315–333, 2002.

[11] N. Azizi, S. Chauhan, S. Salhi, and N. Vidyarthi. The impact of hub failure in hub-and-spoke networks: Mathematical formulations and solution techniques. *Computers and Operations Research*, 65:174–188, 2016.

[12] A. Balakrishnan, T. L. Magnanti, and R. T. Wong. A dual-ascent procedure for large-scale uncapacitated network design. *Operations Research*, 37(5):716–740, 1989.

[13] N. Balakrishnan and R. T. Wong. A network model for the rotating workforce scheduling problem. *Networks*, 20(1):25–42, 1990.

[14] E. Balas. Disjunctive programming. *Annals of Discrete Mathematics*, 5:3–51, 1979.

[15] E. Balas and M. W. Padberg. On the set-covering problem. *Operations Research*, 20(6):1152–1161, 1972.

[16] E. Balas and M. Perregaard. Lift-and-project for mixed 0–1 programming: recent progress. *Discrete Applied Mathematics*, 123(1):129–154, 2002.

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

[18] E. Balas, S. Ceria, and G. Cornuéjols. Mixed 0–1 programming by lift-and-project in a branch-and-cut framework. *Management Science*, 42(9):1229–1246, 1996.

[19] J. J. Bartholdi, J. B. Orlin, and H. D. Ratliff. Cyclic scheduling via integer programs with circular ones. *Operations Research*, 28(5):1074–1085, 1980.

[20] M. Beckmann, C. McGuire, and C. Winsten. *Studies in the Economics of Transportation*. Yale University Press, 1956.

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

[22] J.-F. Benders. Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4:238–252, 1962.

[23] O. Berman, D. Krass, and J. Wang. Locating service facilities to reduce lost demand. *IIE Transactions*, 38(11):933–946, 2006.

[24] D. Bertsimas and V. Goyal. On the power of robust solutions in two-stage stochastic and adaptive optimization problems. *Mathematics of Operations Research*, 35(2):284–305, 2010.

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

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

[27] D. Bertsimas, E. Litvinov, X. Sun, J. Zhao, and T. Zheng. Adaptive robust optimization for the security constrained unit commitment problem. *IEEE Transactions on Power Systems*, 28(1):52–63, Feb 2013.

[28] D. Bertsimas, I. Dunning, and M. Lubin. Reformulation versus cutting-planes for robust optimization. *Computational Management Science*, 13(2):195–217, 2016.

[29] D. Bienstock and O. Günlük. Capacitated network design–polyhedral structure and computation. *INFORMS Journal on Computing*, 8(3):243–259, 1996.

[30] D. Bienstock and N. Özbay. Computing robust basestock levels. *Discrete Optimization*, 5(2):389–414, 2008.

[31] J. Billheimer and P. Gray. Network design with fixed and variable cost elements. *Transportation Science*, 7(1):49–74, 1973.

[32] W. R. Black. An analysis of gravity model distance exponents. *Transportation*, 2(3):299–312, Oct 1973.

[33] M. Bodur and J. R. Luedtke. Mixed-integer rounding enhanced Benders decomposition for multiclass service-system staffing and scheduling with arrival rate uncertainty. *Management Science*, 63(7):2073–2091, 2017.

[34] M. Bodur, S. Dash, O. Günlük, and J. Luedtke. Strengthened Benders cuts for stochastic integer programs with continuous recourse. *INFORMS Journal on Computing*, 29(1):77–91, 2017.

[35] T. Boffey and A. Hinxman. Solving the optimal network problem. *European Journal of Operational Research*, 3(5):386–393, 1979.

[36] T. Boonekamp, J. Zuidberg, and G. Burghouwt. Determinants of air travel demand: The role of low-cost carriers, ethnic links and aviation-dependent employment. *Transportation Research Part A: Policy and Practice*, 112:18–28, 2018.

[37] Q. Botton, B. Fortz, L. Gouveia, and M. Poss. Benders decomposition for the hop-constrained survivable network design problem. *INFORMS Journal on Computing*, 25(1):13–26, 2013.

[38] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, 2004.

[39] M. Bruglieri, M. Ehrgott, H. W. Hamacher, and F. Maffioli. An annotated bibliography of combinatorial optimization problems with fixed cardinality constraints. *Discrete Applied Mathematics*, 154(9):1344–1357, 2006.

[40] J. F. Campbell and M. E. O'Kelly. Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169, 2012.

[41] J. F. Campbell, A. Ernst, and M. Krishnamoorthy. Hub arc location problems: part I: Introduction and results. *Management Science*, 51(10):1540–1555, 2005.

[42] H. C. Carey. *Principles of social science*. J.B. Lippincott & Co , Philadelphia, PA, 1858.

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

[44] E. Cipriani, S. Gori, and M. Petrelli. Transit network design: A procedure and an application to a large urban area. *Transportation Research Part C: Emerging Technologies*, 20(1):3–14, 2012. Special issue on Optimization in Public Transport.

[45] S. Climer and W. Zhang. Cut-and-solve: An iterative search strategy for combinatorial optimization problems. *Artificial Intelligence*, 170(8):714–738, 2006.

[46] I. Contreras. Hub location problems. In G. Laporte, F. Saldanha-da-Gama, and S. Nickel, editors, *Location Science*, pages 311–344. Springer, 2015.

[47] I. Contreras and E. Fernández. Hub location as the minimization of a supermodular set function. *Operations Research*, 62(3):557–570, 2014.

[48] I. Contreras, J. A. Díaz, and E. Fernández. Lagrangean relaxation for the capacitated hub location problem with single assignment. *OR Spectrum*, 31(3): 483–505, Jun 2009.

[49] I. Contreras, J.-F. Cordeau, and G. Laporte. Benders decomposition for large-scale uncapacitated hub location. *Operations Research*, 59(6):1477–1490, 2011.

[50] I. Contreras, J.-F. Cordeau, and G. Laporte. Stochastic uncapacitated hub location. *European Journal of Operational Research*, 212(3):518–528, 2011.

[51] I. Contreras, J. A. Díaz, and E. Fernández. Branch and price for large-scale capacitated hub location problems with single assignment. *INFORMS Journal on Computing*, 23(1):41–55, 2011b.

[52] I. Contreras, M. Tanash, and N. Vidyarthi. Exact and heuristic approaches for the cycle hub location problem. *Annals of Operations Research*, 258(2):655–677, 2016.

[53] J.-F. Cordeau, G. Stojković, F. Soumis, and J. Desrosiers. Benders decomposition for simultaneous aircraft routing and crew scheduling. *Transportation Science*, 35(4):375–388, 2001.

[54] J.-F. Cordeau, F. Pasin, and M. M. Solomon. An integrated model for logistics network design. *Annals of Operations Research*, 144(1):59–82, Apr 2006.

[55] D. Corneil and Y. Perl. Clustering and domination in perfect graphs. *Discrete Applied Mathematics*, 9(1):27–39, 1984.

[56] I. Correia and F. Saldanha-da-Gama. Facility location under uncertainty. In G. Laporte, S. Nickel, and F. Saldanha-da Gama, editors, *Location Science*, pages 177–203. Springer, 2015.

[57] I. Correia, S. Nickel, and F. Saldanha-da-Gama. Single-assignment hub location problems with multiple capacity levels. *Transportation Research Part B: Methodological*, 44(8–9):1047–1066, 2010.

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

[59] A. M. Costa, J.-F. Cordeau, and B. Gendron. Benders, metric and cutset inequalities for multicommodity capacitated network design. *Computational Optimization and Applications*, 42(3):371–392, 2009.

[60] T. G. Crainic. Service network design in freight transportation. *European Journal of Operational Research*, 122(2):272–288, 2000.

[61] T. G. Crainic and J.-M. Rousseau. Multicommodity, multimode freight transportation: A general modeling and algorithmic framework for the service network design problem. *Transportation Research Part B: Methodological*, 20(3): 225–242, 1986.

[62] T. G. Crainic, A. Frangioni, and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1–3):73–99, 2001.

[63] T. G. Crainic, B. Gendron, and G. Hernu. A slope scaling/Lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *Journal of Heuristics*, 10(5):525–545, Sep 2004.

[64] F. Cruz, J. Smith, and G. Mateus. Solving to optimality the uncapacitated fixed-charge network flow problem. *Computers & Operations Research*, 25(1): 67–81, 1998.

[65] S. Dafermos. *Traffic Assignment and Resource Allocation in Transportation Networks*. PhD thesis, Johns Hopkins University, 1968.

[66] C. F. Daganzo. On the design of public infrastructure systems with elastic demand. *Transportation Research Part B: Methodological*, 46(9):1288–1293, 2012.

[67] R. S. de Camargo and G. Miranda. Single allocation hub location problem under congestion: Network owner and user perspectives. *Expert Systems with Applications*, 39(3):3385–3391, 2012.

[68] R. S. de Camargo, G. Miranda, and H. P. Luna. Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers and Operations Research*, 35(4):1047–1064, 2008.

[69] R. Dionne and M. Florian. Exact and approximate algorithms for optimal network design. *Networks*, 9(1):37–59, 1979.

[70] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *Journal of the Association for Computing Machinery*, 19(2):248–264, Apr. 1972.

[71] H. A. Eiselt, G. Laporte, and J.-F. Thisse. Competitive location models: A framework and bibliography. *Transportation Science*, 27(1):44–54, 1993.

[72] S. Elhedhli and F. X. Hu. Hub-and-spoke network design with congestion. *Computers and Operations Research*, 32(6):1615–1632, 2005.

[73] S. Elhedhli and H. Wu. A Lagrangean heuristic for hub-and-spoke system design with capacity selection and congestion. *INFORMS Journal on Computing*, 22 (2):282–296, 2010.

[74] S. P. Evans. Derivation and analysis of some models for combining trip distribution and assignment. *Transportation Research*, 10(1):37–57, 1976.

[75] J. Fernández and E. M. Hendrix. Recent insights in Huff-like competitive facility location and design. *European Journal of Operational Research*, 227(3):581–584, 2013.

[76] M. Fischetti and A. Lodi. Local branching. *Mathematical Programming*, 98 (1-3):23–47, 2003.

[77] M. Fischetti and M. Monaci. Cutting plane versus compact formulations for uncertain (integer) linear programs. *Mathematical Programming Computation*, 4(3):239–273, 2012.

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

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

[80] M. Florian and S. Nguyen. A method for computing network equilibrium with elastic demands. *Transportation Science*, 8(4):321–332, 1974.

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

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

[83] A. S. Fotheringham and M. E. O'Kelly. *Spatial interaction models: formulations and applications*, volume 1. Kluwer academic publishers, Dordrecht, 1989.

[84] I. Fragkos, J.-F. Cordeau, and R. Jans. The multi-period multi-commodity network design problem. Technical Report CIRRELT 2017-63, Université de Montréal, 2017.

[85] A. Frangioni and B. Gendron. 0–1 reformulations of the multicommodity capacitated network design problem. *Discrete Applied Mathematics*, 157(6):1229–1241, 2009.

[86] A. Frangioni and B. Gendron. A stabilized structured Dantzig–Wolfe decomposition method. *Mathematical Programming*, 140(1):45–76, 2013.

[87] A. Frangioni and E. Gorgone. Bundle methods for sum-functions with "easy" components: applications to multicommodity network design. *Mathematical Programming*, 145(1):133–161, 2014.

[88] V. Gabrel, M. Lacroix, C. Murat, and N. Remli. Robust location transportation problems under uncertain demands. *Discrete Applied Mathematics*, 164(1):100–111, 2014.

[89] S. L. Gadegaard, A. Klose, and L. R. Nielsen. An improved cut-and-solve algorithm for the single-source capacitated facility location problem. *EURO Journal on Computational Optimization*, 6(1):1–27, 2018.

[90] B. Gendron, S. Hanafi, and R. Todosijević. Matheuristics based on iterative linear programming and slope scaling for multicommodity capacitated fixed

charge network design. *European Journal of Operational Research*, 268(1):70–81, 2018.

[91] A. M. Geoffrion and G. W. Graves. Multicommodity distribution system design by Benders decomposition. *Management Science*, 26(8):855–856, 1974.

[92] N. Ghaffari-Nasab, M. Ghazanfari, and E. Teimoury. Robust optimization approach to the design of hub-and-spoke networks. *The International Journal of Advanced Manufacturing Technology*, 76:1091–1110, 2015.

[93] I. Ghamlouche, T. G. Crainic, and M. Gendreau. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research*, 51(4):655–667, 2003.

[94] I. Ghamlouche, T. G. Crainic, and M. Gendreau. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Annals of Operations Research*, 131(1):109–133, Oct 2004.

[95] M. Goemans and J. Vondrák. *Stochastic Covering and Adaptivity*, pages 532–543. Springer, Berlin, Heidelberg, 2006.

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

[97] T. Grosche, F. Rothlauf, and A. Heinzl. Gravity models for airline passenger volume estimation. *Journal of Air Transport Management*, 13(4):175–183, 2007.

[98] O. Günlük. A branch-and-cut algorithm for capacitated network design problems. *Mathematical Programming*, 86(1):17–39, 1999.

[99] F. Habibzadeh Boukani, B. Farhang Moghaddam, and M. Pishvaee. Robust optimization approach to capacitated single and multiple allocation hub location problems. *Computational and Applied Mathematics*, 35:45–60, 2016.

[100] H. W. Hamacher, M. Labbé, S. Nickel, and T. Sonneborn. Adapting polyhedral properties from facility to hub location problems. *Discrete Applied Mathematics*, 145(1):104–116, 2004.

[101] K. Haynes and A. Fotheringham. *Gravity and Spatial Interaction Models*. Scientific Geography Series. SAGE Publications, Beverly Hills, CA, 1984.

[102] J. Hellstrand, T. Larsson, and A. Migdalas. A characterization of the uncapacitated network design polytope. *Operations Research Letters*, 12(3):159–163, 1992.

[103] M. Hewitt, G. L. Nemhauser, and M. W. P. Savelsbergh. Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing*, 22(2):314–325, 2010.

[104] M. J. Hodgson. A flow-capturing location-allocation model. *Geographical Analysis*, 22(3):270–279, 1990.

[105] K. Holmberg and J. Hellstrand. Solving the uncapacitated network design problem by a Lagrangean heuristic and branch-and-bound. *Operations Research*, 46 (2):247–259, 1998.

[106] D. L. Huff. Defining and estimating a trading area. *Journal of Marketing*, 28 (3):34–38, 1964.

[107] M. Jeihoonian, M. K. Zanjani, and M. Gendreau. Accelerating Benders decomposition for closed-loop supply chain network design: Case of used durable products with different quality levels. *European Journal of Operational Research*, 251(3):830–845, 2016.

[108] C. Jensen-Butler. Gravity models as planning tools: A review of theoretical

and operational problems. *Geografiska Annaler. Series B, Human Geography*, 54(1):68–78, 1972.

[109] D. S. Johnson, J. K. Lenstra, and A. H. G. R. Kan. The complexity of the network design problem. *Networks*, 8(4):279–285, 1978.

[110] N. Katayama, M. Chen, and M. Kubo. A capacity scaling heuristic for the multicommodity capacitated network design problem. *Journal of Computational and Applied Mathematics*, 232(1):90–101, 2009.

[111] E. Keyvanshokooh, S. M. Ryan, and E. Kabir. 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, 2016.

[112] D. Kim and P. M. Pardalos. A solution approach to the fixed charge network flow problem using a dynamic slope scaling procedure. *Operations Research Letters*, 24(4):195–203, 1999.

[113] D. Kim and P. M. Pardalos. Dynamic slope scaling and trust interval techniques for solving concave piecewise linear network flow problems. *Networks*, 35(3): 216–222, 2000.

[114] M. R. Kılınç and N. V. Sahinidis. Exploiting integrality in the global optimization of mixed-integer nonlinear programming problems with BARON. *Optimization Methods and Software*, 33(3):540–562, 2018.

[115] J. Kratica, D. Tošić, V. Filipović, and I. Ljubić. A genetic algorithm for the uncapacitated network design problem. In R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, and F. Hoffmann, editors, *Soft Computing and Industry: Recent Applications*, pages 329–336. Springer London, London, 2002.

[116] A. K. Kuiteing, P. Marcotte, and G. Savard. Network pricing of congestion-free networks: The elastic and linear demand case. *Transportation Science*, 51(3): 791–806, 2017.

[117] A. K. Kuiteing, P. Marcotte, and G. Savard. Pricing and revenue maximization over a multicommodity transportation network: the nonlinear demand case. *Computational Optimization and Applications*, page In Press, Aug 2018.

[118] M. Labbé, P. Marcotte, and G. Savard. A bilevel model of taxation and its application to optimal highway pricing. *Management Science*, 44(12-part-1): 1608–1622, 1998.

[119] B. W. Lamar, Y. Sheffi, and W. B. Powell. A capacity improvement lower bound for fixed charge network design problems. *Operations Research*, 38(4): 704–710, 1990.

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

[121] C. Lee, K. Lee, and S. Park. Benders decomposition approach for the robust network design problem with flow bifurcations. *Networks*, 62(1):1–16, 2013.

[122] J. P. Lesage and W. Polasek. Incorporating transportation network structure in spatial econometric models of commodity flows. *Spatial Economic Analysis*, 3(2):225–245, 2008.

[123] Á. Lorca, X. A. Sun, E. Litvinov, and T. Zheng. Multistage adaptive robust optimization for the unit commitment problem. *Operations Research*, 64(1): 32–51, 2016.

[124] M. Los and C. Lardinois. Combinatorial programming, statistical optimization and the optimal transportation network problem. *Transportation Research Part B: Methodological*, 16(2):89–124, 1982.

[125] F. Louveaux. Stochastic location analysis. *Location Science*, 1:127 –154, 1993.

[126] T. Magnanti, P. Mireault, and R. Wong. Tailoring Benders decomposition for uncapacitated network design. In G. Gallo and C. Sandi, editors, *Network Flow at Pisa*, Mathematical Programming Studies, volume 26, pages 112–154. 1986.

[127] T. L. Magnanti and R. T. Wong. Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3): 464–484, 1981.

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

[129] A. I. Mahmutogullari and B. Y. Kara. Hub location under competition. *European Journal of Operational Research*, 250(1):214–225, 2016.

[130] V. Marianov and D. Serra. Location models for airline hubs behaving as M/ D/ c queues. *Computers & Operations Research*, 30(7):983–1003, 2003.

[131] V. Marianov, Rios, Miguel, and Barros, Francisco Javier. Allocating servers to facilities, when demand is elastic to travel and waiting times. *RAIRO-Oper. Res.*, 39(3):143–162, 2005.

[132] V. Marianov, M. Ríos, and M. J. Icaza. Facility location for market capture when users rank facilities by shorter travel and waiting times. *European Journal of Operational Research*, 191(1):32–44, 2008.

[133] A. Marín, L. Canovas, and M. Landete. New formulations for the uncapacitated

multiple allocation hub location problem. *European Journal of Operational Research*, 172:274–292, 2006.

[134] E. Martins de Sá, I. Contreras, and J.-F. Cordeau. Exact and heuristic algorithms for the design of hub networks with multiple lines. *European Journal of Operational Research*, 246:186–198, 2015.

[135] E. C. Matsoukis and P. C. Michalopoulos. Road traffic assignment—a review. *Transportation Planning and Technology*, 11(2):117–135, 1986.

[136] M. Merakli and H. Yaman. Robust intermodal hub location under polyhedral demand uncertainty. *Transportation Research Part B: Methodological*, 86:66–85, 2016.

[137] C. E. Miller, A. W. Tucker, and R. A. Zemlin. Integer programming formulation of traveling salesman problems. *Journal of the Association for Computing Machinery*, 7(4):326–329, 1960.

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

[139] L.-M. Munguía, S. Ahmed, D. A. Bader, G. L. Nemhauser, V. Goel, and Y. Shao. A parallel local search framework for the fixed-charge multicommodity network flow problem. *Computers & Operations Research*, 77:44–57, 2017.

[140] J. Naoum-Sawaya and S. Elhedhli. An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, 210(1):33–55, Nov 2013.

[141] G. F. Newell. Some issues relating to the optimal design of bus routes. *Transportation Science*, 13(1):20–35, 1979.

[142] I. Newton. *Philosophiae Naturalis Principia Mathematica*. Londini :Jussu Societatis Regiae ac Typis Josephi Streater, 1687.

[143] M. E. O'Kelly. The location of interacting hub facilities. *Transportation Science*, 20(2):92–106, 1986.

[144] M. E. O'Kelly, J. F. Campbell, R. S. de Camargo, and G. Miranda. Multiple allocation hub location model with fixed arc costs. *Geographical Analysis*, 47 (1):73—96, 2015.

[145] M. E. O'Kelly, H. P. L. Luna, R. S. de Camargo, and G. Miranda. Hub location problems with price sensitive demands. *Networks and Spatial Economics*, 15 (4):917–945, 2015.

[146] F. Ortega and L. Wolsey. A branch-and-cut algorithm for the single-commodity, uncapacitated, fixed-charge network flow problem. *Networks*, 41(3):143–158, 2003.

[147] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte. An exact algorithm for multi-level uncapacitated facility location. Forthcoming in Transportation Science. 2017.

[148] M. Padberg and G. Rinaldi. A branch-and-cut algorithm for the resolution of large-scale symmetric traveling salesman problems. *SIAM Review*, 33(1): 60–100, 1991.

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

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

[151] D. C. Paraskevopoulos, T. Bektaş, T. G. Crainic, and C. N. Potts. A cycle-based evolutionary algorithm for the fixed-charge capacitated multi-commodity network design problem. *European Journal of Operational Research*, 253(2): 265–279, 2016.

[152] S. Peeta and A. K. Ziliaskopoulos. Foundations of dynamic traffic assignment: The past, the present and the future. *Networks and Spatial Economics*, 1(3): 233–265, Sep 2001.

[153] C. Raack, A. M. Koster, S. Orlowski, and R. Wessäly. On cut-based inequalities for capacitated network design polyhedra. *Networks*, 57(2):141–156, 2011.

[154] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: A literature review. *European Journal of Operational Research*, 259(3):801–817, 2017.

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

[156] C. Randazzo and H. Luna. A comparison of optimal methods for local access uncapacitated network design. *Annals of Operations Research*, 106(1-4):263–286, 2001.

[157] E. G. Ravenstein. The laws of migration. *Journal of the Statistical Society of London*, 48(2):167–235, 1885.

[158] W. Rei, J.-F. Cordeau, M. Gendreau, and P. Soriano. Accelerating Benders decomposition by local branching. *INFORMS Journal on Computing*, 21(2): 333–345, 2009.

[159] W. J. Reilly. *Methods for the Study of Retail Relationships.* [Austin, Tex.] The University, 1929.

[160] I. Rodríguez-Martín and J. J. Salazar-González. A local branching heuristic for the capacitated fixed-charge network design problem. *Computers & Operations Research*, 37(3):575–581, 2010.

[161] N. V. Sahinidis. *BARON 17.8.9: Global Optimization of Mixed-Integer Nonlinear Programs,* User's Manual, 2017.

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

[163] M. L. Senior. From gravity modelling to entropy maximizing: a pedagogic guide. *Progress in Geography*, 3(2):175–210, 1979.

[164] E. Z. Serper and S. A. Alumur. The design of capacitated intermodal hub networks with different vehicle types. *Transportation Research Part B: Methodological*, 86:51–65, 2016.

[165] M. Shahabi and A. Unnikrishnan. Robust hub network design problem. *Transportation Research Part E: Logistics and Transportation Review*, 70:356–373, 2014.

[166] T. Sim, T. J. Lowe, and B. W. Thomas. The stochastic p-hub center problem with service-level constraints. *Computers & Operations Research*, 36(12):3166–3177, 2009.

[167] L. Snyder. Facility location under uncertainty: A review. *IIE Transactions*, 38: 537–554, 2006.

[168] Z. Sun and J. Zheng. Finding potential hub locations for liner shipping. *Transportation Research Part B: Methodological*, 93:750–761, 2016.

[169] M. Tanash, I. Contreras, and N. Vidyarthi. An exact algorithm for the modular hub location problem with single assignments. *Computers & Operations Research*, 85:32–44, 2017.

[170] M. Tawarmalani and N. V. Sahinidis. A polyhedral branch-and-cut approach to global optimization. *Mathematical Programming*, 103:225–249, 2005.

[171] W. van Ackooij, A. Frangioni, and W. de Oliveira. Inexact stabilized Benders' decomposition approaches with application to chance-constrained problems with finite support. *Computational Optimization and Applications*, 65(3): 637–669, Dec 2016.

[172] A. G. Wilson. A statistical theory of spatial distribution models. *Transportation Research*, 1(3):253–269, 1967.

[173] A. G. Wilson. A family of spatial interaction models, and associated developments. *Environment and Planning A: Economy and Space*, 3(1):1–32, 1971.

[174] B. Yaged. Minimum cost routing for static network models. *Networks*, 1(2): 139–172, 1971.

[175] M. Yaghini, M. Karimi, M. Rahbar, and M. H. Sharifitabar. A cutting-plane neighborhood structure for fixed-charge capacitated multicommodity network design problem. *INFORMS Journal on Computing*, 27(1):48–58, 2015.

[176] T.-H. Yang. Stochastic air freight hub location and flight routes planning. *Applied Mathematical Modelling*, 33(12):4424–4430, 2009.

[177] Z. Yang, F. Chu, and H. Chen. A cut-and-solve based algorithm for the single-source capacitated facility location problem. *European Journal of Operational Research*, 221(3):521–532, 2012.

[178] R. Zanjirani Farahani, M. Hekmatfar, A. B. Arabani, and E. Nikbakhsh. Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering*, 64(4):1096–1109, 2013.

[179] N. Zarrinpoor, M. S. Fallahnezhad, and M. S. Pishvaee. The design of a reliable and robust hierarchical health service network using an accelerated Benders decomposition algorithm. *European Journal of Operational Research*, 265(3): 1013–1032, 2018.

[180] B. Zeng and L. Zhao. Solving two-stage robust optimization problems using a column-and-constraint generation method. *Operations Research Letters*, 41(5): 457–461, 2013.

[181] C. A. Zetina, I. Contreras, and J.-F. Cordeau. Exact algorithms for the multi-commodity uncapacitated fixed-charge network design problem. Submitted to Computers & Operations Research. 2018.

[182] C. A. Zetina, I. Contreras, E. Fernández, and C. Luna-Mota. Solving the optimum communication spanning tree problem. *European Journal of Operational Research*, 273(1):108–117, 2019.

[183] Y. Zhang, O. Berman, and V. Verter. Incorporating congestion in preventive healthcare facility network design. *European Journal of Operational Research*, 198(3):922–935, 2009.

[184] Y. Zhang, O. Berman, P. Marcotte, and V. Verter. A bilevel model for preventive healthcare facility network design with congestion. *IIE Transactions*, 42 (12):865–880, 2010.

[185] H. Zhong, J. Wang, T. L. Yip, and Y. Gu. An innovative gravity-based approach to assess vulnerability of a hazmat road transportation network: A case study of Guangzhou, China. *Transportation Research Part D: Transport and Environment*, 62:659–671, 2018.