HUB NETWORK DESIGN PROBLEMS WITH PROFITS

Armaghan Alibeyg

A thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy

Concordia University

Montréal, Québec, Canada

March 2017

Concordia University

School of Graduate Studies

This is to certify that the thesis prepared

By: **Miss. Armaghan Alibeyg**

Entitled: **Hub Network Design Problems with Profits**

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

Doctor of Philosophy (Industrial Engineering)

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

Signed by the final examining commitee:

Dr. Chair

Dr. Vladimir Marianov External Examiner

Dr. Masoumeh Kazemi Zanjani Examiner

Dr. Mingyuan Chen Examiner

Dr. Navneet Vidyarthi Examiner

Dr. Ivan Contreras Supervisor

Dr. Elena Fernandéz Co-supervisor

Approved ______________________________

Chair of Department or Graduate Program Director

February 2017 ______________________________

Dr. ,Dean

Faculty of Engineering and Computer Science

Abstract

Hub Network Design Problems with Profits

Armaghan Alibeyg, Ph.D.

Concordia University, 2017

In this thesis we study a new class of hub location problems denoted as *hub network design problems with profits* which share the same feature: a profit oriented objective. We start from a basic model in which only routing and location decisions are involved. We then investigate more realistic models by incorporating new elements such as different types of network design decisions, service commitments constraints, multiple demand levels, multiple capacity levels and pricing decisions. We present mixed-integer programming formulations for each variant and extension and provide insightful computational analyses regarding to their complexity, network topologies and their added value compared to related hub location problems in the literature. Furthermore, we present an exact algorithmic framework to solve two variants of this class of problems. We continue this study by introducing joint hub location and pricing problems in which pricing decisions are incorporated into the decision-making process. We formulate this problem as a mixed-integer bilevel problem and provide feasible solutions using two math-heuristics. The dissertation ends with some conclusions and comments on avenues of future research.

To my beloved grandmas, parents and siblings

Mamani, MamanAzar, Ahmad, Fatemeh, Ali, Assal and AmirAshkan

Acknowledgments

In the name of God, most Gracious, most Merciful...

I would like to express my special thanks to my supervisor, Dr. Ivan Contreras, for his guidance and all the insightful comments and full time enrollment. He was always willing to help me and I was not able to finish this journey without his support. Moreover, I would like to thank my co-supervisor, Dr. Elena Fernández from department of statistics and operations research of Universitat Politècnica de Catalunya, Barcelona, Spain, for her deep knowledge and experience that shared with me. The original idea of this dissertation is coming from Elena and her deep insights helped me at various stages of my research. She was always willing to support me even from long distance.

I am also very thankful to Dr. Martine Labbé from Université Libre de Bruxelles, Brussels, for her contribution to this dissertation and I feel deeply honored for this collaboration. In addition, I am very grateful to the administrative staff of the department of Mechanical and Industrial Engineering.

Special thanks to my lovely family. Words cannot explain how grateful I am from my parents, Ahmad Alibeyg and Fatemeh Rafiei, my grand-moms and my siblings Ali, Assal and AmirAshkan for all their patience, support and unlimited love. There were many times I wanted to give up and come back to them but they always encouraged me to continue although it was very hard for them being far. I also thank my friends Jair Ferari, César Roriguez, Moayad Tanash, Omar Abuobidalla, Mohammad Jeihoonian, and Dua Waraikat for all their support and fun we had together.

Contents

List of Figures

List of Tables

Abbreviations

AP	Australian Post
BB	Branch-and-Bound
CAB	Civil Aeronautics Board
CHLP	Competitive Hub Location Problem
GNDP	General Network Design Problem
HALP	Hub Arc Location Problem
HLP	Hub Location Problem
HLPP	Hub Location and Pricing Problem
HLPP-A	Hub Location and Pricing Problem with Pricing on Arcs
HLPP-P	Hub Location and Pricing Problem with Pricing on Paths
HNDPP	Hub Network Design Problem with Profits
LP	Linear Programming
LR	Lagrangean Relaxation
MHCP	Maximal Hub Covering Problem
MIP	Mixed-integer Programming
O/D	Origin/Destination
PE	Partial Enumeration

QBP	Quadratic Boolean Problem
RT	Reduction Test
UF	User Facility
UFLP	Uncapacitated Facility Location Problem
UHLPP	Uncapacitated Hub Location Problem with Profits
UHLPMA	Uncapacitated Hub Location Problem Multiple Allocation
UNLP	Hub Node Location Problem
UU	User User
VNS	Variable Neighborhood Search

Chapter 1

Introduction

1.1 Overview

Hub-and-spoke networks are frequently employed in transportation and telecommunication systems to efficiently route commodities between many origins and destinations. One of the key features of these networks is that direct connections between *origin/destination* (O/D) pairs can be replaced by fewer, indirect but privileged connections by using transshipment, consolidation, or sorting points, called *hub facilities*. This reduces the total setup cost at the expense of increasing some individual transportation costs. Overall transportation costs may also decrease due to the bundling or consolidating of flows through inter-hub arcs.

Hub location problems (HLPs) deal with joint location and network design decisions so as to optimize a cost-based (or service-based) objective. The location decisions focus on the selection of a set of nodes to place hub facilities, whereas the network design decisions deal with the selection of the links to connect origins and destinations, possibly via hubs, as well as the routing of commodities through the network. Typically, HLPs assume that hubs must be located at the nodes of a given network, distances satisfy the triangle inequality, and there is a constant discount factor on the transportation costs of the arcs connecting hubs. In addition, classical HLPs impose that all flows are routed via the selected hubs and ignore all arc setup costs. Such problems have optimal solutions where an arc exists connecting each pair of hubs, so optimal routing paths consist of at most three arcs, two arcs connecting non-hub nodes and hub nodes, plus one intermediate arc connecting two hub nodes. This optimality condition implies that the network design decisions are mainly determined by the allocation of non-hub nodes to hubs (see Contreras [31]), and has been extensively exploited to develop formulations and solution algorithms for solving these classical HLPs.

Hub arc location problems (HALPs) no longer assume that the above optimality condition holds, and incorporate explicit hub arc selection decisions. HALPs, in which a cardinality constraint on the number of opened hub arcs is considered, were introduced in Campbell et al. [24]. HALPs that incorporate setup costs for the hub nodes and hubs arcs were studied in Contreras and Fernández [38] and Gelareh et al. [58].

In most hub location applications arising in the design of distribution and transportation systems, a profit is obtained for serving (i.e. routing) the demand of a given commodity. Capturing such profit may incur not only a routing cost but also additional setup costs, as the O/D nodes of the commodity may require the a priori installation of transport infrastructure. Classical HLPs and HALPs, however, ignore such profits and associated setup costs, as reflected by the requirement that the demand of every commodity must be served. Indeed, the overall profit obtained when all the commodities must be served is constant, and it does not affect the optimization of the distribution system. Broadly speaking, this requirement expresses the implicit hypothesis that the overall costs of solution networks will be compensated by the overall profits. Of course, such hypothesis does not necessarily hold, and incorporating decisions on the O/D nodes that should be served and their associated commodities may have important implications in the strategic and operational costs.

1.2 Hub Network Design Problems with Profits

In this dissertation we study a new class of problems in hub location denoted as *hub network design problems with profits* (HNDPPs). HNDPPs integrate within the decision-making process additional strategic decisions on the nodes and the commodities that have to be served and consider a profit-oriented objective which measures the tradeoff between the profit of the commodities that are served and the overall network design and transportation costs. Broadly speaking, HNDPPs focus on the following strategic decisions: *i)* where to locate the hubs; *ii)* what edges to activate and of what type; and, *iii)* what commodities to serve (this also dictates the nodes to activate). As usual, the operational decisions determine how to route the commodities that are selected to be served. HNDPPs generalize HLPs and HALPs as they incorporate one additional level to the decision-making process. To the best of our knowledge, hub location models incorporating explicit decisions on the nodes to be served have not yet been addressed in the literature.

Potential transportation applications of HNDPPs arise in the airline and ground transportation industries. As an example, in the case of airline companies network planners have to design their transportation network when they are first entering into the market, or may have to modify already established hub-and-spoke networks through alliances, merges and acquisitions of companies. The involved decisions are to determine the cities that will be part of their network, i.e. what cities they will provide service to (served nodes) and what O/D flights to activate (served commodities) in order to offer air travel services to passengers (served demand) between city pairs. Additional decisions focus on the location of their main airports (hub facilities) and on the selection of the legs used for connecting regional airports (served nodes) with hub airports and for connecting some hub airports between them. Finally, the transportation of passengers using one or more O/D paths on their established network. The objective is to find an optimal hub network structure that maximizes the total net profit for providing air travel services to a set of O/D flights while taking into account the (re)design cost of the network. Depending on the regulations or the company service policy, passenger air travel services could be provided: *i)* only to city pairs that are profitable, *ii)* between all city pairs that are served by the company, or *iii)* to a percentage of them (private companies with service commitment or with market penetration policies).

The first contribution of this thesis is to study the most basic variants of HNDPP denoted as the *uncapacitated hub location problems with profit* (UHLPP) that concentrates on the following strategic decisions: *i)* which nodes should be served by the hub network, *ii)* where the hub facilities should be located among these selected nodes and, *iii)* which commodities are profitable to be routed and through which path. That is, the UHLPP focuses mainly on the location decisions and disregards any link activation decisions. The UHLPP considers a profit oriented objective in which the aim is to maximize the difference between the total revenue obtained from the routing of flows, which depends on the design of the hub network, and set-up costs for designing it. To the best of our knowledge, this is the first time in the literature of HLPs that servicing node decisions are incorporated. This makes UHLPP more challenging than classical HLPs since for routing commodities through the network, both their associated origin and destination nodes must be served. Moreover, some commodities may not be routed if not profitable even if their associated O/D nodes are served. We propose a MIP formulation for UHLPP based on the formulation of

Hamacher et al. [65] for UHLPMA. We also present lagrangean relaxation heuristic that uses a lagrangean function that can be decomposed into subproblems which can be solved efficiently.

The second contribution of this dissertation is to propose and introduce the foundations of alternative HNDPPs of increasing complexity, which incorporate additional features. We start with a pure profit-driven model, and progressively present variations which consider alternative constraints and/or additional decisions, which, in turn, may imply additional costs. The first one, denoted as PO_1, is flexible in the sense that among all commodities associated with served O/D nodes, only those that are actually profitable are indeed routed. The second model, denoted as PO_2, considers a more restrictive scenario in which no commitments or regulations exist and thus, all commodities whose O/D nodes are both activated would have to be served, even if this would reduce the total profit. For decisions we consider the activation, with associated setup costs, of two additional types of edges. Access edges allow non-hub nodes to be connected to a hub whereas bridge edges allow connecting hub nodes without using a discount factor. We further introduce an extension of the primary HNDPPs that incorporates link activation decisions on access and bridge edges denoted as PND. We also consider two more general models that allow serving the existing demand at different levels denoted as POM_1 and POM_2. The second of such models allows, in addition, to activate at different levels the various elements of solution networks. This results in a capacitated HNDPP of notable difficulty. Mathematical programming formulations for these models are presented and computationally tested in terms of: the structure of the solution networks it produces, its sensitivity to the input parameters, its relation to the other models, and its difficulty for being optimally solved with a commercial solver.

In the third part of this dissertation we focus on methodological aspects leading to the exact solution of two primary profit-oriented HNDPPs mentioned above. We propose a unified algorithmic framework applicable to large-scale instances of both PO_1 and PO_2 models involving up to 100 nodes. It is an exact *branch-and-bound* (BB) procedure in which a sophisticated lagrangean relaxation is used to obtain tight bounds at each node of the enumeration tree. In particular, the proposed lagrangean function resorts to the solution of well-known *quadratic boolean problems* (QBPs). We show how, due to the special cost structure associated with the quadratic term of the objective function, the QBPs can be efficiently solved by transforming them to classical *minimum cut problems*. The algorithm is enhanced through several algorithmic refinements that make it more efficient. These include: (i) variable elimination techniques that allow reducing considerably the size of the formulations at the root node, (ii) a *partial enumeration* (PE) phase capable of effectively exploring the solution space by reducing the required number of nodes in the tree, and (iii) the use of simple but effective primal heuristics embedded in the subgradient algorithm that exploit the structure of the problem. Computational experiments confirm the effectiveness of our exact algorithmic framework since it is able to obtain optimal solutions for instances with up to 100 nodes

Finally, in the last part of this dissertation we study joint hub location and pricing problems that we formulate as a mixed-integer bilevel programs. To maximize the profit, which is the difference between the revenue and costs, firms need to integrate within their pricing decisions, the rational reaction of customers to the price. In other words, although firms want to maximize their revenue, customers want to pick the service with minimum cost. For instance in airline transportation, pricing decisions have become very important since there is an intense competition between firms to capture the customers' demand. The objective of the study of this extension is to be able to show this complex decision making process in which there is a leader (firm) that needs to identify the price that is high enough to maximize its profit and low enough so that minimizes the cost of the follower (the customer). If the firm sets the price higher than the competitor, the customer will choose the competitor. *hub location and pricing problems* (HLPPs) involve two decision levels: The upper level is concerned with maximizing the difference between the leader's revenue obtained from prices set on the arcs (or commodities) and the routing and setup costs of opening hub facilities, while in the lower level the follower reacts by choosing a route with minimum price. We study two variants of HLPPs: pricing on arcs and pricing on commodities. We propose a mixed-integer bilevel programing formulation for each variant. We develop a math-heuristic to solve HLPPs with pricing on arcs.

A summary of the proposed models and their main features is given in Table 1.1.

Main features	$UHLPP$	PO_1	PO_2	PND	POM_1	POM_2	$HLPP$
Locational decisions							
- Hub nodes	√	√	√	√	√	√	√
- Served nodes	√	√	√	√	√	√	
- Multiple capacity/operational levels						√	
Link activation decisions							
- Hub edges		√	√	√	√	√	
- Access edges				√			
- Bridge edges				√			
Operational decisions							
- Single demand level	√	√	√	√			√
- Multiple demand levels					√	√	
Demand service type							
- Only to profitable pairs of served nodes	√	√		√	√	√	√
- Between all pairs of served nodes			√				
Pricing Decisions							
- Given as an input of the problem	√	√	√	√	√	√	
- Part of the decision process							√
Objective							
- Pure profit-oriented	√	√		√	√	√	√
- Profit-oriented with service commitments			√				

Table 1.1: Summary of considered HNDPPs.

1.3 Scope and Objectives

The main objective of this dissertation is to introduce and study a new challenging class of HLPs that considers a profit oriented perspective. The specific objectives of this research are summarized as follows:

1. To introduce a class of hub location problems where it is not necessary to provide service to all demand nodes. A profit is associated with each flow between pair of nodes. We present a *mixed-integer programming* (MIP) formulation and a lagrangean relaxation algorithm to solve this problem.

2. To introduce a class of hub network design problems with a profit-oriented objective. These problems relax different unrealistic assumptions usually considered in classical HLPs such as fully connected networks and serving all the nodes. We propose and analyze alternative models and integer programming formulations.

3. To develop an exact algorithmic framework for HNDPPs where a sophisticated lagrangean function that exploits the structure of the problems is used to efficiently obtain bounds at the nodes of an enumeration tree.

4. To introduce hub location problems which consider joint design and pricing decisions. These problems are formulated as mixed-integer bilevel programs where the first level maximizes the profit of the leader by selecting a set of hubs and prices that minimizes the routing costs of the follower in the second level. We obtain feasible solutions for these complex problems by using two variants of a math-heuristic.

1.4 Organization of the Thesis

This manuscript is organized as follows. Chapter 2 reviews the most relevant literature related to HNDPPs and HLPPs. Chapter 3 presents the formal definition, modeling assumption, mathematical formulation and a lagrangean relaxation of UHLPPs. Chapter 4 introduces several variants of HNDPPs and their mathematical programming formulations of each variant as well as some computational experiments to compare the complexity of each problem using a general solver. Chapter 5 presents the lagrangean relaxation-based branch and bound applied to solve PO_1 and PO_2 together with some computational experiments that compares the result of the exact method with a general purpose solver. Chapter 6 introduces the formal definition and the mixed-integer bilevel program of HLPPs and some computational experiments obtained from a math-heuristic. The manuscript ends with conclusions and future research directions in Chapter 7.

Chapter 2

Literature Review

In this Chapter we review the most relevant studies related to this thesis. More specifically, in Section 2.1 we first provide a brief introduction about *general network design problems* (GNDPs). In Section 2.2, we explain HLPs in general, and the more related HLPs to HNDPP including uncapacitated hub location problems with multiple allocation, hub covering problems, hub arc location problems and *competitive hub location problems* (CHLP). In each case we highlight the main differences between the referred problem and HNDPPs. Also, a brief literature review of other network optimization problems with profits is provided in Section 2.3. The chapter ends with a short overview on bilevel programming in Section 2.4, which is related to Chapter 6 of this manuscript.

2.1 General Network Design Problems

Facility location and network design problems are the two classes of $\mathcal{NP}$-hard problems which lie at the heart of network optimization. Facility location refers to siting facilities at nodes while network design problems are about opening links to connect facilities. General network design problems provide a unified view of combined facility location and network design problems (Contreras and Fernández [37]). This class of problems involves two types of decisions: design decisions, which refer to the location of facilities and opening links to connect facilities and operational decisions, which deal with the allocation of customers to facilities and with the routing of their demands.

The p-median location problem is one of the classical discrete location problems, which consists of locating p facilities on a given network (i.e. nodes or arcs) and allocating a set of customers to these facilities with the goal of minimizing the transportation cost. Hakimi [64] showed that there exists at least one optimal solution in which all p facilities are located only at nodes of the network. This is known as the *Hakimi* property. Another classical facility location problem is the *uncapacitated facility location problem* (UFLP), in which there exists a fixed cost for locating each facility and thus the number of facilities to be opened is unknown (Kuehn and Hamburger [77]). The p-center problem (Hakimi [64]), the location-covering problem (Toregas et al. [116]), and the maximum covering location problem (Church and ReVelle [28]), are other types of facility location problems that involve design decisions. A recent review of facility location problems can be found in (Smith et al. [115] and Laporte et al. [83]).

Network design problems consist of design decisions, which means that the network is not given and must be constructed, by choosing some elements from a given underlying network. Indeed, these types of problems involve selecting links to be activated for connecting users and facilities and facilities among them. They can be divided into two main categories: single-commodity and multiple-commodity. Single-commodity network design problems deal with one type of flow while multiple-commodity problems are concerned with networks dealing with different types of flow, which makes them more complicated than single commodity network design problems. Well-known examples of single commodity problems involving design decisions are fixed-charge network design problems, which consider both set-up and flow costs (Magnanti and Wong [88]; Minoux [96]), and network loading problems, which consider only set-up costs for the arcs (Magnanti et al. [87]). Some well-known multi-commodity network design problems are the capacitated multi-commodity network design problem (Gendron et al. [60]), and the hub-and-spoke network design problem (O'Kelly [99]). The interested reader is referred to up-to-date references of network design problems like Croxton et al. [43] and Frangioni and Gendron [56].

Some types of network optimization problems have both location decisions and network design decisions. For instance, location vehicle routing problems (Nagy and Salhi [98], Drexl and Schneider [49]) in which it is required to locate a set of depots and to design routes to visit customers. Also, the minimum cost Steiner tree problem in which it is required to locate the Steiner nodes and to connect the edges of the tree (Winter [118]).

In some classical facility location problems such as p-median and p-center problems, allocation decisions have to be made to indicate the facility that should serve each customer. In some problems, referred to as single allocation, each customer should be served by only one facility while in some others, referred to as multiple allocation, a customer may be assigned to more than one facility. Routes are the paths that are used to deliver user's demands so routing decisions are concerned with designing the routes for sending flows between pairs of nodes. The links that define the routing paths can be seen as the outcome of some network design decisions, so routing and network design problems are closely related. Transportation problems (Monge [97]; Kantorovich [74]) and network flow problems (Ahuja et al. [3]), are the two well-known network optimization problems with routing decisions.

Considering the different types of the decisions in the GNDPs mentioned above, location and allocation decisions identify the set of open facilities and the assignment pattern of customers to facilities. However, in the case of network design the goal of the routing decisions is to find the links to be activated and through which demand will be routed. Thus, these decisions strongly depend on the role of the facilities and type of service demand required by customers. Contreras and Fernández [37] propose a classification of GNDPs based on the type of the demand.

In GNDPs with *user-facility* (UF) demand service is offered at or from facilities and demand is routed between facilities and users. When each user goes to the facility, the goal is to find a path with minimum cost. When servers travel from the facilities to possibly provide service to several users, routing decisions may be involved. If the set-up cost for activating the links is zero, the network design decisions are trivial and will activate all the links. In such a case, the optimal routing decisions simply consist of allocating each user directly to facilities. However, when considering set-up costs the problem will be a GNDP with UF demand involving nontrivial network design decisions. Location-network design problems (Melkote and Daskin [95]; Contreras et al. [41]) and location-vehicle routing problems (Nagy and Salhi [98]) are two common examples of this kind of problems.

In GNDPs with *user-user* (UU) demand, demand relates pairs of users to user by means of facilities. In this type of problems, facilities play the role of providing connection between pairs of users. So network design and routing decisions in GNDPs with UU demand focus on connecting both users and facilities and facilities among them. If there is no fixed cost for opening facilities, like in classical network design problems, location decisions will be trivial because all the facilities will be opened. Well-known examples of GNDPs with UU demand are classical hub location problems (O'Kelly [99]; O'Kelly [101]; Campbell [21]), in which the routing decisions involve the selection of paths for sending commodities through the network. Concentrator location problems (Yaman [119]; Labbé and Yaman [81]; Gouveia and Saldanha-da Gama [62]), tree-star location problems (Contreras et al. [39]; Contreras et al. [40]), and cycle-star location problems (Labbe et al. [78]) are other examples of GNDPs with UU demand and non-trivial routing decisions. This review focuses on GNDPs with UU demand and, in particular, on HLPs.

As mentioned, in GNDPs with UU demand facilities are used as intermediate locations to consolidate or re-route flows between the users. These problems can be classified according to the topological structure induced by the facilities and the edges of the network connecting them. They can be classified as follows: (i) cliques, i.e. fully interconnected facilities, (ii) trees, i.e. connecting facilities by a tree, (iii) cycles, i.e. the edges connecting the facilities form a cycle. Hub location problems (HLPs) are the best-known GNDPs with UU demand, which commonly lie on the first category (fully interconnected facilities). HLPs deal with location decisions for selecting which facilities to open and with network design decisions for selecting which links to route the demand through the network.

2.2 Hub Location Problems

Hub-and-spoke networks are used in many distribution systems such as airline passenger, trucking and postal delivery networks. The key feature of these networks is that direct connections between all spokes are replaced with fewer, indirect connections by using transshipment facilities called hubs. Hub-and-spoke networks reduce the total set-up cost of the network but may increase individual travel times. However, transportation costs may decrease due to the bundling or consolidating of flows on the interhub links.

HLPs deal with the location of hubs, the design of the hub network, and the routing of flows from origins to destinations so as to optimize a considered objective function. The hubs can provide two major functions: *i)* a switching, sorting, or connecting (SSC) function, and *ii)* a consolidation/break-bulk (CB) function (Campbell et al. [25]). As a switching point (SSC function), a hub redirects the flow to enter via one link and depart via another. So, instead of using direct links from every origin to every destination, by redirecting flows based on their destination, all the nodes are connected with fewer links. The CB function aggregates or disaggregates the flows by combining many small separate flows into larger ones or splitting a large flow into separate smaller ones.

HLPs are known to be $\mathcal{NP}$-hard due to the inclusion of elements from facility location and network design which are the main elements of a HLP. The first papers in hub location are probably by Goldman [61] and O'Kelly [100], adapting the node optimality property of Hakimi [64] to the hub median problem. Early reviews of HLPs are Campbell [22], and O'Kelly and Miller [105]. Campbell [22] provide a comprehensive review on network hub location problems, classification of fundamental problems and formulations, and a wide variety of applications of HLPs. O'Kelly and Miller [105] focus on network design decisions in hub location and introduce different possible topological structures for hub-and-spoke networks. Bryan and O'Kelly [20] review hub location research in air transportation applications. More recent reviews of HLPs are Campbell et al. [25], Alumur and Kara [7], Kara and Taner [75], Campbell and O'Kelly [26], Farahani et al. [53] and Contreras [31]. Of particular interest is the work of Campbell and O'Kelly [26], which provides insights into early motivations in hub location research and highlights the most recent and promising research directions.

2.2.1 Features, Assumptions and Properties

The main assumptions made in most HLPs are as follows: *i)* distance satisfy triangle inequality, *ii)* there is no set-up costs on the arcs, *iii)* there is an obligation of routing flows via a set of hubs, *iv)* flows are discounted on the hub arcs to reflect economies of scale. The discount factor is used to provide reduced flow costs on hub arcs due to the consolidation of flows between hubs. i.e., the per unit transportation cost is discounted if the flow passes via a hub arc.

As a consequence of the above assumptions, important properties arise in most HLPs. Since flows are forced to be routed via hubs (assumption (*iii*)), no direct connection is allowed and this means that the OD paths consist of at least one hub node. Also assumption (*i*) together with (*ii*) implies that OD paths contains at most two hubs. However, whenever assumption (*i*) does not satisfied, paths may contain more than two hubs and more than one hub arc.

A hub-and-spoke network generally consists of three type of arcs: *hub arcs* connecting two hub facilities, *access arcs* connecting non-hub nodes to hubs, *bridge arcs* connecting two hub nodes without reduced unit transportation cost of a hub arc. To clarify the idea of using each type of arcs, a more detailed look of O/D paths is provided. An O/D path contains three main legs in general: the collection leg (from the origin to the first hub), the transfer leg (between hubs), and the distribution leg (from the last hub to the destination). The collection and distribution legs contain only one access arc because of the triangle inequality. The transfer leg, if it exists, may consists of several bridge and hub arcs, depending on additional assumptions made on the model. However, not all O/D paths contain a transfer leg (origin-hub-destination). Considering the paths that the origin or destination is a hub node, a bridge arc can occur as the first (or the last) arc in the path or between any two hubs in other paths. We refer to Campbell et al. [24] for more information regarding to the analysis of O/D paths.

2.2.2 The Uncapacitated Hub Location Problem with Multiple Assignments

One of the most fundamental and intensively studied HLPs is the *uncapacitated hub location problem with multiple assignments* (UHLPMA). The main decisions in UHLPs are the location of hub facilities and the routing of commodities through the hub network with the objective of minimizing the total set-up and transportation cost. Several MIP formulations (Campbell [21], Hamacher et al. [65], Marín et al. [92]) and solution algorithms (Contreras et al. [32], de Camargo et al. [44], Klincewicz [76]) have been introduced for the UHLPMA. Various extensions of the UHLPMA have also been studied, such as capacitated models (Ebery et al. [50], Marín [90]), stochastic models (Alumur et al. [8], Contreras et al. [34]), and dynamic models (Contreras et al. [33]).

We can formally define the UHLPMA as follows. Let $G = (N, A)$ be a directed complete graph, where $N=\{1, 2, ..., n\}$ represents the set of potential nodes to provide service and A represents the set of arcs. Let $H \subseteq N$ be the set of potential hub locations, and K represent the set of commodities whose origin and destination points belong to N. W_k is the amount of flow $k \in K$ to be routed from origin $o(k) \in N$ to destination $d(k) \in N$. For each node $k \in N$, f_k is the fixed cost of opening a hub at node k. Let d_{ij} be the distance between nodes i and j, which we assume to satisfy the triangle inequality.

Considering the assumption of fully interconnection between hubs and that distances satisfy the triangle inequality, we know every O/D path contains at least one and at most two hubs. Thus, paths between two nodes are of the form $(o(k), i, j, d(k))$, where $(i, j) \in H \times H$ is the ordered pair of hubs and $o(k)$ and $d(k)$ are the origin and destination of commodity k, respectively. Thus, the directed transportation costs of routing commodity k through the path $(o(k), i, j, d(k))$ is $F_{ijk} = W_k(\chi d_{o(k)i} + \alpha d_{ij} + \delta d_{jd(k)})$, where χ, α and δ represent the collection, transfer and distribution costs along the path. We assume that $\alpha < \chi$ and $\alpha < \delta$ to reflect economies of scale between hubs. In order to provide an MIP formulation for the UHLPMA, we define the following sets of decision variables

$$z_i = \begin{cases} 1 & \text{if a hub facility is located at node } i; \\ 0 & \text{otherwise.} \end{cases}$$

$$x_{ijk} = \begin{cases} 1 & \text{if commodity } k \text{ is routed using hub arc } (i, j); \\ 0 & \text{otherwise.} \end{cases}$$

The UHLPMA can be stated as follows (Hamacher et al. [65], Marín [90]):

$$
\begin{aligned}
\text{minimize} \quad & \sum_{k \in K} \sum_{(i,j) \in A} F_{ijk} x_{ijk} + \sum_{i \in H} f_i z_i & & \\
\text{subject to} \quad & \sum_{(i,j) \in A} x_{ijk} = 1 & \forall\, k \in K \qquad & (2.1) \\
& \sum_{j \in H} x_{ijk} + \sum_{j \in H : i \neq j} x_{jik} \leq z_i & k \in K, i \in H \qquad & (2.2) \\
& x_{ijk} \geq 0 & \forall\, (i,j) \in A \; \forall\, k \in K \qquad & (2.3) \\
& z_i \in \{0, 1\} & \forall\, i \in H. \qquad & (2.4)
\end{aligned}
$$

The objective function minimizes the set-up cost for opening the hubs and the transportation costs for routing the commodities through the network. Constraints (2.1) guarantee that there is a single path connecting the origin and destination nodes of every commodity. Constraints (2.2) prohibit commodities from being routed via a non-hub node. We note that this formulation has $O(n^4)$ variables and $O(n^3)$ constraints and is known to provide tight *linear programming* (LP) relaxation bounds.

2.2.3 Hub Arc Location Problems

HALPs relax the assumption of fully interconnection of hub nodes previously used in UHLPMA. Although the assumption that the hub arcs form a complete graph on the hub nodes simplifies the network design decisions, the topology and cost structure imposed by this assumption may not be very realistic in some applications in which there is a considerable set-up cost associated with the hub arcs. For this reason, HALPs incorporate an additional network design decision that considers the location of a set of hub arcs, and their associated hub nodes. This considerably increases the complexity of designing the hub network, as a new type of arc may arise. In UHLPMA, it is assumed that the transportation cost is discounted between all hub pairs. However, in hub arc location models, it is possible to connect two hubs with a bridge arc. For this reason, the structure of O/D paths become more involved in HALPs (Campbell et al. [24]).

The HALP consists of locating a set of hub arcs and hub nodes, and of determining the routing of flows through the network, with the objective of minimizing the set-up cost for designing the network and the transportation costs for routing the commodities through the network. Some HALPs impose particular topological structures such as tree-star (Contreras et al. [40]), star-star (Labbé and Yaman [82]), ring-star (Contreras et al. [42]), and hub lines (Martins de Sá et al. [93, 94]). Figure 2.1, compares the topological structure of classical UHLPMA and HALPs.

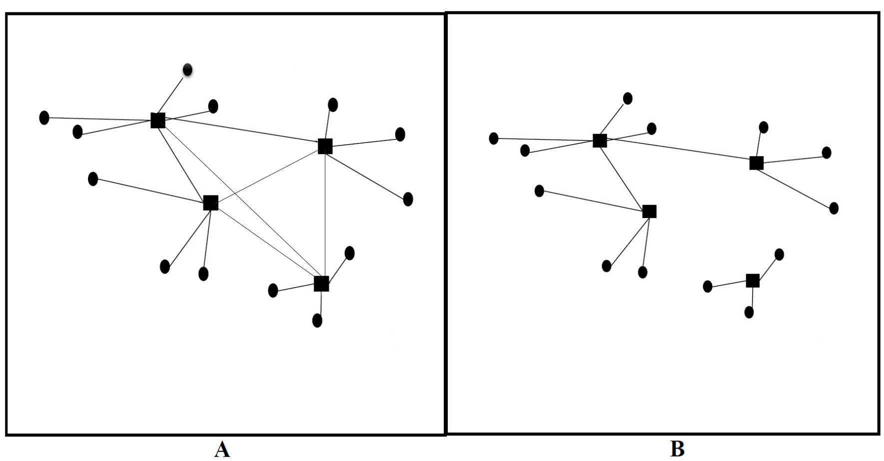


Figure 2.1: Solution network of a UHPMA (A) and of a HALP (B).

2.2.4 Hub Covering Problems

Contrary to most HLPs and HALPs that optimize a cost-based (or service-based) objective, HNDPPs deal with a profit-oriented objective for the simultaneous optimization of the revenue obtained for the service offered and the costs due to the design of the network and to transportation. This feature relates HNDPPs to *maximal hub* *covering problems* (MHCPs) in which commodities between O/D pairs have to be delivered within a time limit (service level). It is implicitly assumed that a commodity is served whenever its O/D nodes are within a predefined radius of some hub node. Because MHCPs restrict the length of the arcs of O/D paths to a given coverage radius, applications of these problems frequently arise in the design of telecommunication networks, where the signal deterioration must be taken into account (Campbell and O'Kelly [26]). Campbell [21] introduces different MHCPs, which have also been studied and extended by other authors (see Alumur and Kara [7], O'Kelly and Miller [105], Zanjirani Farahani et al. [120]). More recently, Hwang and Lee [70] study the uncapacitated single allocation p-hub maximal covering problem, which maximizes the overall demand that can be covered by p facilities within a fixed coverage radius. Lowe and Sim [85] study a MHCP that considers jointly hub setup costs and flow transportation costs, subject to covering constraints. Similarly to HNDPPs, in MHCPs some commodities may remain unserved. However, in contrast to HNDPPs, MHCPs implicitly assume that the setup cost for providing service to O/D nodes is zero and thus, they do not incorporate decisions on the nodes to be served.

2.2.5 Competitive Hub Location Problems

Another class of problems related to HLPPs are the so-called CHLPs since they also consider that it may be possible not to capture the total demand due to the presence of competitors. While most HLPs are concerned with the design of the hub network of a single firm, CHLPs consider an environment in which several firms exist in a market and compete to provide service to customers. In CHLPs each commodity chooses the competing firm that will serve its demand, based on several criteria such as travel time or service cost. The usual objective in CHLPs is to maximize the market share of some firm. Marianov et al. [89] introduce CHLPs with two competitors in which the follower looks for the best location for a set of hubs so as to maximize the captured demand. The first model assumes that a commodity demand will be fully captured if its routing cost does not exceed the current competitor's cost. A more realistic model is also considered, in which the fraction of the commodity demand that is captured is modeled using a stepwise linear function, which is used for the comparison with the competitor's routing costs. In both models, at most one path can be used to route commodities between each O/D pair. Eiselt and Marianov [51] extend these models to allow using more than one path to connect an O/D pair. The fraction of commodity demand that is routed on a particular path is modeled with a gravity-like attraction function that depends on both, the routing cost and the travel time.

Gelareh et al. [59] present a model arising in liner shipping networks, where a new liner service provider designs its network to maximize its market share, using a stepwise attraction function, which depends on service times and routing costs. Lüer-Villagra and Marianov [86] study a competitive model in which a new company wants to enter the market of an existing company. The aim is to determine the prices to charge to served commodities so as to maximize the profit of the entering company, rather than its market share. Commodities preferences for the selected firm and service route are modeled using a logit model. O'Kelly et al. [104] present a model with price-sensitive demands. It considers three different service levels for routing commodities between O/D pairs that use either two-hub O/D paths, one-hub O/D paths or direct connections. The model is formulated as an economic equilibrium problem that maximizes a nonlinear concave utility function minus the routing costs and the setup cost for the location of the hubs.

CHLPs have also been studied under a game theoretic framework, such as Stackelberg hub location models, cooperative game theoretic models with alliances and mergers, and non-cooperative game theoretic models (see Adler and Smilowitz [2], Lin and Lee [84], Asgari et al. [13], Sasaki et al. [112], Contreras [31]). We note that HNDPPs can be clearly differentiated from CHLPs, as the focus of the former is to optimize the individual decision related to one single firm rather than on competition aspects. To the best of our knowledge, besides Sasaki et al. [111] all CHLPs previously studied focus on the location of hubs and do not explicitly consider hub arc selection decisions. Moreover, none of them consider other relevant decisions such as the activation of access/bridge arcs and servicing decisions for O/D nodes.

2.2.6 Hub Location Problems in the Airline Industry

Besides MHCPs and CHLPs, other studies have also considered the design of airline hub networks. The seminal work by Grove and O'Kelly [63] analyses the relationship between hub-and-spoke networks and congestion after the U.S. Airline Deregulation Act of 1978. They considered an existing hub network with fixed hubs and performed a simulation of daily operations to analyze schedule delay. Aykin [14] study the design of hub networks in air transportation systems under different network policies: *(i)* nonstrict hubbing, in which direct connections between non-hub nodes are allowed if found cost efficient, and *(ii)* strict and restrictive hubbing, in which all demand flows to/from a node have to be routed through the same hub (single assignments). In the former case, three service policies are considered: nonstop, one-hub-stop, and two-hub-stop, whereas in the later case only one and two-hub-stops are allowed. Although the considered variants focuses mainly on the minimization of the setup cost for establishing hubs and routing flows, other extensions are discussed in Aykin [14] where the objective is to maximize profit when considering demand variations under competitive market conditions. In particular, it is assumed that demand flows are dependent on the service route selected for each OD pair. Sasaki et al. [113] study the particular case in which only the one-stop service policy is allowed. They show

how the p-hub median problem with multiple assignments under such policy can be modeled as a p-median problem. Contrary to HNDPPs, these previously mentioned models only focus on the location of hubs and do not consider other relevant decisions such as the activation of hub, access, and bridge arcs and servicing decisions for O/D nodes.

Jaillet et al. [71] consider a different approach for designing airline hub networks. Instead of explicitly considering the locational decision and associated setup costs for the hubs, the authors model the design of these networks as capacitated multi-commodity network design problems in which a predefined set of service policies are considered. Solution networks of these models will suggest the presence of hubs if cost efficient. Moreover, the number and type of aircraft of different capacity activated at each link are determined by these models. Three variants associated with different service policies are considered. The one-stop model considers only two possible services for each OD pair: *(i)* non-stop and *(ii)* one-hub-stop. The two-stop model is an extension of the first one in which a two-hub-stop service is added. Finally, the all-stop model considers no restrictions on the number of stops in each OD path. As already pointed out by Bryan and O'Kelly [20], caution must be taken when comparing the results of the models given in Jaillet et al. [71] to the ones obtained with classical hub location models as the definition of a hub is not the same. In most HLPs, a hub corresponds to a transshipment point with sufficient installed infrastructure to consolidate and reroute flows. However, in Jaillet et al. [71] hubs are defined as any city that receives a large amount of flow and thus, any city is allowed to serve as a transshipment point without ensuring that sufficient infrastructure exists to consolidate and reroute large amounts of flow. Therefore, these models focus only on link activation decisions and disregard the locational and servicing decisions considered in HNDPPs. We refer to Bryan and O'Kelly [20], O'Kelly [103], and Saberi and Mahmassani [110] for additional studies of HLPs arising in airline transportation.

2.3 Related Network Optimization Problems

HNDPPs are also related to other network optimization problems, aiming at maximizing the captured demand or optimizing some profit-oriented objective. Examples of the former are the maximal covering location (Church and ReVelle [28]) or the competitive facility location problem (Aboolian et al. [1]). Examples of the latter are prize-collecting versions of problems that do not consider locational decisions: traveling salesman (Feillet et al. [54]), vehicle routing (Aras et al. [12]), rural postman (Aráoz et al. [11]), and prize-collecting Steiner tree problems (Álvarez-Miranda et al. [9]). The above mentioned prize-collecting problems share with HNDPPs a distinguishing feature: they generalize their corresponding *classical* version by incorporating one additional level to the strategic decision-making process, so as to determine the demand customers to be served. In its turn, such decisions induce additional network design decisions. Nevertheless, according to the classification of Contreras and Fernández [37], all mentioned problems are user-facility demand. That is, service demand relates users (nodes) and service centers (facilities). Instead, HNDPPs are user-user demand, as service demand relates pairs of users among them (O/D nodes of commodities).

2.4 Bilevel Programming

A bilevel programming problem is a hierarchical optimization problem in which part of the constraints translate the fact that some of the variables constitute an optimal solution to a second optimization problem, (Labbé and Violin [80]). Bracken and McGill [17] first studied these problems and called them mathematical programs with optimization problems in the constraints. Candler and Norton [27] later introduced the terms bilevel and multilevel. In a bilevel programming problem the first optimization problem is known as the leader or the first level while the second one is called as the follower or the second level. The main point in any bilevel programming problem is that the optimality of the follower is one of the constraints for the leader's optimization problem meaning that solution of the first level is not $feasible$, if it does not lead to an *optimal* solution for the second level. A generic bilevel programming problem can be formulated as:

$$\max_{x,y} \quad f(x,y)$$

$$(x,y) \in X \tag{2.5}$$

$$\text{where} \quad \text{y solves} \tag{2.6}$$

$$\min_{y} \quad g(x,y) \tag{2.7}$$

$$\text{s.t.} \quad (x,y) \in Y. \tag{2.8}$$

where x and y denote decision vectors and X and Y the set of feasible solutions of the leader and the follower, respectively. f is the objective function of the first level while g is the objective function of the second level.

From a computational point of view, Jeroslow [72] showed that even the simple version of bilevel programming problem with linear objective function and constraints is $\mathcal{NP}$-hard. Hansen et al. [66] later proved the strong $\mathcal{NP}$-hardness of the problem. Considering this difficulty, most of the studies have focused on particular cases such as linear and convex functions to be able develop efficient solution methods (see Vicente et al. [117], Dempe [46], Colson et al. [29], Colson et al. [30]).

2.4.1 Classification and Applications

Bilevel programming has been applied to different optimization problems. Labbé and Violin [80] reviews four general types of applications of bilevel programming. We summarize them below and provide a general definition for each application briefly.

- *Pricing setting problems*: In a price setting problem the prices or taxes to some activities are set by the leader (first level) and the follower (second level) selects activities among the taxed and untaxed ones to minimize its operating costs. Examples of such problems are the optimization of highways toll systems (Labbé et al. [79], Dewez et al. [48], Heilporn et al. [68] Heilporn et al. [69]), telecommunications (Bouhtou et al. [16], Bouhtou et al. [15]), freight tariff-setting (Brotcorne et al. [18]), network design and pricing (Brotcorne et al. [19]), and setting the price for the transportation of hazardous materials (Amaldi et al. [10]).

- *Network pricing problems with pricing on arcs*: a network pricing problem on arcs is a pricing problem on a network, in which the leader is an authority which owns a subset of arcs and imposes tolls on them, and the follower is the user who travels on the network. The authority wants to maximize his/her revenue while the follower wants to minimize its costs and travel on a minimum cost path. Labbé et al. [79] introduced a bilevel network pricing for a multicommodity network, which is an example of an arc pricing. They also proved that the single toll arc network pricing is a particular case that can be solved polynomially. They provide a one level MIP formulation for network pricing on arcs, which is nonlinear. They linearized it by adding extra variables and constraints and later Dewez et al. [48] present several families of valid inequalities in order to reinforce the linear relaxation. For more studies on arc pricing we refer the reader to Roch et al. [109], Garey and Johnson [57] and Joret [73] among others.

- *Network pricing problems with pricing on paths*: A network pricing problem on paths involves tolls associated with paths. Heilporn et al. [68] proved that the multicommodity path network pricing problem is strongly $\mathcal{NP}$-hard, whether toll arcs are single or bidirectional. However, the path network pricing with only one toll path is equivalent to the single toll arc mentioned above, and thus is polynomial. Moreover, the path network pricing with only one commodity is also polynomial (Dewez and Labbé [47], Heilporn et al. [68]).

- *Product pricing problems*: The product pricing problem has at the first level a company producing and pricing a set of products to maximize its revenue, while in the second level the customers want to maximize their total utility or minimize their costs. Different versions of this problem have been widely studied in economics having different objective functions and constraints. Heilporn et al. [67] provide parallel between the pricing of substitutable products in economics and pricing of arcs of a highway network (with a polynomial number of paths) in transportation. In other words, customers correspond to commodities, products to toll arcs, prices to tolls and flows to flows. See Heilporn et al. [69] and Shioda et al. [114] for different formulations of product pricing problems.

2.4.2 The Single Level Reformulation of a Bilevel Program Problem

A very common approach to deal with bilevel programmings is to reformulate them as a single level MIP formulation using the approach presented in Labbé et al. [79]. They show how the second level optimization problem can be replaced by its primal and dual constraints and its optimality conditions stating that the primal and dual objective functions of each follower must be equal. More specifically, the single level reformulation of a bilevel programming problem consists of : *i*) the objective function and constraints of the first level, *ii*) the primal and dual constraints of the second level, and *iii*) the optimality conditions of the second level. For example, consider the following bilevel program:

$$\max_{T,x,y} \quad Tx$$

$$TC \geq f \tag{2.9}$$

$$\text{where (x,y) solves} \tag{2.10}$$

$$\min_{x,y} \quad Tx + dy \tag{2.11}$$

$$\text{s.t.} \quad Ax + By \geq b \tag{2.12}$$

where T is the price of the leader and d the price of the competitor. x is a decision vector which gets the value 1 if the follower choose the leader and 0, otherwise. y is the decision vector gets value of 1 if the follower chooses the competitor and 0, otherwise. (2.11) is the objective of the second level and (2.12) the constraints of the second level. To write the dual of the second level, we define λ as the dual variable associated with constraint (2.12). The bilevel programming can be reformulated as single level as following:

$$\max_{T,x,y} \quad Tx \tag{2.13}$$

$$\text{s.t.} \quad TC \geq f \tag{2.14}$$

$$Ax + By \geq b \tag{2.15}$$

$$\lambda A = T \tag{2.16}$$

$$\lambda B = d \tag{2.17}$$

$$Tx + dy = \lambda b \tag{2.18}$$

Constraints (2.16) and (2.17) are the dual constraints of the second level. Constraints (2.15) to (2.18) are the KKT conditions. The problem with this reformulation is that it has a nonlinear term both in the objective function (2.13) and constraint (2.17). Depending on the type of problem, sometimes it is possible to linearize formulations by defining additional variables and constraints.

Chapter 3

A Lagrangean Relaxation for Uncapacitated Hub Location Problems with Profits

In this chapter we introduce the most basic variants of *hub network design problems with profits* (HNDPPs) denoted as the *uncapacitated hub location problem with profits* (UHLPP), where it is not necessary to provide service to all demand nodes. The UHLPP releases the classical requirement of most HLPs that all service demand must be satisfied, and incorporates one additional level to the decision making process so as to determine the O/D nodes and associated commodities whose demand must be served. The rationale behind the UHLPP is that in many applications a profit is obtained for serving the demand of a given commodity. Capturing such a profit is likely to incur not only a routing cost but also additional setup costs, as the O/D nodes of the served commodities may require the installation of additional infrastructure. Classical HLPs, however, ignore these considerations, as reflected by the requirement that the demand of every commodity must be served. Broadly speaking, this requirement expresses the implicit hypothesis that the overall cost of solution networks will be compensated by the overall profit. Since such hypothesis does not necessarily hold, incorporating decisions on the nodes where service should be offered and the commodities that should be routed have important implications in the strategic and operational costs.

In UHLPPs a profit is associated with each flow between pair of nodes. The goal is the simultaneous optimization of the collected profit, the set-up cost of the hub network and the cost for routing the flow. Potential applications appear in the design of airline and ground transportation networks. A mathematical model and a lagrangean relaxation approach are presented to solve this class of problems. Numerical results on a set of benchmark instances are reported. The material used for this chapter is from Alibeyg et al. [4].

The reminder of this chapter is organized as follows. Section 3.1 provides the formal definition of the UHLPP and presents the MIP formulation. Section 3.2 describes the proposed lagrangean relaxation and analyzes the structure of the subproblems and their solutions. The chapter ends with some computational results in Section 3.3.

3.1 Problem Description

Let $G = (N, A)$ be a directed complete graph, where $N=\{1, 2, ..., n\}$ represents the set of potential nodes to provide service and A represents the set of arcs. Let $H \subseteq N$ be the set of potential hub locations, and K represents the set of commodities whose origin and destination points belong to N. W_k is the amount of flow $k \in K$ to be routed from origin $o(k) \in N$ to destination $d(k) \in N$. For each node $i \in N$, f_i is the fixed cost of opening a hub at node i, and c_i the set-up cost for serving node i. Let d_{ij} be the distance between nodes i and j, which we assume to satisfy the triangle inequality. We define P_k the per unit price of routing commodity $k \in K$ which does not depend on the path used to route the commodity.

Considering the assumption of fully interconnection between hubs and that distances satisfy the triangle inequality, every O/D path will contain at least one and at most two hubs. Thus, paths between two nodes are of the form $(o(k), i, j, d(k))$, where $(i, j) \in H \times H$ is the ordered pair of hubs and $o(k)$ and $d(k)$ are the origin and destination of commodity k, respectively. Thus, the directed transportation costs of routing commodity k through the path $(o(k), i, j, d(k))$ is $F_{ijk} = W_k(\chi d_{o(k)i} + \alpha d_{ij} + \delta d_{jd(k)})$, where χ, α and δ represent the collection, transfer and distribution costs along the path. We assume that $\alpha < \chi$ and $\alpha < \delta$ to reflect economies of scale between hubs. Using known properties of optimal solutions of classical HLPs (Contreras et al. [32]), we can define undirected transportation costs. In particular, we define a hub edge as a set $e \in E$, where E is the set of subsets of H containing one or two hubs. We define e as $\{e_1, e_2\}$ if $|e| = 2$ and as $\{e_1\}$ if $|e| = 1$. The undirected transportation cost F_{ek} for each $e \in E$ and $k \in K$ is defined as $F_{ek} = \min\{F_{ijk}, F_{jik}\}$ if $e = \{i, j\}$, and $F_{ek} = F_{iik}$ if $e = \{i\}$. Using these undirected transportation costs, we can define a set of candidate hub edges E_k for each commodity $k \in K$ as

$$E_k = \begin{cases} \{e \in E : |e| = 1\} \cup \{e \in E : |e| = 2 \text{ and } F_{ek} < \min\{F_{\{e_1\}k}, F_{\{e_2\}k}\}\}, & \text{if} \quad o(k) \neq d(k), \\ \\ \{e \in E : |e| = 1\}, & \text{otherwise.} \end{cases}$$

The UHLPP consists of selecting a set of O/D nodes to be served, of locating a set of hubs facilities and of determining the routing of a subset of flows through the network, with the objective of maximizing the difference between total revenue obtained from routing commodities minus the set-up cost for designing the network and the transportation cost for routing the commodities through the network.

3.1.1 A Mixed Integer Programming Formulation

In order to provide a MIP formulation for the UHLPP, we define the following sets of decision variables. We define binary location variables $z_i, i \in H$, equal to 1 if and only if a hub is located at node i . We also introduce binary serving node variables $s_i, i \in N$ equal to 1 if node i is served as a non-hub node. Routing variables are defined as $x_{ek}, e \in E_k$ and $k \in K$ equal to 1 if commodity k is routed using hub edge e. Using these three sets of variables, the UHLPP can be stated as follows:

$$
\begin{aligned}
(UHLPP) \quad & \text{maximize} \sum_{k \in K} \sum_{e \in E_k} W_k(P_k - F_{ek}) x_{ek} - \sum_{i \in N} f_i z_i - \sum_{i \in N} c_i s_i \\
& \text{subject to} \sum_{e \in E_k} x_{ek} \leq s_{o(k)} + z_{o(k)} && \forall\, k \in K && (3.1) \\
& \sum_{e \in E_k} x_{ek} \leq s_{d(k)} + z_{d(k)} && \forall\, k \in K && (3.2) \\
& \sum_{e \in E_k : i \in e} x_{ek} \leq z_i && \forall\, k \in K, \forall\, i \in N && (3.3) \\
& s_i + z_i \leq 1 && \forall\, i \in N && (3.4) \\
& x_{ek} \geq 0 && \forall\, e \in E_k\ \forall\, k \in K && (3.5) \\
& z_i \in \{0, 1\} && \forall\, i \in N && (3.6) \\
& s_i \in \{0, 1\} && \forall\, i \in N && (3.7)
\end{aligned}
$$

The first term of the objective function represents the net profit of routing the flows and the second and third terms represent the total set-up cost of opening a set of hubs and servicing a set of non-hub nodes, respectively. Constraints (3.1) and (3.2) state that to route each commodity, its origin and destination should be in the network, either as hub or non-hub node. Constraints (3.3) prohibit commodities from being routed via a non-hub node, whereas constraints (3.4) state that a node can be either hub or non-hub node, in case it becomes part of the solution network. Finally, Constraints (3.5)-(3.7) are the standard integrality constraints.

3.2 Solution Algorithm

Lagrangean relaxation (LR) is a well-known decomposition technique used for solving large-scale combinatorial optimization problems (Fisher [55]). It exploits the structure of the problems to compute bounds on the optimal solution value. LR has been successfully applied to different variants of HLPs (Contreras et al. [33], Marín [91]). In this section we describe a LR for the UHLPP.

3.2.1 Lagrangean Relaxation

In the case of model UHLPP, if we relax constraints (3.2) and (3.3) incorporating them to the objective function with weights given by a multiplier vector (u, v) of appropriate dimension, we obtain the following lagrangean function:

$$\begin{aligned} L(u, v) = \text{maximize} \quad & \sum_{k \in K} \sum_{e \in E_k} W_k(P_k - F_{ek}) x_{ek} - \sum_{i \in N} f_i z_i - \sum_{i \in N} c_i s_i \\ & - \sum_{k \in K} v_k \big(\sum_{e \in E_k} x_{ek} - s_{d(k)} - z_{d(k)} \big) \\ & - \sum_{i \in N} \sum_{k \in K} u_{ik} \big(\sum_{e \in E_k : i \in e} x_{ek} - z_i \big) \\ \text{subject to} \quad & (3.1), (3.4) - (3.7). \end{aligned}$$

Constraint (3.1) can be grouped into $|N|$ independent blocks, one for each possible origin node of commodities. Let K_i be the set of commodities originated at node $i \in N$. We thus observe that the location and servicing decisions of each potential hub/service node $i \in N$ only depend on the subset of commodities K_i. Therefore,

$L(u,v)$ is separable into N subproblems, one for each origin node $i \in N$ as follows:

$$
\begin{aligned}
L_i(u,v) = \text{maximize} & \sum_{k \in K} \sum_{e \in E_k} \bar{P}_k x_{ek} - \sum_{i \in N} \bar{f}_i z_i - \sum_{i \in N} \bar{c}_i s_i & \\
\text{subject to} & \sum_{e \in E_k} x_{ek} \leq s_i + z_i & \forall\, k \in K^i \\
& s_i + z_i \leq 1 & \\
& x_{ek} \in \{0,1\} & \forall\, k \in K^i \\
& z_i \in \{0,1\} & \\
& s_i \in \{0,1\}, &
\end{aligned}
$$

where

$$
\bar{P}_k = \begin{cases} (P_k - F_{ek})W_k - v_k - u_{e_1k} - u_{e_2k}, & if(e_1 \neq e_2), \\ (P_k - F_{ek})W_k - v_k - u_{e_1k}, & if(e_1 = e_2), \end{cases}
$$

$$
\bar{c}_i = \Big\{ \; c_i - \sum_{k \in K: d(k) = i} v_k,
$$

$$
\bar{f}_i = \Big\{ \; f_i - \sum_{k \in K} u_{ik} - \sum_{k \in K: d(k) = i} v_k.
$$

The optimal solution to each $L_i(u,v)$ can be efficiently obtained by evaluating the three following cases:

i) Node i will not be a hub facility nor served, i.e., $z_i = s_i = 0$, and thus no commodity originated at i will be routed through the network, i.e., $x_{ek} = 0$, for each $k \in K_i$. This case will be optimal whenever

$$
\sum_{k \in K_i} \max \left\{ 0, \max_{e \in E_k} \bar{P}_k \right\} - \min \left\{ \bar{c}_i, \bar{f}_i \right\} < 0.
$$

ii) Node i will be used to locate a hub facility, i.e., $z_i = 1$, $s_i = 0$, and a subset of commodities in $k \in K_i$ with strictly positive benefit will be routed. This case will be optimal whenever $-\bar{c}_i > -\bar{f}_i$, and

$$\sum_{k \in K_i} \max\left\{0, \max_{e \in E_k} \bar{P}_k\right\} - \bar{f}_i > 0.$$

iii) Node i will not be a hub but will be served, i.e., $z_i = 0$, $s_i = 1$, and a subset of commodities in $k \in K_i$ with strictly positive benefit will be routed. This case will be optimal whenever $-\bar{c}_i < -\bar{f}_i$, and

$$\sum_{k \in K_i} \max\left\{0, \max_{e \in E_k} \bar{P}_k\right\} - \bar{c}_i > 0.$$

The following proposition follows from the above analysis.

Proposition 3.1 $L(u, v) = \sum_{i \in N} L_i(u, v)$.

In order to obtain the best upper bound, we solve the lagrangean dual of UHLPP, which is given by:

$$(D) \qquad Z_D = \min_{(u,v) \geq 0} \quad L(u, v) = \sum_{i \in N} L_i(u, v) \tag{3.8}$$

We use a standard implementation of the subgradient optimization algorithm to solve D. The output of this algorithm is an upper bound on the optimal value of the original problem.

3.2.2 Primal Heuristic

At each iteration of the subgradient optimization algorithm, we use the lagrangean solutions to construct feasible solutions for the UHLPP. In particular, the lagrangean solutions consist of a set of open hub facilities, a set of served non-hub nodes and a set of commodities routed via some paths. However, because we relax the set of constraints that links routing decisions with locations of hub nodes and the destination of each commodity, routing solutions may not be feasible for the original problem. Therefore, to construct a feasible solution to the UHLPP we only consider the sets of hubs and served nodes from the lagrangean solution and we then route commodities via the shortest path on the solution network if its associated profit is strictly positive.

3.3 Computational Experiments

We have run a set of computational experiments in order to analyze and compare the performance of the proposed formulation and solution algorithm. The most commonly used set of instances in the hub location literature, the *australian post* (AP) set of instances, has been used to perform the computational experiments. We used different problem sizes (N=10, 20, 25, 40, 50, 60, 70, 75) and three different transportation discount factors ($\alpha = 0.2, 0.5, 0.8$) in our experiments. The LR algorithm was coded in C and run on a server with an Intel(R) processor running at 3.4GHz and 24 GB of RAM under Windows 7 environment.

We compare the *linear programming* (LP) relaxation bounds of formulation UHLPP obtained with CPLEX 12.5.1 and the upper bounds obtained with the LR. We also compare the CPU time required for CPLEX to solve the UHLPP to optimality and the CPU time required for the proposed LR to obtain the lower and upper bounds. The detailed results are summarized in Table 3.1. The first column shows the problem size. The second column is the value for α. The third column shows the LP %gap found by CPLEX while column 4 shows the time used by CPLEX to solve each problem to optimality. Nodes in column 5 presents the number of branched nodes. The second part of the table is for LR which shows the %gap obtained by LR and the associated CPU time, respectively.

		CPLEX			*LR*	
N	α	*% LP GAP*	*Time(sec)*	*Nodes*	*% LR GAP*	*Time(sec)*
10	0.2	7.61	1.70	6	7.61	0.50
10	0.5	0.00	1.68	0	0.00	0.33
10	0.8	0.00	1.77	0	0.00	0.17
20	0.2	0.00	4.38	0	0.85	3.02
20	0.5	0.00	3.74	0	0.60	3.93
20	0.8	0.00	4.22	0	0.00	4.06
25	0.2	0.00	30.08	0	0.10	8.38
25	0.5	0.00	41.41	2	0.12	8.61
25	0.8	0.00	43.90	0	0.06	9.19
40	0.2	0.00	231.00	0	0.46	88.33
40	0.5	0.00	242.97	0	0.32	88.62
40	0.8	0.00	182.00	0	0.18	88.81
50	0.2	0.00	1259.00	0	0.20	208.84
50	0.5	0.00	1115.95	0	0.29	213.37
50	0.8	0.00	1094.78	0	0.13	212.67
60	0.2	N.A.	N.A.	N.A.	0.50	453.26
60	0.5	N.A.	N.A.	N.A.	0.41	454.84
60	0.8	N.A.	N.A.	N.A.	0.19	453.63
70	0.2	N.A.	N.A.	N.A.	0.67	861.91
70	0.5	N.A.	N.A.	N.A.	0.67	869.27
70	0.8	N.A.	N.A.	N.A.	0.57	877.85
75	0.2	N.A.	N.A.	N.A.	0.39	1145.69
75	0.5	N.A.	N.A.	N.A.	0.60	1148.64
75	0.8	N.A.	N.A.	N.A.	0.25	1159.12

Table 3.1: Computational results for UHLPP with AP instances

As can be seen in Table 3.1, formulation UHLPP has small optimality gap associated with its LP relaxation bound. In all but one instance with up to 50 nodes, the optimal solution of the LP relaxation was integer. However, CPLEX is not able to load larger instances having more than 50 nodes due to memory limitations. On the other hand, the proposed LR is able to obtain tight lower and upper bounds for instances with up to 75 nodes in reasonable CPU times. The CPU time required by the LR to obtain a close estimation of the LP bound is considerable smaller than the CPU time required by CPLEX to optimally solve the LP relaxation. Moreover, the primal heuristic is able to find the optimal solution for all instances that CPLEX was able to solve the problem.

Chapter 4

Hub Network Design Problems with Profits

This chapter presents a class of hub network design problems with profit-oriented objectives, which extend several families of classical hub location problems. Potential applications arise in the design of air and ground transportation networks. These problems include decisions on the origin/destination nodes that will be served as well as the activation of different types of edges, and consider the simultaneous optimization of the collected profit, setup cost of the hub network and transportation cost.

We propose and analyze alternative models and integer programming formulations. Results from computational experiments show the complexity of such models and highlight their superiority for decision-making. Given the inherent difficulty of the considered models, CPLEX was only able to solve small to medium-size problems. In chapter 5, we present an exact solution algorithm that is capable of solving more realistic, large-scale instances for the primary models. The material used for this chapter is from Alibeyg et al. [6].

The chapter is organized as follows. Section 4.1 presents the formal definition and modeling assumptions of a primary HNDPP. It also presents a mathematical programming formulation for the problem and some variants of it. Sections 4.2 and 4.3 provide more realistic and complex extensions of HNDPPs. Section 4.4 describes the computational experiments we have run. The results produced by each model are presented and analyzed. The results of the different models are compared among them.

4.1 Primary HNDPPs

In this section we first introduce a primary model where the core strategical and operational decisions in HNDPPs are identified. In this model, the main criterion that guides decisions is profit. It is applicable to private companies where their ultimate goal is to maximize their net profit, independently of any other consideration. Companies would only provide service to O/D nodes that increase their profit and, among all commodities associated with served O/D nodes, only the profitable ones would be actually routed. We next describe possible variants in which: (*i*) external regulations could force companies to provide transportation services to any commodity where both its origin and destination nodes are served, even if this would reduce their profit, and (*ii*) market penetration policies are applied to ensure a predefined presence of a company in the market by forcing to serve a minimum number of customer demands, even if this is suboptimal from a profit perspective.

4.1.1 Formal Definition and Modeling Assumptions

We can formally define a HNDPP as follows. Let $G = (N, A)$ be a complete directed graph, where $N = \{1, 2, \ldots, n\}$ represents the set of nodes and A represents the set of arcs. For each $i \in N$, $c_i \geq 0$ denotes the setup cost for serving node i and for each $i \in H$, $f_i \geq 0$ is the fixed setup cost for opening a hub at node i. If a node $i \in H$ is selected to be a hub, it is assumed that it will be possible to serve commodities originated (or with destination) at i without activating node i as a servicing node. That is, there is no need to incur in the setup cost c_i for serving node i if it becomes a hub. Each node will thus be exactly one of the following: a hub node, a served node, or an unserved node. For $(i, j) \in A$, $d_{ij} \geq 0$ denotes the distance or unit transportation cost between nodes i and j, which we assume to be symmetric, i.e., $d_{ij} = d_{ji}$, and to satisfy the triangle inequality. Let $H \subset N$ be the set of potential hub locations and $A_H \subset A$ the subset of arcs connecting two potential hub nodes, i.e. $A_H = \{(i, j) \in A \mid i, j \in H\}$, where it is possible that the two hubs coincide, i.e., $i = j$. We also consider the following two sets of undirected edges. The set of edges connecting two potential hubs, denoted as $E_H = \{\{i, j\} \mid i, j \in H\}$, and the set of edges where at least one endnode is a potential hub, denoted by $E_B = \{\{i, j\} \mid i \in N, j \in H, i \neq j\}$. Since N and H are different sets, so are E_H and E_B. Any edge $\{i, j\} \in E_H$ is indistinctively denoted as $\{j, i\}$. Instead, when we write $\{i, j\} \in E_B$, we assume that $i \in N$, $j \in H$ (this representation is not in conflict with the fact that such an edge can be traversed both in the direction from i to j and in the direction from j to i). The elements of E_H are called *hub edges* whereas the elements of E_B are either *access* or *bridge edges* and will be discussed in detail later in this section. In the literature hub edges are often referred to as hub arcs. Throughout this chapter we prefer to maintain the distinction between edges and arcs.

Edges in E_H can be activated incurring setup costs. We denote by $r_e \geq 0$ the setup cost of hub edge $e \in E_H$. When edges in E_H are activated, their associated arcs can be used for sending flows in any of their two directions. A hub edge $e =$

$\{i,j\} \in E_H$ has a per unit flow cost αd_{ij}. The parameter $\alpha, (0 \leq \alpha \leq 1)$ is used as a constant discount factor to provide reduced unit transportation costs on hub edges to represent economies of scale. It has been already pointed out in O'Kelly [102] and other works (see Contreras [31]), the limitations of using a constant flow-independent discount factor to model economies of scale between hubs in classical HLPs. However, Campbell [23] has shown that, even if setup costs for hub edges are not considered, this assumption is reasonable for HALPs given that hubs nodes are not fully interconnected. In the case of HNDPPs, in which setup costs of hub edges are considered, such limitation could be further mitigated as the activation of hub edges will occur only if enough flow is routed through them. Similarly to other HALPs, in this primary HNDPP variant edges in E_B are activated without incurring any setup cost. Also, no discount factor is applied to flows sent via edges in E_B. The per unit transportation cost of the two arcs associated with edge $e = \{i,j\} \in E_B$ is d_{ij}.

Let K denote the set of commodities where each $k \in K$ is defined as $(o(k), d(k), W_k)$, where $o(k), d(k) \in N$, respectively denote its origin and its destination, also referred to as its O/D pair, and W_k denotes its service demand, i.e., the amount of flow that must be routed from $o(k)$ to $d(k)$ if commodity k is served. The effect of serving commodity k is threefold. On the one hand it forces the activation of its O/D nodes $o(k)$ and $d(k)$. On the other hand, it produces a per unit revenue $R_k \geq 0$, which is independent of the path used to send the commodity demand W_k through the solution network. Finally, serving commodity k also incurs a transportation cost which depends on W_k and on the path that is used to route it from $o(k)$ to $d(k)$.

Similarly to most HLPs, we require that all O/D paths include at least one hub node. That is, the solution network contains no direct connections between two non-hub nodes. We assume that served nodes can be assigned to more than one hub node, i.e. multiple assignments. Moreover, we require solution networks to contain at most three edges in each O/D path. While this hypothesis is common in classical hub location models, it may seem restrictive as compared to general network design models. Note, however, that this hypothesis is consistent with the potential applications that we mention, mainly air transportation where paths with three legs already correspond to two intermediate transfers. On the other hand, our models are profit-oriented so they include additional decisions on the commodities to be served, increasing their difficulty with respect to cost-oriented models.

For a given commodity k let $(o(k), i, j, d(k))$ denote the path connecting $o(k)$ and $d(k)$, which uses a collection edge between $o(k)$ and hub i, a transfer edge between hubs i and j, and a distribution edge between hub j and $d(k)$. When $i \neq j$, not only both i and j are hub nodes, but also the intermediate leg, $\{i, j\}$, must be a hub edge. Note that O/D paths of the form $(o(k), o(k), d(k), d(k))$, using just one hub edge, may arise only when both $o(k)$ and $d(k)$ are hub nodes. O/D paths with $i = j$ do not use any hub edge and consist solely of the collection and distribution legs, i.e. $(o(k), i, i, d(k))$ (origin-hub-destination) with $o(k) \neq i$ and $d(k) \neq i$.

Paths using at least two edges necessarily contain a collection or a distribution leg, i.e. some edge from E_B used with no discount factor. Such edges are of one of the following two classes: *access* or *bridge* edges. The only difference between an access and a bridge edge is that the former connects a non-hub node to a hub node whereas the latter connects two hub nodes. Even if a bridge edge connects two hub nodes, it differs from a hub edge in its setup cost and its per unit (non-discounted) routing cost. In the primary HNDPP we assume that no bridge edge will be used as intermediate transfer edge in a three-leg O/D path. The reader is addressed to Campbell et al. [24] for further details and an extensive analysis on possibilities for O/D paths in hub location.

Taking into account the above mentioned assumptions and requirements on the structure of O/D paths, we define the per unit transportation cost for routing commodity k on the path $(o(k), i, j, d(k))$ as $F_{ijk} = (\chi d_{o(k)i} + \alpha d_{ij} + \delta d_{jd(k)})$, where the parameters χ and δ reflect weight factors for collection and distribution, respectively.

The HNDPP consists of (i) selecting a set of O/D nodes to be served; (ii) locating a set of hub facilities; (iii) activating a set of hub edges; (iv) selecting a set of commodities to be served, both of whose O/D nodes have been selected in (i); and, (v) determining the routs of the selected commodities through the solution network, with the objective of maximizing the difference between the total revenue obtained for routing the demand of the served commodities minus the sum of the setup costs for the design of the network and the transportation costs for routing the commodities. The HNDPP is clearly NP-hard given that it has as a particular case the classical *uncapacitated hub location problem with multiple assignments* (UHLPMA), which is known to be NP-hard (Contreras and Fernández [38]) Indeed, the HNDPP reduces to the UHLPMA when $c_i = 0$, for $i \in N$, $r_e = 0$, for $e \in E_H$, and $R_k = \sum_{i \in N} f_i + \max\{F_{ijk} : (i, j) \in A_H\}$, for $k \in K$.

4.1.2 An Integer Programming Formulation

For $i \in H$, we introduce binary location variables z_i equal to 1 if and only if a hub is located at node i, and for $i \in N$ we define binary variables s_i equal to 1 if and only if node i is served (i.e. activated as a non-hub node). For $e \in E_H$, we define y_e equal to 1 if and only if hub edge e is activated. Finally, for $k \in K$, $i, j \in H$, we define routing variables x_{ijk} equal to 1 if and only if commodity k is routed via arc $(i, j) \in A_H$. When $i = j$, $x_{iik} = 1$ indicates that commodity k is routed on the path $(o(k), i, d(k))$ visiting only hub i and thus, it is not routed via a hub edge. Using these sets of variables, the HNDPP can be formulated as follows:

$$
\begin{aligned}
(PO_1) \quad \text{maximize} \quad & \sum_{k\in K} \sum_{(i,j)\in A_H} W_k(R_k - F_{ijk})x_{ijk} - \sum_{i\in H} f_i z_i - \sum_{i\in N} c_i s_i \\
& - \sum_{e\in E_H} r_e y_e && && (4.1)\\
\text{subject to} \quad & \sum_{(i,j)\in A_H} x_{ijk} \le s_{o(k)} + z_{o(k)} && k\in K && (4.2)\\
& \sum_{(i,j)\in A_H} x_{ijk} \le s_{d(k)} + z_{d(k)} && k\in K && (4.3)\\
& \sum_{j\in H} x_{ijk} + \sum_{j\in H: i\neq j} x_{jik} \le z_i && k\in K, i\in H && (4.4)\\
& x_{ijk} + x_{jik} \le y_e && k\in K, e=\{i,j\}\in E_H && (4.5)\\
& x_{ijk} \ge 0 && (i,j)\in A_H, k\in K && (4.6)\\
& z_i \in \{0,1\} && i\in H && (4.7)\\
& s_i \in \{0,1\} && i\in N && (4.8)\\
& y_e \in \{0,1\} && e\in E_H && (4.9)
\end{aligned}
$$

The first term of the objective function represents the net profit for routing the commodities. The other terms represent the total setup costs of the hubs that are chosen, the non-hub nodes that are selected to be served, and the hub edges that are used. Constraints (4.2) and (4.3) impose that the O/D nodes of each routed commodity are activated, either as hub or served nodes. Constraints (4.4) prevent commodities from being routed via non-hub nodes, whereas constraints (4.5) activate hub edges. Finally, constraints (4.6) to (4.9) define the domain for the decision variables. As usual in uncapacitated hub location models, the above formulation does not require to explicitly impose the integrality of the routing variables x. Each commodity, if routed, will use exactly one path of the solution network. Also, given that $f_i \ge 0$ and $c_i \ge 0$, in any optimal solution to PO_1 a hub node will not be activated also as a served node, that is $s_i + z_i \le 1$ for each $i \in H$.

The above formulation has a very large number of variables and constraints. However, we can exploit the following properties to reduce its size.

Property 4.1 *There is an optimal solution to formulation* (4.1)–(4.9) *where* $x_{ijk} = 0$*, for every* $k \in K$ *and* $(i, j) \in A_H$*, with* $R_k - F_{ijk} \leq 0$.

Property 4.1 is a direct consequence of the modeling assumption that only profitable commodities will be routed. According to it, for each commodity $k \in K$ all the routing variables whose cost is not strictly smaller that its revenue R_k can be eliminated, as routing them will not increase the system overall profit.

Property 4.2 *Let* $Q = \{(z, s, y, x)$ *that satisfy* $(4.2)-(4.9)\}$ *be the domain of feasible solutions to* PO_1*. Then,*

For every $k \in K$ *and* $e = \{i, j\} \in E_H$*,* $y_e \leq z_i$ *and* $y_e \leq z_j$*.*

Property 4.2 is a direct consequence of the fact that points (z, s, y, x) that satisfy constraints (4.4) and (4.5) ensure that $y_e = 1$ if its endnodes are hubs.

4.1.3 HNDPPs with Service Commitments

Model PO_1, is "flexible", in the sense that, among all commodities connecting served O/D nodes, only those that are actually profitable will be routed. In PO_1 it is thus possible that a commodity is not routed even if both its origin and destination are activated. It will only be served if routing it produces an additional profit. As mentioned, such a model can be applicable, for instance, in airline and ground transportation systems. Servicing a city does not mean that connections between this city and any other servicing city in a company's network will be necessarily offered. Only connections between such city and other cities that are profitable will be offered.

A more restrictive variant of PO_1, denoted as PO_2, arises in applications where either service commitments or external regulations impose the decision maker to serve any commodity whose O/D nodes are both activated, even if this would reduce the total profit. An important consequence of this requirement is that the solution networks to PO_2 will consist of a single connected component with no isolated hub nodes. PO_2 can be formulated by adding to PO_1 the following constraint:

$$s_{o(k)} + z_{o(k)} + s_{d(k)} + z_{d(k)} \leq \sum_{(i,j)\in A_H} x_{ijk} + 1 \quad k \in K. \tag{4.10}$$

Constraints (4.10) force commodities to be routed if their O/D nodes are both activated. We note that Property 1 no longer holds for PO_2 because of the addition of constraints (4.10). As it will be shown in Section 4.4, this additional requirement considerably increases the complexity for optimally solving PO_2 with a general purpose solver.

Previous models can be easily adapted to deal with market penetration policies that ensure a predefined presence of a company in the market by servicing a minimum number of customers demands, even if this is suboptimal from a profit perspective. This can be attained by imposing to serve a fraction of the total number of commodities or to route a fraction of the total demand, for example.

A constraint that imposes that a minimum fraction $0 \leq \beta_1 \leq 1$ of the total number of commodities are served is:

$$\sum_{k\in K} \sum_{(i,j)\in A_H} x_{ijk} \geq \beta_1 |K|, \tag{4.11}$$

Similarly, a constraint that imposes that the overall flow that is routed through the network is at least a fraction $100\beta_1$ of the overall demand $\sum_{k\in K} W_k$ is:

$$\sum_{k\in K} \sum_{(i,j)\in A_H} W_k x_{ijk} \geq \beta_2 \sum_{k\in K} W_k. \tag{4.12}$$

4.1.4 HNDPPs with Direct Connections

Previous models assume that all O/D paths include at least one hub node. That is, the solution network contains no direct connections between two non-hub nodes. In airline passenger transportation, there might be pairs of cities that produce higher profits when connected with nonstop flights than via one or two-hub stop flights. PO_1 can be easily extended to allow such direct connections. To model this case, we need to define an extra set of routing variables ψ_k equal to 1 if and only if commodity k is routed via a direct link between $o(k)$ and $d(k)$. The objective function (4.1) is modified by adding the extra term $\sum_{k \in K} W_k(R_k - d_{o(k)d(k)})\psi_k$, to account for the revenue of serving commodities that are connected directly. Moreover, constraints (4.2) and (4.3) need to be modified as below.

$$\sum_{(i,j) \in A_H} x_{ijk} + \psi_k \leq s_{o(k)} + z_{o(k)} \qquad k \in K$$
$$\sum_{(i,j) \in A_H} x_{ijk} + \psi_k \leq s_{d(k)} + z_{d(k)} \qquad k \in K.$$

4.2 HNDPPs with Setup Costs on Access/Bridge Edges

We now introduce an extension of the primary HNDPPs presented in the previous section that incorporates link activation decisions on access and bridge edges. This makes more challenging not only the design of the hub network, but also the routing of commodities, which in turn makes the problem considerably more difficult to solve. We recall that E_B denotes the set of edges which can be activated as access or bridge edges. Let q_e denote the setup cost of edge $e \in E_B$.

Contrary to previous models where bridge edges can only appear in a three-leg

O/D path as collection or distribution leg, in this new model, we now consider that a bridge edge can also appear as a transfer (or intermediate) leg. Consequently, the transportation cost of commodities that are routed through bridge arcs is different from the ones that use hub arcs since there is no discount factor on the bridge arcs. We thus define the per unit transportation cost for routing commodity k on the path $(o(k), i, j, d(k))$ (where arc $(i, j) \in E_B$ is a bridge arc) as $F'_{ijk} = (\chi d_{o(k)i} + d_{ij} + \delta d_{jd(k)})$. Given that now it is possible to route commodities between hub nodes with a bridge edge on a three-leg O/D path, the model will have to select whether to activate a link as a hub edge if enough flow is being routed, so as to compensate the higher setup cost associated with a hub edge. Otherwise, it may active the link only as a bridge edge to get a smaller profit out of a set of commodities. Note that, because of the setup costs on bridge edges, optimal solutions may concatenate two consecutive bridge arcs, despite of the triangle inequality on routing costs.

We introduce two new sets of decision variables. For $e \in E_B$, t_e equals to 1 if and only if edge e is activated as an access or bridge edge. The new set of routing variables are used to differentiate the type of routing used for each commodity. In particular, we define x'_{ijk} equals to 1 if and only if commodity $k \in K$ is routed via bridge arc $(i, j) \in A_H$. the HNDPPs with setup costs on access/bridge edges can then be formulated as:

$$
\begin{aligned}
(PND) \quad \text{maximize} \quad & \sum_{k \in K} \sum_{(i,j) \in A_H} W_k (R_k - F_{ijk}) x_{ijk} + \sum_{k \in K} \sum_{(i,j) \in A_H} W_k (R_k - F'_{ijk}) x'_{ijk} \\
& - \sum_{i \in H} f_i z_i - \sum_{i \in N} c_i s_i - \sum_{e \in E_H} r_e y_e - \sum_{e \in E_B} q_e t_e \qquad (4.13) \\
\text{subject to} \quad & (4.4) - (4.9)
\end{aligned}
$$

$$\sum_{(i,j)\in A_H} (x_{ijk} + x'_{ijk}) \leq s_{o(k)} + z_{o(k)} \qquad k \in K \quad (4.14)$$

$$\sum_{(i,j)\in A_H} (x_{ijk} + x'_{ijk}) \leq s_{d(k)} + z_{d(k)} \qquad k \in K \quad (4.15)$$

$$\sum_{j\in H} x'_{ijk} + \sum_{j\in H: i\neq j} x'_{ijk} \leq z_i \qquad k \in K, i \in H \quad (4.16)$$

$$x'_{ijk} + x'_{jik} \leq t_e \qquad k \in K, e = \{i,j\} \in E_B \quad (4.17)$$

$$\sum_{(i,j)\in A_H} (x_{ijk} + x'_{jik}) \leq 1 \qquad k \in K \quad (4.18)$$

$$\sum_{j\in H} (x_{ijk} + x'_{jik}) \leq t_{o(k)i} \qquad k \in K, (o(k), i) \in E_B \quad (4.19)$$

$$\sum_{i\in H} (x_{ijk} + x'_{jik}) \leq t_{d(k)j} \qquad k \in K, (d(k), j) \in E_B \quad (4.20)$$

$$x'_{jik} \geq 0 \qquad (i,j) \in A_H, k \in K \quad (4.21)$$

$$t_e \in \{0,1\} \qquad e \in E_B. \quad (4.22)$$

The first term of the objective function represents the net profit of routing commodities through hub edges (with discount factor) while the second term is the net profit of routing commodities through bridge edges (without discount factor). The setup costs are the same as in PO_1 with additional setup costs of the access/bridge edges. Constraints (4.14), (4.15), and (4.16) are equivalent to constraints (4.2), (4.3), and (4.4). Constraints (4.17) activate bridge edges. Constraints (4.18) indicate that commodities can be routed using either hub edges or bridge edges. Constraints (4.19) and (4.20) impose that collection and distribution edges are activated (either as access or bridge edges).

4.3 HNDPPs with Multiple Demand Levels

In all previous models, it is assumed that if a commodity $k \in K$ is served then all its demand W_k will be routed and a revenue R_k will be received. However, in practice, for a given O/D pair the amount of demand W_k that is actually served can be related to the price set to provide such transportation service. That is, the amount of demand that requires service associated with a commodity k will depend on the per unit revenue R_k set by the company. Therefore, an additional operational decision can be considered, which is to select for each commodity $k \in K$ the revenue level that will allow the company to capture the optimal portion of the total demand W_k.

In this section we extend the primary model PO_1 to the case with multiple demand levels, and consider profit-oriented models where the above mentioned decisions are taken into account. The amount of price-dependent demand that is captured for each commodity, is usually modeled with various nonlinear continuous functions (see, for instance Lüer-Villagra and Marianov [86],O'Kelly et al. [104]). In this study, to keep the model tractable while maintaining the rest of the decisions already considered, we employ a discrete approximation function that considers a set of possible values for commodities demands, each of them associated with a profit. We use L as the index set of demand and revenue levels for the commodities. For each commodity $k \in K$ and level $l \in L$, let now W_k^l denote the amount of demand that is routed if commodity k is served at level l, and R_k^l the corresponding revenue. All other data remains as in the previous models.

To formulate the first profit-oriented model with multiple demand levels, denoted as POM_1, for each $l \in L, i, j \in H$ and $k \in K$, we substitute the original set of routing variables x by an extended set of continuous routing variables, x_{ijk}^l, which denote the fraction of commodity k served at demand level l that is routed via arc $(i, j) \in A_H$. The remaining decision variables are the same as in previous primary models, since we assume that they do not depend on demand levels. The POM_1 can be formulated as follows:

$$
\begin{aligned}
(POM_1) \text{ maximize} \quad & \sum_{l\in L}\sum_{k\in K}\sum_{(i,j)\in A_H} W_k^l(R_k^l - F_{ijk})x_{ijk}^l - \sum_{i\in H} f_i z_i \\
& - \sum_{i\in N} c_i s_i - \sum_{e\in E_H} r_e y_e \\
\text{subject to} \quad & (4.6) - (4.9) \\
& \sum_{l\in L}\sum_{(i,j)\in A_H} x_{ijk}^l \le s_{o(k)} + z_{o(k)} && k\in K \quad (4.23) \\
& \sum_{l\in L}\sum_{(i,j)\in A_H} x_{ijk}^l \le s_{d(k)} + z_{d(k)} && k\in K \quad (4.24) \\
& \sum_{l\in L}\sum_{j\in H} x_{ijk}^l + \sum_{l\in L}\sum_{j\in H: i\neq j} x_{jik}^l \le z_i && k\in K, i\in H \quad (4.25) \\
& x_{ijk}^l + x_{jik}^l \le y_e && k\in K, e=\{i,j\}\in E, l\in L \quad (4.26) \\
& x_{ijk}^l \ge 0 && (i,j)\in A_H, k\in K, l\in L. \quad (4.27)
\end{aligned}
$$

The first term of the objective function represents the net profit for routing commodities at their different demand levels. The other terms are as in previous models. Constraints (4.23)-(4.27) are the analog to (4.2)-(4.6) taking into account the possible demand levels of the commodities.

Given that POM_1 does not consider any capacity constraints on the hubs or edges, it has a very useful property which can be exploited to considerably reduce the size of the above formulation. In particular, it can be shown that there is always an optimal solution to POM_1 in which for each served commodity, exactly one demand level and one path are selected. Moreover, for each commodity $k \in K$, its optimal demand level can be identified a priori. This observation is formalized in the following result.

Proposition 4.1 *For each $k \in K$, let $\bar{l}_k \in \arg\max_{l\in L}\{W_k^l R_k^l\}$. Then,*

1. *There is an optimal solution to POM_1 where $x_{ijk}^l = 0$, for $l \neq \bar{l}_k$, $(i,j) \in A_H$.*

2. *An optimal solution to POM_1 can be found by solving PO_1 with $W_k = W_k^{\bar{l}_k}$ and*

$R_k = R_k^{\bar{l}_k}$, *for each* $k \in K$.

Proposition 4.1 is a direct consequence of the fact that in POM_1 we assume that the demand levels of the commodities have no effect on of the setup costs of the network design decisions, particularly, on the setup costs of the hubs and served nodes.

Instead, in the model that we present next, denoted as POM_2, we assume that hubs and served nodes can be activated at different operation levels, incurring setup costs, which depend on the amount of flow that is processed at the nodes. That is, POM_2 is a capacitated model which considers multiple capacity levels to limit the maximum flow processed at a hub or served node. To this end, we denote as T the index set of operation levels for the hubs and for the served nodes (for ease of notation and without loss of generality we assume they are the same). For each potential hub $i \in H$ and operation level $t \in T$, let f_i^t denote the setup cost for hub i with operation level t, which allows serving a maximum amount of flow φ_i^t. Similarly, for each $i \in N$ and $t \in T$, let c_i^t denote the setup cost for serving node i with operation level t, which allows serving a maximum amount of flow ρ_i^t. The per unit transportation cost for routing commodity $k \in K$ on the path $(o(k), i, j, d(k))$ is computed as in previous models, i.e. $F_{ijk} = (\chi d_{o(k)i} + \alpha d_{ij} + \delta d_{jd(k)})$, and is independent of the levels at which the facilities $i, j \in H$ are opened. We now extend the set of decision variables for the hubs and served nodes to the following. For each $i \in H$ and $t \in T$, variable z_i^t takes the value 1 if and only if a hub is located at node i with operation level t. For $i \in N$ and $t \in T$, variable s_i^t is equal to 1 if and only if node i is served with operation level t. POM_2 can be formulated as follows:

$$
\begin{aligned}
(POM_2)\ \text{maximize}\ & \sum_{l\in L}\sum_{k\in K}\sum_{(i,j)\in A_H} W_k^l(R_k^l - F_{ijk})x_{ijk}^l - \sum_{i\in H}\sum_{t\in T} f_i^t z_i^t \\
& -\sum_{i\in N}\sum_{t\in T} c_i^t s_i^t - \sum_{e\in E_H} r_e y_e \\
\text{subject to}\ & (4.6)-(4.9),(4.26)-(4.27) \\
& \sum_{l\in L}\sum_{(i,j)\in A_H} x_{ijk}^l \le \sum_{t\in T}(s_{o(k)}^t + z_{o(k)}^t) \qquad k\in K \qquad (4.28) \\
& \sum_{l\in L}\sum_{(i,j)\in A_H} x_{ijk}^l \le \sum_{t\in T}\left(s_{d(k)}^t + z_{d(k)}^t\right) \qquad k\in K \qquad (4.29) \\
& \sum_{l\in L}\left(\sum_{j\in H} x_{ijk}^l + \sum_{j\in H: i\ne j} x_{jik}^l\right) \le \sum_{t\in T} z_i^t \qquad k\in K, i\in H \qquad (4.30) \\
& \sum_{t\in T} s_i^t + \sum_{t\in T} z_i^t \le 1 \qquad i\in H \qquad (4.31) \\
& \sum_{k\in K}\sum_{l\in L} W_k^l\left(\sum_{j\in H} x_{ijk}^l + \sum_{l\in L}\sum_{j\in H: i\ne j} x_{jik}^l\right) \le \\
& \sum_{t\in T}\varphi_i^t z_i^t \qquad i\in H \qquad (4.32) \\
& \sum_{(i,j)\in A}\sum_{l\in L}\left(\sum_{k\in K: o(k)=h} W_k^l x_{ijk}^l + \sum_{k\in K: d(k)=h} W_k^l x_{ijk}^l\right) \le \\
& \sum_{t\in T}\rho_h^t s_h^t + M\sum_{t\in T} z_h^t \qquad h\in N \qquad (4.33) \\
& z_i^t, s_i^t \in \{0,1\} \qquad i\in N, t\in T. \qquad (4.34)
\end{aligned}
$$

The objective function and constraints (4.28)-(4.31) have a similar interpretation to those of PO_1. Constraints (4.32) guarantee that the service level at which a hub is opened allows to serve all the incoming and outgoing flow that is routed through it. Constraints (4.33) have a similar interpretation, with respect to the served nodes. They state that the total incoming and outgoing flow at a served node must not exceed its installed operational capacity. The last term $M\sum_{t\in T} z_m^t$ on the right hand side of the constraints is used to deactivate the constraint in case node h becomes a hub node, where M stands for a sufficiently large constant.

Of course more general models could be considered where different operation levels and associated setup costs are considered also for all edges. This will indeed increase further the complexity of the models, although the modeling techniques will be quite similar to the ones we have used so far. We close this section by noting that Property 4.2 holds for all the considered models.

4.4 Computational Experiments

In this section we describe the computational experiments we have run in order to analyze the performance and various aspects of the HNDPPs we have introduced in Sections 4.1 and 4.2. We give numerical results that allow quantifying the quality of the formulations we have presented and comparing the computational difficulty of the different HNDPPs models. We also provide insight on the tradeoff of the decisions involved in our models by analyzing their optimal network structures and by evaluating the effect of the different parameters on the characteristics of the optimal solutions obtained with each of the considered models.

This section is structured in several parts. We first describe the computational environment and the set of benchmark instances we have used. In Section 4.4.1, we compare some of the proposed profit-oriented models we have proposed with their classical cost-oriented counterparts. In Sections 4.4.2 to 4.4.4 we respectively give numerical results to analyze the computational performance and limitations of the formulations for the primary models PO_1 and PO_2, model PND that incorporates decisions on bridge arcs, and model POM_2, which allows multiple service levels. We close the section with a sensitivity analysis with respect to some of the parameters in Section 4.4.5, and with a focus on decision-making aspects in Section 4.4.6, where we give managerial insight by analyzing the structure of the solution networks produced by the different models.

All experiments were run on an HP station with an Intel Xeon CPU E3-1240V2 processor at 3.40 GHz and 24 GB of RAM under Windows 7 environment. All formulations were coded in C and solved using the callback library of CPLEX 12.6.3. We use a traditional (deterministic) branch-and-bound solution algorithm with all CPLEX parameters set to their default values. In all experiments the maximum computing time was set to 86,000 seconds (one day).

The benchmark instances we have used for the experiments are the well-known CAB data set of the US *Civil Aeronautics Board* (CAB), with additional data that we generated for the missing information. 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 between each pair of cities, W_k, where $o(k) \neq d(k)$. We have considered instances with $n \in \{15, 20, 25, 30, 35, 40, 45, 50, 60, 70\}$ and $\alpha \in \{0.2, 0.5, 0.8\}$. The largest 70 nodes instances have only been used with the primary formulation PO_1. Since CAB instances do not provide the setup costs for opening facilities, we use as the setup cost of opening hubs, i.e. f_i, generated by de Camargo et al. [45]. The setup costs c_i, $i \in N$, for served nodes are $c_i = \nu f_i$, where $\nu = 0.1$ unless otherwise stated. The setup costs r_e, $e = \{i, j\} \in E_H$, for activating hub edges are $r_e = \tau(f_i + f_j)/2$, where $\tau \in \{0.3, 0.6, 0.4\}$ is a parameter used to model the increase (decrease) in setup costs on the hub edges when considering smaller (larger) discount factors α. The setup costs q_e, $e = \{i, j\} \in H_B$, for activating access/bridge edges are set to $q_e = \sigma(f_i + f_j)/2$, where $\sigma = 0.01$ unless otherwise stated. The revenues R_k, $k \in K$, for routing commodities are randomly generated as $R_k = \varphi \sum_{(i,j)\in A_H} F_{ijk}/|A_H|$, where φ is a continuous random variable following a uniform distribution $\varphi \sim U[0.25, 0.35]$. The collection and distribution factors are $\chi = \delta = 1$.

4.4.1 Profit-oriented vs Cost-oriented Comparison

Since all the models we have proposed are profit-oriented, a natural question is how they compare to traditional cost-oriented models in which all nodes and associated commodities must be served. We next present a comparison of these two classes of models. Our main goal is to appreciate the added value of integrating within the decision-making process additional strategic decisions on the nodes and the commodities that have to be served, which are not explicitly considered in cost-oriented models. The profit-oriented models we have considered for this comparison are the primary HNDPP formulated via PO_1 and the HNDPP with setup costs on access/bridge edges, formulated with PND.

In order to make this comparison as fair as possible, we have selected cost-oriented hub arc location models which already incorporate the other strategic decisions (with their associated setup costs) such as where to locate the hubs, what hub edges to activate, and the operational decisions to determine how to route commodities. They impose activating all demand nodes and routing all commodities, and aim at minimizing the total setup cost for the hub nodes and hub edges and the transportation cost for routing commodities. Such models belong to the class of HALPs studied in Contreras and Fernández [38]. From a network topology view-point, they can also be seen as hub network design problems with *Protocol F* networks (O'Kelly and Miller [105]).

For each of the considered profit-oriented models PO_1 and PND, we obtain an optimal solution to an associated cost-oriented model, denoted as $PO_1 - HALP$ and $PND - HALP$, as follows. We first solve formulations PO_1 and PND with $c_i = 0$, for $i \in N$, and $R_k = \sum_{i \in N} f_i + \sum_{e \in E_H} r_e + \max\{F_{ijk} : (i,j) \in A_H\}$, for $k \in K$. For

PND we also set $q_e = 0$, for $e \in E$. The optimal PO_1 and PND solutions obtained with this data consist, in each case, of sets of served nodes, open hub nodes, hub edges (plus access/bridge edges for PND), and OD paths for each commodity. Since all commodities are routed in these solutions, even if not profitable, we do a post-processing step to improve the obtained solutions in which all commodities with a negative profit are removed from the solution. The idea of this last step is that a decision-maker can use a cost-based model to obtain the design of the network and decide after which commodities to route based on their profits.

The information on the structure of optimal network and on operational aspect of solution network for PO_1 and $PO_1 - HALP$ with a set of 21 instances with up to 60 nodes is summarized in Table 4.1, and for PND and $PND - HALP$ in Table 4.2. These tables contain one block with six columns for each model. Columns *Open n-H* and *Open H* respectively show the number of nodes activated as non-hub and as hubs, whereas *hub edges* give the actual number of hub edges relative to its maximum possible value. Recall that the number of hub edges in a fully interconnected hub-level network is $\sum_{i \in H} z_i (\sum_{i \in H} z_i - 1)/2$. The last three columns in each block help analyzing the operational implications of the obtained solutions: *%Served nodes* indicate the percentage of nodes served (including both non-hub and hub nodes), *% Served O/D* show the percentage of commodities served in the solution network, computed as $100 \sum_{k \in K} \sum_{i,j \in H} x_{ijk}/|K|$, and *% Routed Flows* give the percentage of all the demand that is served, computed as $100 \sum_{k \in K} \sum_{i,j \in H} W_k x_{ijk} / \sum_{k \in K} W_k$. The last column shows %deviations between the optimal solution value of the profit-oriented model and the feasible solution obtained with the cost-oriented model, computed as $100(v_c - v^*)/v^*$, where v^* denote the optimal value of PO_1 in Table 4.1 (PND in Table 4.2), and v_c the solution value of $PO_1 - HALP$ ($PND - HALP$ in Table 4.2).

The results of the last columns of Table 4.1 show that the solutions provided

		PO_1							$PO_1 - HALP$							
α	n	*Open*		*Hub*	*%Served*	*%Served*	*%Routed*	*Open*		*Hub*	*%Served*	*%Served*	*%Routed*	*%Dev*
		n-H	*H*	*Edges*	*Nodes*	*O/D*	*Flows*	*n-H*	*H*	*Edges*	*Nodes*	*O/D*	*Flows*	
0.2	25	19	6	10/15	100.00	75.67	91.87	19	6	10/15	100	75.67	91.87	0.00
	30	24	6	9/15	100.00	78.51	91.48	24	6	10/15	100	78.39	91.46	0.09
	35	28	7	11/21	100.00	83.61	92.91	28	7	11/21	100	83.61	92.91	0.00
	40	33	7	11/21	100.00	80.64	91.77	32	8	12/28	100	84.62	93.94	0.20
	45	37	8	11/28	100.00	75.61	91.17	37	8	12/28	100	80.35	92.67	0.07
	50	42	8	12/28	100.00	78.12	93.00	42	8	12/28	100	78.12	93.00	0.00
	60	52	8	13/28	100.00	75.14	92.42	51	9	14/36	100	78.62	94.74	1.59
0.5	25	16	5	4/10	84.00	34.50	58.81	19	6	9/15	100	48.83	71.71	12.77
	30	21	5	4/10	86.67	36.21	59.30	24	6	10/15	100	51.61	71.11	13.76
	35	24	6	4/15	85.71	38.07	60.91	28	7	11/21	100	57.06	75.64	11.84
	40	28	6	4/15	85.00	37.24	59.05	32	8	12/28	100	56.86	75.62	14.19
	45	30	5	3/10	77.78	32.02	51.71	37	8	12/28	100	55.10	74.58	12.54
	50	34	6	4/15	80.00	32.94	57.79	42	8	13/28	100	54.45	74.86	13.30
	60	44	7	6/21	85.00	39.83	63.42	51	9	13/36	100	49.80	75.37	12.33
0.8	25	16	3	1/3	76.00	26.83	48.57	20	5	4/10	100	40.67	60.61	17.82
	30	20	4	1/6	80.00	30.80	50.28	24	6	4/15	100	42.30	60.50	15.75
	35	23	5	1/10	80.00	32.69	52.03	28	7	5/21	100	44.20	62.60	14.58
	40	27	5	1/10	80.00	32.95	50.49	33	7	5/21	100	45.19	62.89	14.91
	45	29	5	1/10	75.56	28.84	49.41	38	7	6/21	100	43.03	59.54	15.75
	50	31	5	1/10	72.00	26.04	48.42	43	7	7/21	100	37.02	60.69	17.05
	60	36	5	1/10	68.33	24.35	48.66	52	8	7/28	100	40.34	63.54	13.75

Table 4.1: Structure of optimal networks for PO_1 and $PO_1 - HALP$.

by $PO_1 - HALP$ are actually optimal or very close from being optimal when the discount factor α is small. However, the solutions significantly deteriorate as the discount factor increases, with deviations ranging from 11% to 18%. This can be partially explained by the need to install additional hub nodes (between 1 and 3) and activate additional hub edges (between 3 and 9) to serve all demand flows. Even with such increase on the available infrastructure, the average increase on the percentage of routed flows which are actually profitable is only about 25%.

Table 4.2 shows that somehow similar results are obtained for the case of $PND - HALP$. However, the percentage deviation of the solutions of obtained with $PND - HALP$ are much higher, ranging between 12% and 142%. Note that deviations higher than 100% indicate that the objective function is negative, meaning that losses (instead of profits) are observed in these instances. This can be partially attributed to

		PND								*PND − HALP*								
α	n	*Open*		*Access*	*Bridge*	*Hub*	*%Served*	*%Served*	*%Routed*	*Open*		*Access*	*Bridge*	*Hub*	*%Served*	*%Served*	*%Routed*	*%Dev*
		n-H	*H*	*Edges*	*Edges*	*Edges*	*Nodes*	*O/D*	*Flows*	*n-H*	*H*	*Edges*	*Edges*	*Edges*	*Nodes*	*O/D*	*Flows*	
0.2	25	19	6	28/114	1/15	10/15	100.00	74.67	91.32	17	8	32/136	4/28	14/28	100	82.33	94.05	19.22
	30	22	5	29/110	1/10	7/10	90.00	61.84	79.81	19	11	40/209	11/55	21/55	100	90.69	99.54	43.91
	35	28	7	41/196	1/21	11/21	100.00	76.89	91.02	25	10	50/250	9/45	17/45	100	87.48	98.76	18.21
	40	33	7	49/231	1/21	11/21	100.00	74.36	90.19	29	11	56/319	12/55	18/55	100	86.35	98.65	22.17
	45	38	7	55/266	1/21	11/21	100.00	70.30	88.52	33	12	76/396	15/66	20/66	100	82.93	97.53	25.05
	50	41	7	58/287	1/21	11/21	96.00	66.82	87.98	38	12	77/456	13/66	20/66	100	83.51	99.03	25.25
0.5	25	16	5	22/80	1/10	4/10	84.00	34.33	58.61	16	9	34/144	7/36	14/36	100	56.33	77.70	45.48
	30	21	5	29/105	0/10	4/10	86.67	35.75	58.66	27	3	35/81	0/3	3/3	100	32.18	53.19	12.83
	35	24	6	37/144	1/15	4/15	85.71	37.23	60.21	21	14	52/294	22/91	27/91	100	68.74	86.81	77.01
	40	26	6	42/156	2/15	4/15	80.00	34.04	56.92	27	13	65/351	18/78	24/78	100	62.69	82.35	68.49
	45	28	5	46/140	1/10	3/10	73.33	30.00	50.24	32	13	74/416	15/78	24/78	100	59.14	81.60	61.83
	50	34	6	54/204	1/15	4/15	80.00	31.80	57.38	35	15	86/525	17/105	36/105	100	62.86	84.68	94.37
0.8	25	16	3	23/48	0/3	1/3	76.00	26.83	48.57	10	15	30/150	26/105	19/105	100	58.00	74.93	128.40
	30	20	4	33/80	1/6	1/6	80.00	30.57	49.91	20	10	52/200	12/45	13/45	100	47.24	71.31	67.63
	35	23	5	41/115	2/10	1/10	80.00	32.27	51.72	20	15	63/300	28/105	18/105	100	54.37	78.24	97.37
	40	25	5	46/125	2/10	1/10	75.00	30.00	48.53	24	16	65/384	29/120	24/120	100	53.85	75.73	122.55
	45	27	5	51/135	2/10	1/10	71.11	26.57	47.67	27	18	77/486	33/153	28/153	100	55.30	76.90	142.50
	50	30	5	54/150	2/10	1/10	70.00	24.16	47.54	38	12	96/456	17/66	18/66	100	45.80	71.04	73.98

Table 4.2: Structure of optimal networks for PND and $PND-HALP$.

two points: *(i)* contrary to PND, $PND-HALP$ does not explicitly considers setup costs for the activation of access/bridge edges (as in all HALPs previously considered in the literature), and *(ii)* the need to install more hub nodes and hub edges to serve all demand, as compared to PND.

Figure 4.1 compares the optimal networks obtained with the profit-oriented models and the cost-oriented counterparts for a particular instance with $n = 25$ and $\alpha = 0.6$. Figures 4.1a and 4.1c show the optimal networks of PO_1 and PND, respectively, whereas Figures 4.1b and 4.1d show the optimal networks PO_1-HALP and $PND-HALP$, respectively. Triangles represent hubs, full circles served nodes, and empty circles unserved nodes. Black lines represent hub edges while gray lines represent access and bridge edges.

Given that cost-based hub models impose that all commodities are served, they imply a larger number of hub nodes and hub edges. As can be seen, in both cases

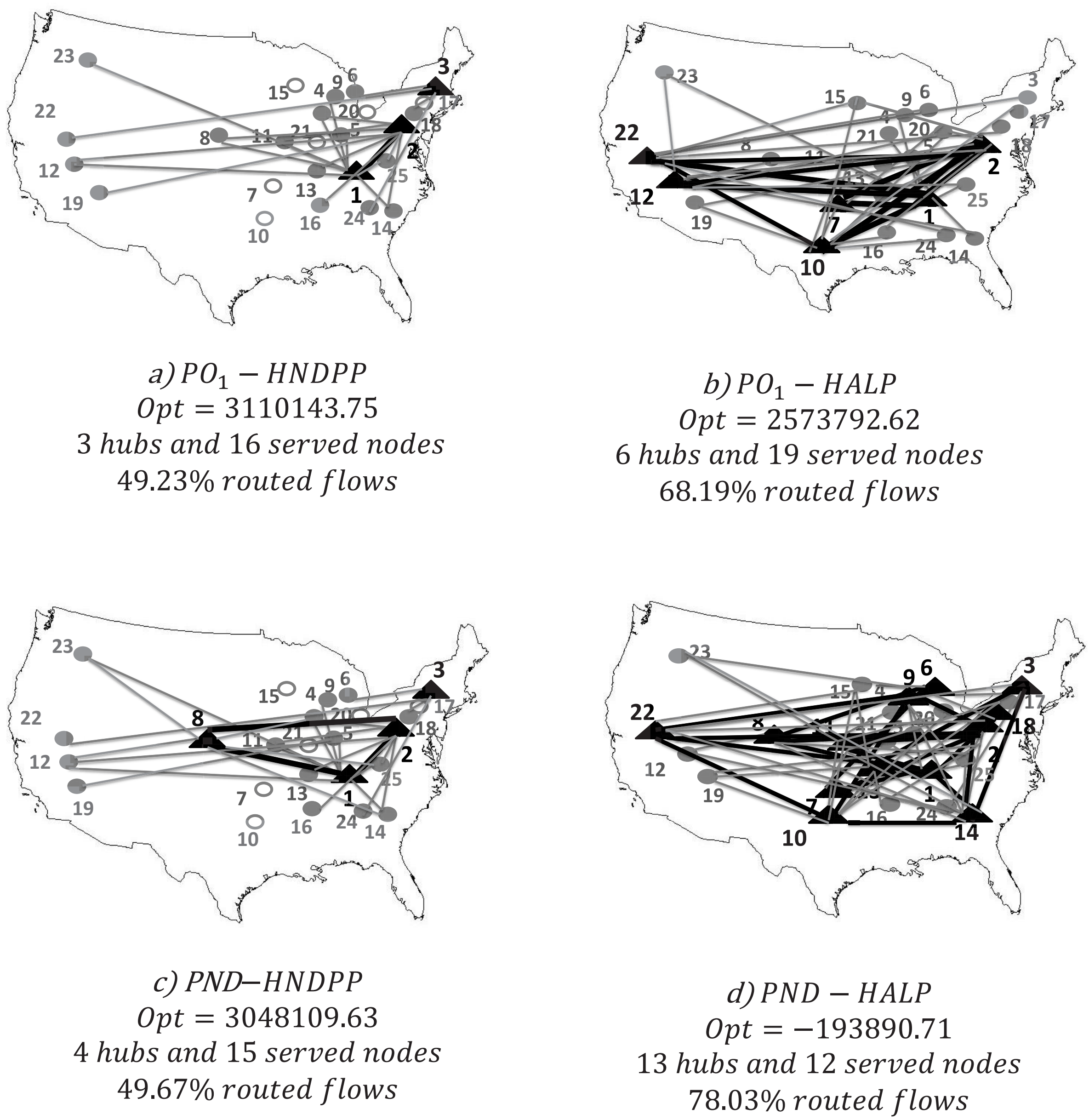


Figure 4.1: Optimal networks for PO_1 and PND with $n = 25$ and $\alpha = 0.6$.

the profit-oriented model which incorporates decisions on the nodes and demand that must be served produces a considerably better solution than the one obtained with the cost-oriented counterpart. This is particularly true in the case of the formulations that incorporate decisions on access/bridge edges, which produces a negative total profit. This creates an increase in the setup cost of the network and, as a result, solutions networks with a notable decrease in the total profits when compared to their profit-oriented counterpart.

Figure 4.1 also allows to compare the optimal networks produced by the profit-oriented formulations PO_1 and PND, so we can analyze the effect of incorporating decisions on the use of access/bridge edges, inducing additional setup costs. As can be observed the difference on the number of served nodes and served commodities is very small. Nevertheless, the total profit obtained with the solution produced by PO_1 is about 2% higher than the one obtained with PND.

4.4.2 Numerical Results for Primary HNDPPs

Our first series of experiments was oriented to study the computational performance of the primary HNDPPs, represented by formulations PO_1 and PO_2, whose numerical results are summarized in Tables 4.3 and 4.4.

Table 4.3 gives results on the computational effort needed to solve the primary profit-oriented models with a set of 24 instances with up to 70 nodes for PO_1 and the subset with the 21 instances with up to 60 nodes in the case of PO_2. The first two columns give information on the instances: α, the discount factor on hub edges, and n, the number of nodes. The first of the two 4-columns blocks corresponds to PO_1, whereas the second one corresponds to PO_2. Each column within each block gives information about the performance of the solution algorithm and its associated bounds for the corresponding model. *% LP GAP* shows percentage gaps between the values of the *linear programming* (LP) relaxations and optimal values, computed as $100(v_{LP} - v^*)/v^*$, where v^* and v_{LP} denote the optimal and LP values, respectively. *Optimal value* give optimal solution values (v^*), *Time(sec)* the computing times (in seconds) needed to optimally solve each instance, and *Nodes* the number of nodes explored by CPLEX in the enumeration tree.

		PO_1				PO_2			
α	n	*% LP gap*	*Time (s)*	*Nodes*	*Optimal value*	*% LP gap*	*Time (s)*	*Nodes*	*Optimal value*
0.2	25	0.00	3.00	0	4940995.75	0.00	25.15	0	4794995.92
	30	0.00	9.64	0	5475562.01	0.00	130.19	0	5286060.60
	35	0.00	30.05	0	6219389.73	0.00	386.81	0	6007414.96
	40	0.00	126.12	0	5992658.91	0.00	1162.89	0	5808861.17
	45	0.00	249.46	0	6171839.75	0.00	2364.55	0	5963158.91
	50	0.00	513.20	0	6471414.64	0.00	5557.70	0	6255892.56
	60	0.00	2370.97	0	7416226.43	0.00	37065.80	0	7153438.48
	70	0.00	10460.44	0	7795411.69				
0.5	25	0.00	1.60	0	3315967.83	0.00	10.24	0	2926230.46
	30	0.00	4.55	0	3624158.73	0.00	31.81	0	3201896.50
	35	0.00	10.16	0	4048946.70	0.00	109.04	0	3488014.39
	40	0.00	21.53	0	3909497.15	0.00	216.29	0	3405507.62
	45	0.00	42.17	0	3894805.98	0.00	566.33	0	3316412.33
	50	0.00	75.90	0	4219096.87	0.00	1364.90	0	3522264.30
	60	0.00	309.06	0	4777497.83	0.00	8339.37	0	3990983.53
	70	0.00	1006.20	0	5006776.75				
0.8	25	0.00	1.36	0	2940862.05	0.00	7.06	0	2557728.67
	30	0.00	3.62	0	3125465.17	0.00	17.87	0	2701547.65
	35	0.00	7.89	0	3503413.58	0.00	42.68	0	2939186.01
	40	0.00	15.42	0	3354761.27	0.00	88.42	0	2879733.41
	45	0.00	27.38	0	3451620.79	0.00	160.15	0	2846768.07
	50	0.00	44.93	0	3679333.83	0.00	305.75	0	2991043.52
	60	0.00	121.70	0	4083218.91	0.00	805.18	0	3300799.91
	70	0.00	293.20	0	4276872.10				

Table 4.3: Computational experiments for PO_1 and PO_2.

The results of Table 4.3 show that CPLEX can solve to optimality all considered PO_1 instances with up to 70 nodes. The computing times largely depend not only on the sizes of the instances but also on the discount factor α. While for $\alpha = 0.8$ all instances were solved in less than 5 minutes, the largest 70 node instance required almost three hours of computing time for the smallest discount factor $\alpha = 0.2$, which still can be considered small for an instance of that size. These small computing times are attributed to the effectiveness of Property 1 for eliminating a large number of x_{ijk} variables and constraint (4.5). In particular, for $\alpha = 0.2$, $\alpha = 0.5$ and $\alpha = 0.8$, respectively, the average percentage of eliminated variables is 94% , 97% and 98%, whereas the average percentage of eliminated constraint (4.5) is 87%, 94%, and 96%. Note that all 24 considered instances were optimally solved at the root node, as the solutions to their LP relaxations were already optimal. From this point of view, PO_1 has a performance similar to other traditional hub location models without capacity constraints, that very often have integer LP solutions (Hamacher et al. [65]).

The last four columns of Table 4.3, which summarize the results for model PO_2, allow us to quantify the effect of the constraint (4.10) on the difficulty for solving the basic models. Recall that these constraints force commodities to be routed if their O/D nodes are both activated. We can observe a notable increase of the computing times relative to those of PO_1, particularly for the instances with the smallest discount factor $\alpha = 0.2$ (observe the 10 hours of computing time that were needed to solve the largest instance with $n = 60$). This is indeed due to the fact that Property 1 no longer holds for PO_2 so it is not possible to eliminate *a priori* variables and constraints. Nevertheless, PO_2 has shown to be a tight formulation, in the sense that, similarly to PO_1, the LP relaxation of all the considered instances was already integer, so no additional enumeration was needed.

The information on the structure of optimal networks and on operational aspects of solution networks for PO_1 and PO_2 is summarized in Table 4.4, which contains one block with six columns for each formulation. Columns *Open n-H* and *Open H* respectively show the number of nodes activated as non-hub and as hubs, whereas *hub edges* give the actual number of hub edges relative to its maximum possible value. Recall that the number of hub edges in a fully interconnected hub-level network is $\sum_{i \in H} z_i(\sum_{i \in H} z_i - 1)/2$. The last three columns in each block help analyzing the operational implications of the obtained solutions: *%Served nodes* indicate the percentage of nodes served (including both non-hub and hub nodes), *% Served O/D* show the percentage of commodities served in the solution network, computed as $100 \sum_{k\in K} \sum_{i,j\in H} x_{ijk}/|K|$, and *% Routed Flows* give the percentage of all the demand that is served, computed as $100 \sum_{k\in K} \sum_{i,j\in H} W_k x_{ijk} / \sum_{k\in K} W_k$.

		PO_1						PO_2					
α	n	*Open*		*Hub*	*%Served*	*%Served*	*%Routed*	*Open*		*Hub*	*%Served*	*%Served*	*%Routed*
		n-H	*H*	*Edges*	*Nodes*	*O/D*	*Flows*	*n-H*	*H*	*Edges*	*Nodes*	*O/D*	*Flows*
0.2	25	19	6	10/15	100.00	75.67	91.87	17	6	9/15	92.00	84.33	91.43
	30	24	6	9/15	100.00	78.51	91.48	19	5	6/10	80.00	63.45	78.91
	35	28	7	11/21	100.00	83.61	92.91	27	7	11/21	97.14	94.29	95.59
	40	33	7	11/21	100.00	80.64	91.77	31	8	12/28	97.50	95.00	95.86
	45	37	8	11/28	100.00	75.61	91.17	36	8	12/28	97.78	95.56	95.77
	50	42	8	12/28	100.00	78.12	93.00	40	8	12/28	96.00	92.08	94.92
	60	52	8	13/28	100.00	75.14	92.42	49	8	12/28	95.00	90.17	94.22
	70	62	8	13/28	100.00	75.61	92.10						
0.5	25	16	5	4/10	84.00	34.50	58.81	12	4	5/6	64.00	40.00	57.48
	30	21	5	4/10	86.67	36.21	59.30	15	4	5/6	63.33	39.31	56.94
	35	24	6	4/15	85.71	38.07	60.91	18	4	5/6	62.86	38.82	56.38
	40	28	6	4/15	85.00	37.24	59.05	21	4	5/6	62.50	38.46	56.35
	45	30	5	3/10	77.78	32.02	51.71	21	3	3/3	53.33	27.88	44.25
	50	34	6	4/15	80.00	32.94	57.79	26	4	4/6	60.00	35.51	54.04
	60	44	7	6/21	85.00	39.83	63.42	33	5	6/10	63.33	39.72	56.95
	70	50	7	5/21	81.43	35.76	61.20						
0.8	25	16	3	1/3	76.00	26.83	48.57	12	2	1/1	56.00	30.33	45.28
	30	20	4	1/6	80.00	30.80	50.28	14	3	1/3	56.67	31.26	45.52
	35	23	5	1/10	80.00	32.69	52.03	17	3	1/3	57.14	31.93	45.69
	40	27	5	1/10	80.00	32.95	50.49	20	3	1/3	57.50	32.44	45.33
	45	29	5	1/10	75.56	28.84	49.41	21	3	1/3	53.33	27.88	43.43
	50	31	5	1/10	72.00	26.04	48.42	22	3	1/3	50.00	24.49	42.12
	60	36	5	1/10	68.33	24.35	48.66	26	3	1/3	48.33	22.94	41.59
	70	41	5	1/10	65.71	22.42	47.91						

Table 4.4: Structure of optimal networks for PO_1 and PO_2.

All indicators point out the high influence of the discount factor α on the design of optimal networks for both PO_1 and PO_2. As could be expected, the value of α has an important effect on the number of hub edges in optimal networks, but its effect is also noticeable on the number of hubs opened and non-hub nodes activated. This indicates that, even if it is not explicit in the formulations, large values of the discount factor for hub edges have a *discouraging* effect on the number of nodes that are activated (as hubs or as non-hubs) in optimal networks.

In particular, for PO_1 the number of open hubs and activated non-hub nodes range in the intervals [3, 8] and [16, 62], respectively. For n fixed, both numbers decrease as α increases. For PO_2, the effect of constraint (4.10) on the number of activated non-hub nodes is evident for all values of α. In contrast, the effect of constraint (4.10) on the number of hubs opened in optimal networks is influenced by the value of α. For the smallest value $\alpha = 0.2$, this number is quite similar to that of PO_1, whereas when α increases, PO_2 produces optimal networks where the number of open hubs is smaller than in the case of PO_1.

For both models, the hub-level solution network is incomplete for all instances. Again the effect of the discount factor is relevant, as the sparsity of the hub-level solution networks clearly increases with the value of α. The reduction on the number of open hubs of PO_2 relative to PO_1 produces, in its turn, an increase on the sparsity of the hub-level solution networks of PO_2, particularly for the instances with the highest value of $\alpha = 0.8$, whose optimal solutions always have just one hub edge.

Focusing on the operational aspects of solution networks, for PO_1 we can observe that the percentage of served nodes and served O/D pairs range between 65%-100% and 48%-98%, respectively. For instances of all sizes, both percentages clearly decrease as the value of α increases, although the decrease is more evident for the served O/D pairs, which for $\alpha = 0.8$ is below 33%, than for the served nodes, which for the same value of $\alpha = 0.8$ ranges in $65\% - 80\%$. The percentage of overall demand routed in optimal networks, ranges between 48%-93%, which clearly indicates that optimal networks tend to serve commodities with higher demand. The effect of α on these values is also clear and similar to that on the served nodes.

Despite the increase of the sparsity of the hub-level solution networks of PO_2, the effect of constraint (4.10) is not so evident on the operational indicators of its solution networks. While there is a slight decrease with respect to PO_1 on the percentage of served nodes, which ranges between 48%-98%, it is difficult to appreciate a decrease on the percentage of served O/D pairs, which ranges between 23%-95%. The percentage of overall demand routed in optimal networks, ranges in 41%-96%. While this percentage is higher for PO_2 than for PO_1 when $\alpha = 0.2$ (with the exception of the 30 nodes instance), it is smaller for PO_2 than for PO_1 when $\alpha = 0.8$. Taking into account that for each instance and value of α, the net profit obtained with PO_2 is always smaller than that of PO_1 (see the optimal values in the corresponding columns of Table 4.3), it seems clear that model PO_1 should be preferred to model PO_2 for larger values of α. Nevertheless for smaller values of α the comparison is not clear, as PO_2 produces solutions in which the percentage of served demand is higher than with PO_1.

4.4.3 Numerical Results for HNDPPs with Setup Costs on Access/Bridge Edges

Next we discuss the results we have obtained with formulation PND, which incorporates network design decisions on bridge and access arcs. For these experiments we have considered the subset of 21 instances with up to 60 nodes and values of $\alpha \in \{0.2, 0.5, 0.8\}$. The obtained results are summarized in Table 4.5, where columns *Access Edges* and *Bridge Edges* give the number of edges of each type in optimal solutions.

A first observation is that the computing times of PND are considerably higher than those of the most time consuming primary formulation, PO_2, particularly as the number of nodes of the instances increase. This is not surprising as PND has a larger number of both binary variables (those associated with the activation of access and bridge edges) and continuous variables (those associated to the flows routed via inter-hub bridge arcs). In any case, for PND we can again observe the influence of the discount factor α on the difficulty for solving the instances. For the largest

α	n	*% LP gap*	*Time (sec)*	*Nodes*	*Optimal value*	*Open n-H*	*H*	*Access Edges*	*Bridge Edges*	*Hub Edges*	*Served Nodes*	*%Served O/D pairs*	*%Routed Flows*
0.2	25	2.09	15.77	0	4850184.96	19	6	28/114	1/15	10/15	100.00	74.67	91.32
	30	2.08	87.97	0	5362669.76	22	5	29/110	1/10	7/10	90.00	61.84	79.81
	35	2.01	325.68	0	6086281.06	28	7	41/196	1/21	11/21	100.00	76.89	91.02
	40	3.31	1568.98	0	5831812.30	33	7	49/231	1/21	11/21	100.00	74.36	90.19
	45	3.20	4127.66	0	5969894.50	38	7	55/266	1/21	11/21	100.00	70.30	88.52
	50	2.86	9856.98	0	6239525.42	41	7	58/287	1/21	11/21	96.00	66.82	87.98
	60	2.80	66650.76	3	7105816.66	51	8	88/408	5/28	13/28	98.33	70.85	91.22
0.5	25	9.62	12.48	0	3246177.24	16	5	22/80	1/10	4/10	84.00	34.33	58.61
	30	8.14	54.49	0	3530787.94	21	5	29/105	0/10	4/10	86.67	35.75	58.66
	35	9.64	158.94	4	3928183.15	24	6	37/144	1/15	4/15	85.71	37.23	60.21
	40	9.73	421.83	0	3773879.34	26	6	42/156	2/15	4/15	80.00	34.04	56.92
	45	10.69	951.06	6	3752423.62	28	5	46/140	1/10	3/10	73.33	30.00	50.24
	50	8.85	1675.92	4	4051369.73	34	6	54/204	1/15	4/15	80.00	31.80	57.38
	60	10.79	21927.88	13	4521280.50	39	6	62/234	2/15	5/15	75.00	31.16	59.15
0.8	25	14.91	6.94	0	2875551.32	16	3	23/48	0/3	1/3	76.00	26.83	48.57
	30	15.87	39.66	7	3011761.00	20	4	33/80	1/6	1/6	80.00	30.57	49.91
	35	16.73	154.96	26	3363665.77	23	5	41/115	2/10	1/10	80.00	32.27	51.72
	40	17.03	244.24	2	3210437.26	25	5	46/125	2/10	1/10	75.00	30.00	48.53
	45	17.02	413.09	7	3283658.34	27	5	51/135	2/10	1/10	71.11	26.57	47.67
	50	15.78	624.69	2	3507899.84	30	5	54/150	2/10	1/10	70.00	24.16	47.54
	60	15.50	4450.06	8	3864127.25	36	5	70/180	3/10	1/10	68.33	24.18	48.63

Table 4.5: Computational experiments for PND.

value of $\alpha = 0.8$ the computing time for the largest 60 nodes instance is moderate, as it can be solved in less than 1.50 hours. Instead, the same instance becomes really challenging when $\alpha = 0.2$, as the computational effort needed to solve it rises to more than 18 hours.

From a computational point of view, a clear difference of PND with respect to PO_1 and PO_2 is that, for all instances and values of α, the LP relaxation of PND produced non-integral solutions with strictly positive percentage gaps. These gaps are quite small for $\alpha = 0.2$ (smaller 3.5%), and except for the largest instance with $n = 60$, they can be closed already at the root node by the CPLEX cuts added by default. As α increases the values of *% LP gap* get larger and range in 15%-17% for $\alpha = 0.8$. Still, the computational effort required to close such gaps is moderate.

Looking at the structure of optimal networks produced by PND we can observe that, similarly to the previous formulations, the hub-level solution networks are incomplete in all instances and their sparsity increases with the value of α. Furthermore, we can appreciate that adding decisions on access/bridge edges does not seem to have an important effect on the number of hub nodes that are opened, which is quite similar to that of PO_1. It can be also observed that, for most of the instances, the number of hub edges in optimal solutions is the same as in PO_1 and decreases slightly in only three instances. There is usually a small number of bridge edges, which seems independent of the value of α. For the smaller value of $\alpha = 0.2$ the number of hub edges is always higher than that of bridge edges, although this relation tends to change as α increases. The explanation is clear: hub edges are activated only if the discount factor α produces enough reduction in the routing costs since their setup cost is higher than that of bridge edges; otherwise profitable commodities are routed via bridge edges.

When analyzing the operational indicators of solution networks, it can be seen that the percentage of served nodes and served O/D pairs ranges in 68%-100% and 24%-75%, respectively. These values are very similar to those of PO_1, although some decreases can be appreciated, mainly in the instances where the number of hub edges does not coincide. Something similar can be observed with the percentage of routed flows, which ranges in 48%-92%, and are always slightly smaller than those of PO_1 except for the instances where there is a reduction on the number of hub edges, where the decrease on the flows that are routed may reach 11%. Similarly to the previous models, the difference on the percentage of served O/D pairs and routed flows indicates that optimal networks tend to serve commodities with higher demand.

4.4.4 Numerical Results for HNDPPs with Multiple Demand Levels

We have run a last series of computational experiments to evaluate the more general model POM_2, in which hubs and served nodes can be activated at different operation levels, incurring setup costs, which depend on the amount of flow that is being processed at the nodes. Now we have only considered the instances with up to 35 nodes as the computing times of larger instances become prohibitive. Moreover, for larger instances, in most cases no feasible integer solution was known at termination.

For these experiments we have adapted the CAB instances used in the previous sections to incorporate demand and revenue levels for the commodities and multiple capacity levels for the hub facilities in the following way. For each $k \in K$, we set W_k^1 and R_k^1 to W_k and R_k, respectively. Data for the other levels are generated by decreasing demand and increasing revenue. That is, we defined $W_k^l = 0.3\, W_k^{l-1}$ and $1.2\, R_k^l = R_k^{l-1}$ for $l = 2, \ldots, |L|$. In addition, for each $i \in H$, we set $f_i^1 = f_i$ and $f_i^t = 0.9 f_i^{t-1}$, $t = 2, ..., |T|$. We generated in the same way different levels of setup costs for served nodes. That is, for each $i \in N$, $c_i^1 = c_i$ and $c_i^t = 0.9\, c_i^{t-1}$, $t = 2, ..., |T|$. We have also generated different levels of capacities for the hub and served nodes. For each $i \in H$, we set $\varphi_i^1 = \lambda \sum_{i \in H} O_i / \sum_{i \in H} z_i^*$, where λ is a continuous random variable following a uniform distribution $\lambda \sim U\,[0.9, 1.1]$, and O_i is the total flow passing through hub i at the optimal solution of POM_1 (denoted as z^*). For other capacity levels of hub nodes, $\varphi_i^t = 0.7\, \varphi_i^{t-1}$, $t = 2, ..., |T|$. Capacities of the served nodes are generated as a fraction of the capacities for the hubs, i.e. $\rho_i^t = \gamma\, \varphi_i^t$, $t = 1, ..., |T|$, and $\gamma = 0.5$. Finally, we have considered $|L| = |T| = 5$.

The obtained numerical results are summarized in Table 4.6, which highlights the computational difficulty of POM_2. Since optimality of the best-known solution could not be proven in all cases, column *%gap end* gives the percentage optimality gap at termination and *%LP gap* the percentage deviation of the LP bound with respect to the best-known solution at termination. The value of such solution is given in column *Best-known value*. A value 0.01 in column *%gap end* indicates that such value corresponds to an optimal solution.

From the obtained results it can be seen that formulation POM_2 is quite tight, producing rather small % LP gaps at the root node, which do not exceed 6.5%, even

α	n	*% LP gap*	*% gap end*	*Time (sec)*	*Nodes*	*Best-known value*	*Best bound*	*Open n-H*	*H*	*Hub Edges*	*%Served Nodes*	*%Served O/D pairs*	*%Routed Flows*
0.2	25	3.82	0.00	17379.39	3306	3571879.69	3571537.82	16	9	4/36	100.00	96.83	68.11
	30	4.45	2.90	86404.12	1435	3485278.93	3387018.15	19	11	3/55	100.00	97.47	65.13
	35	6.49	5.52	86407.77	495	3372157.52	3195736.83	23	12	2/66	100.00	86.64	56.42
0.5	25	4.93	0.00	2367.17	1967	2947676.67	2947461.66	17	7	1/21	96.00	83.17	53.99
	30	4.87	0.00	43392.63	13324	2893186.37	2892897.33	22	8	1/28	100.00	86.55	51.77
	35	5.91	0.74	86410.32	3655	2964770.23	2943064.70	24	10	1/45	97.14	82.27	50.71
0.8	25	5.25	0.00	595.45	1981	2839814.46	2839554.07	18	6	0/15	96.00	81.17	48.84
	30	5.50	0.00	2398.61	3489	2822441.80	2822176.98	22	7	0/21	96.67	80.34	49.55
	35	5.87	0.00	21198.95	15736	2936660.82	2936367.74	25	9	0/36	97.14	80.76	50.01

Table 4.6: Computational experiments for POM_2.

for the instances that could not be solved to optimality. Unfortunately, these small gaps are very difficult to close, as indicated by the high number of nodes that are explored in the search trees and by the high computing times, which, for the small size instances, are affordable when $\alpha = 0.8$, but become prohibitive as α decreases or as the size of the instances increases.

Anyway, Table 4.6 allows to appreciate that in the optimal/best-known solution networks produced by POM_2 the total number of activated nodes is roughly the same as in all previous models for instances of the same size and value of α, although the number of open hubs is slightly higher (so the number of activated non-hubs is slightly lower). At the hub-level network, however, we can observe very small values for the ratio of the number of hub edges relative to the number of hub nodes, resulting in highly sparse hub-level networks. As can be seen, only the instances with $\alpha = 0.2$ produced solution networks with more than one (but very few) hub edges, whereas the hub-level networks for the $\alpha = 0.5$ instances have just one hub edge, and no hub edge is activated in the solution networks to the instances with $\alpha = 0.8$.

In all cases, the percentages of served nodes are very high, and follow a similar trend as in previous models, with full node service for instances with $\alpha = 0.2$ and decreasing slightly when $\alpha = 0.5, 0.8$. Nevertheless, even for the instances with a higher value of $\alpha = 0.8$ this percentage is never below 96% which is considerably higher than the percentage of nodes served with the other models for the same value of α. Something similar happens with the percentage of served O/D pairs, which is never below 80%, independently of the value of α. On the contrary, the percentages of routed flows are noticeably smaller than in previous models. This is indeed a sign of the selective nature of POM_2 where activated nodes and hubs can operate at different service levels and demand flows partially routed. This can be better appreciated in Table 4.7 that shows the service levels of the solution networks of POM_2.

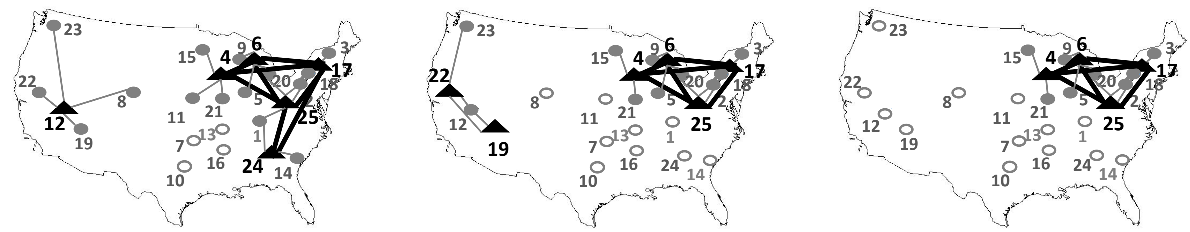
α	n	*Open Hubs level l*					*Served Nodes level l*					*Routed Flows level l*				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
0.2	25	7	0	1	0	1	2	2	5	1	6	224	75	97	95	105
	30	8	1	1	0	1	1	4	6	3	5	331	127	159	160	94
	35	9	1	0	0	2	2	4	8	1	8	348	141	143	176	250
0.5	25	3	2	1	0	1	2	2	4	3	6	170	61	72	78	126
	30	5	1	1	0	1	3	2	5	4	8	242	119	103	117	188
	35	7	1	0	1	1	3	3	7	6	5	344	119	137	161	242
0.8	25	2	2	1	0	1	1	4	4	3	6	154	68	64	86	127
	30	4	2	0	0	1	3	2	6	3	8	228	89	99	117	184
	35	7	0	1	0	1	3	4	7	5	6	327	138	139	161	229

Table 4.7: Service levels for solution networks to POM_2.

As can be seen, most of the hub nodes are activated at the lowest service level, and this trend is more evident for the lowest value $\alpha = 0.2$. In contrast, served nodes tend to be activated at higher service levels. In particular, the percentage of served nodes activated at the highest service level ranges in 30% – 60%. Routed flows are served at all service levels, although higher frequencies correspond either to the lowest or highest service levels.

4.4.5 Sensitivity Analysis

Below we present a sensitivity analysis of the presented models with respect of some of their input data. Figure 4.2 compares the optimal hub networks produced by primary formulation PO_1 for the CAB instance with $n = 25$ and $\alpha = 0.5$ when setting c_i as 0%, 15% and 40%, of the setup cost f_i.



a) $c_i = 0.00 f_i$
Opt: 8,811,009.83
6 hubs and 15 served nodes
54.38% routed flows

b) $c_i = 0.15 f_i$
Opt: 7,388,362.45
6 hubs and 10 served nodes
43.57% routed flows

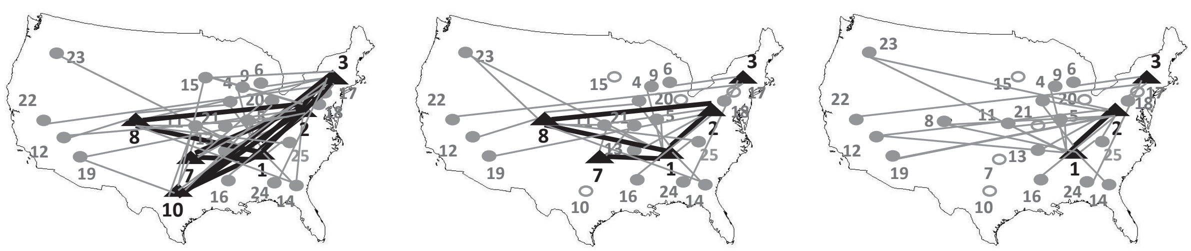
c) $c_i = 0.40 f_i$
Opt: 5,636,245.08
4 hubs and 8 served nodes
38.80% routed flows

Figure 4.2: Optimal network for PO_1 with different setup costs c_i with $n = 25$ and $\alpha = 0.5$.

Figure 4.2a depicts the optimal solution network with no setup costs for serviced nodes. It consists of two disconnected components with five interconnected hubs, one isolated hub, and 15 served nodes. Even if there are no setup costs for activating served nodes, four nodes remain unserved. Figures 4.2b and 4.2c show that, as could be expected, increasing the setup costs for serving nodes reduces the number of served nodes. In particular, using setup costs $c_i = 0.15 f_i$ (Figure 4.2b), reduces to 10 the number of served nodes. Moreover, the topology of the hub-level network also changes, even if the number of hubs has not changed. The overall profit is reduced by 16.14%. When setup costs are further increased to $c_i = 0.40 f_i$ (Figure 4.2c), the optimal solution network consists of a single connected component with four fully interconnected hubs. Now the number of served nodes has decreased to eight and the total profit is reduced by 36.03% with respect to the case where $c_i = 0$.

Figure 4.3 allows to compare the effect of the discount factor α in solution networks. It shows the optimal networks produced by the primary formulation PO_1 for the CAB instance with $n = 25$ and three different values of the discount factor α.

The optimal network for $\alpha = 0.2$ (Figure 4.3a) consists of a single connected component with six hubs, seven hub edges, and six unserved nodes. When increasing the discount factor to $\alpha = 0.5$ (Figure 4.3b), the solution network consists of two disconnected components but one hub node less and and nine unserved nodes. This causes a considerable reduction in both the number of served O/D pairs and routed flows. Figure 4.3c shows the solution network for the highest value $\alpha = 0.8$. Now the number of hub nodes has further decreased to three. Even if the number of served nodes remains the same as with $\alpha = 0.5$ there is a further decrease on the served O/D pairs and the total routed flow.



$a)\ \alpha = 0.20$
$Optimal: 4940995.75$
6 *hubs and* 19 *served nodes*
91.87% *routed flows*

$b)\ \alpha = 0.50$
$Optimal: 3315967.83$
5 *hubs and* 16 *served nodes*
58.81% *routed flows*

$c)\ \alpha = 0.80$
$Optimal: 2940862.05$
3 *hubs and* 16 *served nodes*
48.57% *routed flows*

Figure 4.3: Optimal network for PO_1 with different discount factors α with $n = 25$ and $\nu = 0.1$.

4.4.6 Comparison and Tradeoff of Proposed Models

We conclude the section by analyzing the tradeoff of our different profit-oriented models among them. In particular, we analyze the profit each of them produces per O/D pair served and per unit of flow routed. For this we use Figures 4.4 and 4.5, which respectively depict the profit per O/D pair served and the profit per unit of flow routed for the primary models, represented by formulations PO_1, PO_2, the model with access/bridge edge decisions, represented by PND, and the capacitated multiple level model, represented by POM_2. For a better visualization, each figure is separated in three parts, one for each tested value of α.

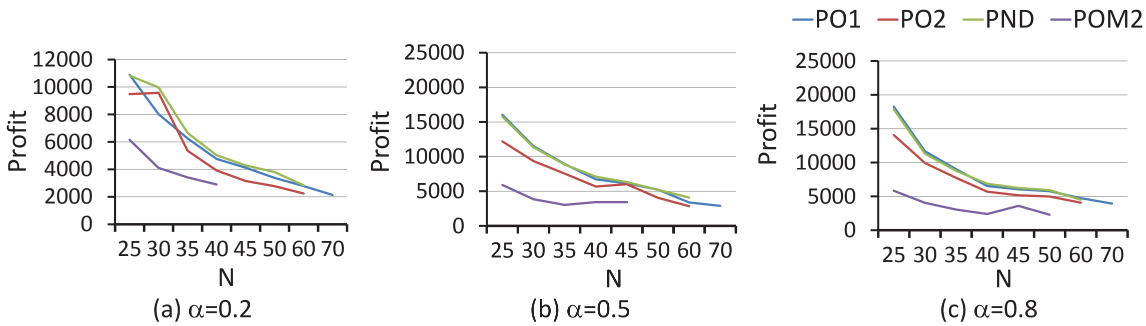


Figure 4.4: Comparison of Models: Profit per served O/D pair.

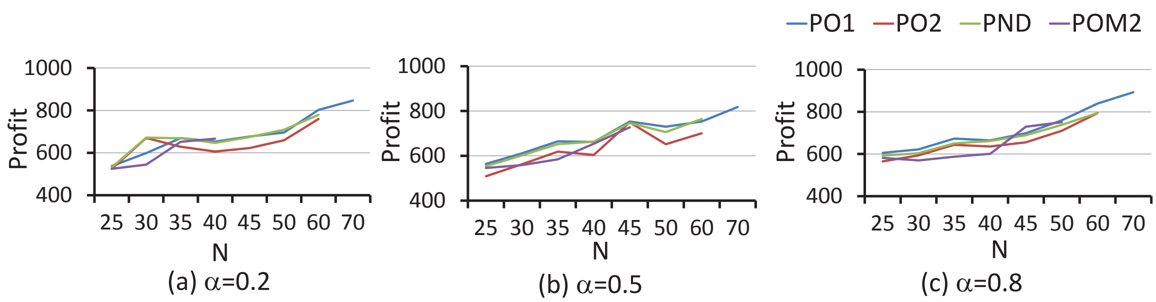


Figure 4.5: Comparison of Models: Profit per routed unit of flow.

Figure 4.4 clearly illustrate the superiority of PO_1 and PND models with respect to the other models in terms of the profit per served O/D pair. For these values of α, the *quality* of the models, measured in terms of their ability of producing solutions with a better tradeoff between their profit and the service level attained, is inversely proportional to their sophistication. Thus the primary HNDPP and the model with access/bridge edge decisions, outperforms the primary model that forces to serve any commodity whose O/D nodes are both activated, which, in turn, outperforms the multiple level model. Figure 4.5 shows that, in terms of the profit obtained per unit of flow routed, PO_1 and PND models are also superior to the other models, although the differences are rather minor. In our opinion, both figures could be very useful to help a decision maker chose among the presented models, taking into account the potential context and priorities.

Chapter 5

Exact Solution of Hub Network Design Problems with Profits

Given the inherent difficulty of the HNDPPs introduced in Chapter 4 and the fact that CPLEX was not able to solve them for larger size instances, in this chapter we propose an exact algorithm to solve the two primary HNDPPs denoted as PO_1 and PO_2. This algorithmic framework uses a lagrangean relaxation that exploit the structure of the problems and can be solved efficiently to efficiently obtain bounds at the nodes of an enumeration tree. In particular, the lagrangean functions can be decomposed in two independent subproblems: one of them is trivial and the other one can transformed into a *quadratic boolean problem* (QBP), which can be solved efficiently as a max-flow problem. The lagrangean dual problems were solved with a subgradient optimization algorithms that applied simple primal heuristics, which produced valid lower bounds. The lagrangean relaxation was embedded within exact branch-and-bound algorithms for each of the considered problems. Moreover, reduction tests were applied at the root node, which helped to considerably reduce the number of variables and constraints. These tests were enhanced with the application of a partial enumeration phase to reduce the number of branches of the enumeration phase. The results from computational experiments with benchmark instances with up to 100 nodes assessed the efficiency of the proposed framework, and its superiority over CPLEX. The material used for this chapter is from Alibeyg et al. [5].

The chapter is organized as follows. Section 5.1 describes the proposed lagrangean relaxations of PO_1 and PO_2 and the solution of their associated lagrangean duals. Section 5.2 explains the variable elimination techniques used whereas Section 5.3 presents the partial enumeration and the overall branch-and-bound algorithm. Section 5.4 describes the computational experiments we have run.

5.1 Lagrangean Relaxation

Lagrangean relaxation (LR) is a well-known decomposition method that exploits the inherent structure of the problems to compute dual bounds on the value of the optimal solution. Pirkul and Schilling [108], Elhedhli and Wu [52], and Contreras et al. [36] provide some examples of successful implementations of LR for obtaining tight bounds for various classes of HLPs.

We use the following MIP formulation for the first primary HNDPP, denoted as PO_1 presented in Chapter 4:

$$
\begin{aligned}
(PO_1) \quad \text{maximize} \quad & \sum_{k\in K}\sum_{(i,j)\in A_H} W_k(R_k - F_{ijk})x_{ijk} - \sum_{i\in H} f_i z_i - \sum_{i\in N} c_i s_i \\
& - \sum_{e\in E_H} r_e y_e && && (5.1)\\
\text{subject to} \quad & s_i + z_i \leq 1 && i \in H && (5.2)\\
& \sum_{(i,j)\in A_H} x_{ijk} \leq s_{o(k)} + z_{o(k)} && k \in K && (5.3)\\
& \sum_{(i,j)\in A_H} x_{ijk} \leq s_{d(k)} + z_{d(k)} && k \in K && (5.4)\\
& \sum_{j\in H} x_{ijk} + \sum_{j\in H: i\neq j} x_{jik} \leq z_i && k \in K, i \in H && (5.5)
\end{aligned}
$$

$$x_{ijk} + x_{jik} \le y_e \qquad k \in K, e = \{i,j\} \in E_H \tag{5.6}$$

$$x_{ijk} \ge 0 \qquad k \in K, (i,j) \in A_H \tag{5.7}$$

$$z_i \in \{0,1\} \qquad i \in H \tag{5.8}$$

$$s_i \in \{0,1\} \qquad i \in N \tag{5.9}$$

$$y_e \in \{0,1\} \qquad e \in E_H. \tag{5.10}$$

The first term of the objective function is the net profit of the commodities that are routed. The other terms represent the total setup costs of the hubs that are chosen, the non-hub nodes that are selected to be served, and the hub edges that are used. Constraints (5.2) guarantee that if a node is activated as a hub then it is not activated as a served node. Constraints (5.3) and (5.4) impose that the O/D nodes of each routed commodity are activated, either as hub or served nodes. When $o(k)$ or $d(k)$ do not belong to H then the right hand side of constraints (5.3) and (5.4) reduces to $s_{o(k)}$ and $s_{d(k)}$, respectively. Constraints (5.5) prevent commodities from being routed via non-hub nodes, whereas constraints (5.6) activate hub edges. Finally, constraints (5.7)-(5.10) define the domain for the decision variables. (5.1) - (5.10) does not require to explicitly impose the integrality of the routing variables x, since each routed commodity will use exactly one path of the solution network. Also, (5.1) - (5.10) uses $|N| + |H| + |E_H|$ binary variables, $|K||A_H|$ continuous variables, and $|H| + |K|(2 + |H| + |E_H|)$ constraints.

An extension of the above primary HNDPP, denoted as PO_2, considers service commitments that impose to serve any commodity whose O/D nodes are both activated, even if this would reduce the total profit. An MIP formulation for this more restrictive model can be obtained by adding to (5.1) - (5.10) the following set of constraints (Alibeyg et al. [6]):

$$s_{o(k)} + z_{o(k)} + s_{d(k)} + z_{d(k)} \leq \sum_{(i,j)\in A_H} x_{ijk} + 1 \quad k \in K. \tag{5.11}$$

Constraints (5.11) force to route any commodity where both its O/D nodes are activated. PO_2 has the same number of variables as PO_1 but $|K|$ additional constraints. The effect of constraints (5.11) in the actual difficulty for solving the problem is notorious. The results of Alibeyg et al. [6] show that the required CPU times for solving PO_2 with a commercial solver are at least one order of magnitude higher than those of PO_1 for all considered benchmark instances. As we will show later in Section 5.4, our algorithmic framework is capable of considerably mitigating the effect of (5.11) in the CPU times.

Our algorithmic framework uses LR to obtain upper bounds of PO_1 and PO_2. In the case of PO_1 we relax the sets of constraints (5.5) and (5.6), whereas for PO_2 we also relax the additional set of constraints (5.11). Hence, the structure of the resulting lagrangean function is very similar in both cases: the domain is the same and only the objective functions differ. In both cases the lagrangean function can be decomposed in two subproblems: one of them is trivial and the other one can transformed into a QBP. Due to the structure of the cost coefficients, we show how the lagrangean function can actually be evaluated in polynomial time. We next provide the details of the entire process for PO_1 and then briefly describe how to proceed in a similar fashion for PO_2.

5.1.1 The Lagrangean function for PO_1

When we relax constraints (5.5) and (5.6), and incorporate them to the objective function of PO_1, with weights given by a multiplier vector (λ, μ) of appropriate dimension, we obtain the following lagrangean function:

$$
\begin{aligned}
L_1(\lambda,\mu) = \text{maximize} \quad & \sum_{k\in K}\sum_{(i,j)\in A_H} W_k(R_k - F_{ijk})x_{ijk} - \sum_{i\in H} f_i z_i - \sum_{i\in N} c_i s_i \\
& - \sum_{e\in E_H} r_e y_e - \sum_{k\in K}\sum_{i\in H}\lambda_{ik}(\sum_{j\in H} x_{ijk} + \sum_{j\in H: i\neq j} x_{jik} - z_i) \\
& - \sum_{e=\{i,j\}\in E_H}\sum_{k\in K}\mu_{ek}(x_{ijk} + x_{jik} - y_e) \\
\text{subject to} \quad & (5.2) - (5.4), (5.7) - (5.10),
\end{aligned}
$$

which is equivalent to

$$
\begin{aligned}
L_1(\lambda,\mu) = \text{maximize} \quad & \sum_{k\in K}\sum_{(i,j)\in A_H} \overline{P}_{ijk} x_{ijk} - \sum_{i\in H} \overline{f}_i z_i - \sum_{i\in N} c_i s_i - \sum_{e\in E_H} \overline{r}_e y_e \\
\text{subject to} \quad & (5.2) - (5.4), (5.7) - (5.10),
\end{aligned}
$$

where

- $\overline{P}_{ijk} = \begin{cases} (R_k - F_{ijk})W_k - \lambda_{ik} - \lambda_{jk} - \mu_{\{i,j\}k}, & \text{if } (i \neq j) \\ (R_k - F_{iik})W_k - \lambda_{ik}, & \text{if } (i = j), \end{cases}$
- $\overline{f}_i = f_i - \sum_{k\in K} \lambda_{ik}$,
- $\overline{r}_e = r_e - \sum_{k\in K} \mu_{ek}$.

Note that $L_1(\lambda, \mu)$ can be decomposed in two independent subproblems, one in the y space, that we denote $L_y(\mu)$, and another one in the (z, s, x) space, that we denote $L_{z,s,x}(\lambda, \mu)$. The first subproblem reduces to

$$
L_y(\mu) = \max\left\{ -\sum_{e\in E_H} \overline{r}_e y_e : y \in \{0,1\}^{|E_H|} \right\},
$$

and an optimal solution can be obtained by inspection. That is, we set $y_e = 1$ for all $e \in E_H$ with $\overline{r}_e < 0$, and $y_e = 0$ otherwise. Subproblem $L_{z,s,x}(\lambda, \mu)$ can be stated as

$$
\begin{aligned}
L_{z,s,x}(\lambda, \mu) = \text{maximize} \quad & \sum_{k \in K} \sum_{(i,j) \in A_H} \overline{P}_{ijk} x_{ijk} - \sum_{i \in H} \overline{f}_i z_i - \sum_{i \in N} \overline{c}_i s_i \\
\text{subject to} \quad & (5.2) - (5.4), (5.7) - (5.9).
\end{aligned}
$$

We next show that $L_{z,s,x}(\lambda, \mu)$ can be reformulated as a QBP involving only $|N|$ binary variables.

5.1.1.1 Solution to Subproblem $L_{z,s,x}(\lambda, \mu)$

Given (5.2), for each $i \in H$ we can replace $s_i + z_i$ with a new binary variable h_i, with cost coefficient $F_i = \min\left\{c_i, \overline{f_i}\right\}$. For each $i \in N \setminus H$ we just define $h_i = s_i$ with coefficient $F_i = c_i$. We can now express $L_{z,s,x}(\lambda, \mu)$ as

$$
\begin{aligned}
L_{h,x}(\lambda, \mu) = \text{maximize} \quad & \sum_{k \in K} \sum_{(i,j) \in A_H} \overline{P}_{ijk} x_{ijk} - \sum_{i \in N} F_i h_i & & \\
\text{subject to} \quad & \sum_{(i,j) \in A_H} x_{ijk} \leq h_{o(k)} & k \in K & \quad (5.12) \\
& \sum_{(i,j) \in A_H} x_{ijk} \leq h_{d(k)} & k \in K & \quad (5.13) \\
& h_i \in \{0, 1\} & i \in N. &
\end{aligned}
$$

Given that (5.12) and (5.13) imply that, in an optimal solution to $L_{h,x}(\lambda, \mu)$ when both $h_{o(k)} = h_{d(k)} = 1$, commodity k will be routed via arc $(i_k, j_k) \in \arg\max\left\{\overline{P}_{ijk} : (i,j) \in A_H\right\}$, provided $\overline{P}_{i_k j_k k} > 0$. This allows us to project out the x_{ijk} variables and to rewrite $L_{h,x}(\lambda, \mu)$ only in terms of the h variables. For each $k \in K$, let $\overline{Q}_k = \max\left\{0, \max_{(i,j) \in A_H} \left\{\overline{P}_{ijk}\right\}\right\}$ and

$$
L_h(\lambda, \mu) = \max\left\{\sum_{k \in K} \overline{Q}_k h_{o(k)} h_{d(k)} - \sum_{i \in N} F_i h_i : h \in \{0, 1\}^{|N|}\right\}.
$$

We note that the only difference between the above expression for $L_h(\lambda, \mu)$ and a standard QBP formulation is that the former is stated on a directed graph, whereas QBP is typically stated on an undirected graph. Indeed, this difference can be easily overcome by redefining the cost coefficients as follows. For each pair $l, m \in N$, with $l < m$ let $k, \bar{k} \in K$ denote the two commodities with endnodes l and m, i.e. $o(k) = l$, $d(k) = m$, and $o(\bar{k}) = m$, $d(\bar{k}) = l$. By setting $Q_{lm} = \overline{Q}_k + \overline{Q}_{\bar{k}}$, we finally obtain the following QBP reformulation of $L_{z,s,x}(\lambda, \mu)$:

$$L_h(\lambda, \mu) = \max \left\{ \sum_{l,m \in N: l<m} Q_{lm} h_l h_m - \sum_{i \in N} F_i h_i : h \in \{0,1\}^{|N|} \right\}.$$

Although QBP is $\mathcal{NP}$-hard in the general case, there are some particular cases which are known to be polynomially solvable. Picard and Ratliff [107] show that when all cost coefficients of the quadratic term are non-negative, the QBP reduces to a *minimum cut problem* in an auxiliary network. Given that by definition $Q_{lm} \geq 0$ for all $l, m \in N$, $l < m$, for any feasible multiplier vector $(\lambda, \mu) \geq 0$, $L_h(\lambda, \mu)$ can thus be evaluated in polynomial time. For the sake of completeness, we next provide a sketch of the procedure to define the auxiliary network used for solving $L_h(\lambda, \mu)$ as a minimum cut problem. The reader is addressed to Picard and Ratliff [107] for further details.

Let $G^{Aux} = (V^{Aux}, A^{Aux})$ be a digraph where the set of nodes V^{Aux} contains the original nodes $l \in N$, denoted as v_l, plus an artificial source s_0 and an artificial sink s_n. The set of arcs A^{Aux} is characterized as follows. There is an arc (s_0, v_l) connecting the source with each $l \in N$ of capacity $\sum_{m \in N} Q_{lm}$, if $F_l \geq 0$, and $2(\sum_{m \in N} Q_{lm} - F_l)$, otherwise. There is also an arc (v_l, s_n) connecting each $l \in N$ with the sink of capacity F_l, if $F_l \geq 0$, and $\sum_{m \in N} Q_{lm} - F_l$, otherwise. For each pair $l, m \in N$ with $l < m$ there is also an arc (v_l, v_m) with capacity Q_{lm}. Finally, there is an arc (s_0, s_n) with capacity $K - \sum_{l,m \in N: l<m} Q_{lm}$, where $K = \sum_{l,m \in N: l<m} Q_{lm}$.

Any (s_0, s_n)-cut in the above network can be associated with a solution $\bar{h}$ to

$L_h(\lambda, \mu)$ (and vice-versa) as follows. If, for a given $l \in N$, (s_0, v_l) does not belong to the (s_0, s_n)-cut, then $\bar{h}_l = 1$ in the associated solution to $L_h(\lambda, \mu)$. Moreover, the arcs of the cut of the form (v_l, v_m) correspond to the pairs $l, m \in N$, $l < n$, where both $\bar{h}_l = \bar{h}_m = 1$. Furthermore, the value of the cut is precisely the value of $L_h(\lambda, \mu)$ for the solution $\bar{h}$ plus the constant K. An optimal solution to $L_h(\lambda, \mu)$ can thus be obtained by finding a minimum (s_0, s_n)-cut in G^{Aux}.

An optimal solution $(\bar{z}, \bar{s}, \bar{x})$ to $L_{z,s,x}(\lambda, \mu)$ in the original space can be retrieved from an optimal solution $(\bar{h}, \bar{y})$ to $L_h(\lambda^t, \mu^t)$ as follows. Note first that the only non-zero components of $\bar{x}$ are associated with commodities $k \in K$ with $\bar{h}_k = 1$. For each such commodity, we set $\bar{x}_{i_k j_k k} = 1$ if $\overline{P}_{i_k j_k k} > 0$, and 0 otherwise. As for the s variables, we set $\bar{s}_i = \bar{h}_i$ for each $i \in N \setminus H$ such that $F_i = c_i$, and 0 otherwise. Finally, we set $\bar{z}_i = \bar{h}_i$ for all $i \in H$ such that $F_i = \overline{f_i}$, and 0 otherwise.

Proposition 5.1 *For a given vector of multipliers (λ, μ), the lagrangean function $L_1(\lambda, \mu)$ can be solved in $O(|K||A_H| + |N|^3)$ time.*

Proof The solution of $L_y(\mu)$ has complexity $O(|E_H|)$, which is dominated by the evaluation of coefficients Q_{lm} for $l, m \in N$ for $l < m$, with complexity $O(|K||A_H|)$. Given that $|V^{Aux}| = O(|N|)$ and $|A^{Aux}| = O(|N|^2)$, the solution of $L_{z,s,x}(\lambda, \mu)$ can be obtained in $O(|N|^3)$ time using the max-flow algorithm given in [106] and the result follows. ■

5.1.1.2 Solution to the Lagrangean Dual

In order to obtain the best upper bound for PO_1 using $L_1(\lambda, \mu)$ we solve its associated lagrangean dual problem

$$(D_1) \qquad Z_{D_1} = \min_{(\lambda,\mu) \geq 0} \quad L_1(\lambda, \mu) = L_y(\mu) + L_h(\lambda, \mu).$$

We use subgradient optimization to solve D_1. The algorithm follows the usual iterative scheme $(\lambda^{t+1}, \mu^{t+1}) = (\lambda^t, \mu^t) + \varepsilon_t \gamma^t$, where ε_t is the step length and γ^t is a subgradient of L_1 at (λ^t, μ^t). A subgradient of L_1 at a given point (λ^t, μ^t) can be easily obtained from an optimal solution $(\bar{s}, \bar{z}, \bar{x}, \bar{y})$ to $L_1(\lambda^t, \mu^t)$. In particular,

$$\gamma^t = \left(\left(\sum_{j \in H} \bar{x}_{ijk} + \sum_{j \in H: i \neq j} \bar{x}_{jik} - \bar{z}_i \right)_{i,k}, (\bar{x}_{ijk} + \bar{x}_{jik} - \bar{y}_e)_{i,j,e} \right).$$

We update the step length according to $\varepsilon_t = \Lambda^t (L_1(\lambda^t, \mu^t) - \underline{\eta})/||\gamma^t||^2$, where $\underline{\eta}$ is a valid lower bound on the optimal value of PO_1 and Λ^t is a given parameter whose value is updated at certain iterations (see Section 6.5.1 for the specific details of our implementation). Algorithm summarizes the subgradient optimization algorithm that we apply. The algorithm terminates when one of the following criteria is met: (i) all the components of the subgradient are zero. In this case the current solution is proven to be optimal, (ii) the difference between the upper and lower bounds is bellow a threshold value, i.e., $|Z_{D_1} - Z^*| < \epsilon$, (iii) there is no improvement on the value of the upper bound after *niter* consecutive iterations, and (iv) the maximum number of iteration $Iter_{max}$ is reached.

Algorithm 5.1 Subgradient Optimization for PO_1

Initialization
  $Z_{D_1} = +\infty$; Initialize (λ^0, μ^0); Λ^0
  Let $\underline{\eta}$ be a lower bound on the optimal solution value
while Stopping criteria not satisfied **do**
  Solve $L_1(\lambda^t, \mu^t)$ and obtain an optimal solution $(\bar{s}, \bar{z}, \bar{x}, \bar{y})$
  if $L_1(\lambda^t, \mu^t) < Z_{D_1}$ **then**
    $Z_{D_1} \leftarrow L_1(\lambda^t, \mu^t)$
  end if
  Compute the subgradient γ^t
  Compute the step length $\varepsilon_t \leftarrow \Lambda^t (L_1(\lambda^t, \mu^t) - \underline{\eta})/||\gamma^t||^2$
  $(\lambda^{t+1}, \mu^{t+1}) \leftarrow (\lambda^t, \mu^t) + \varepsilon_t \gamma^t$
  $t \leftarrow t + 1$
end while

5.1.2 The Lagrangean Function for PO_2

Similarly to PO_1, in our LR of PO_2 we relax (5.5) and (5.6), incorporating them to the objective function with a multiplier vector (λ, μ). Moreover, we also relax (5.11), weighted with a multiplier vector π. An important property of this relaxation is that the domain of the lagrangean function

$$
\begin{aligned}
L_2(\lambda, \mu, \pi) = \sum_{k \in K} \pi_k + \max \quad & \sum_{k \in K} \sum_{(i,j) \in A_H} \overline{P}_{ijk} x_{ijk} - \sum_{i \in H} \overline{f}_i z_i - \sum_{i \in N} \overline{c}_i s_i - \sum_{e \in E_H} \overline{r}_e y_e \\
\text{s.t.} \quad & (5.2) - (5.4), (5.7) - (5.10),
\end{aligned}
$$

where

- $\overline{P}_{ijk} = \begin{cases} (R_k - F_{ijk})W_k - \lambda_{ik} - \lambda_{jk} - \mu_{\{i,j\}k} - \pi_k, & \text{if } (i \neq j) \\ (R_k - F_{iik})W_k - \lambda_{ik} - \pi_k, & \text{if}(i = j), \end{cases}$
- $\overline{c}_i = c_i - \sum_{k \in K: o(k) = i \text{ or } d(k) = i} \pi_k,$
- $\overline{f}_i = f_i - \sum_{k \in K} \lambda_{ik} - \sum_{k \in K: o(k) = i \text{ or } d(k) = i} \pi_k,$
- $\overline{r}_e = r_e - \sum_{k \in K} \mu_{ek},$

remains the same as in $L_1(\lambda, \mu)$ and the only difference is the objective function. It now consists of the constant $\sum_{k \in K} \pi_k$, which does not appear in $L_1(\lambda, \mu)$ but is irrelevant for the optimization, and two terms, one in the y space, which has exactly the same cost coefficients as in $L_1(\lambda, \mu)$, and another one in the (z, s, x) space, where the cost coefficients are now different from those of $L_1(\lambda, \mu)$. As before, $L_{z,s,x}(\lambda, \mu, \pi)$ can be transformed into a QBP on an undirected graph with non-negative cost coefficients. Thus, $L_2(\lambda, \mu, \pi) = \sum_{k \in K} \pi_k + L_y(\mu) + L_{z,s,x}(\lambda, \mu, \pi)$ can also be solved in polynomial time by transforming $L_{z,s,x}(\lambda, \mu, \pi)$ into a min-cut problem.

Similarly to PO_1, in order to obtain the best upper bound for PO_2 using $L_2(\lambda, \mu, \pi)$

we solve its associated lagrangean dual problem

$$(D_2) \qquad Z_{D_2} = \min_{(\lambda,\mu,\pi)\geq 0} \quad L_2(\lambda,\mu,\pi) = \sum_{k\in K} \pi_k + L_y(\mu) + L_{z,s,x}(\lambda,\mu,\pi).$$

We apply a subgradient optimization algorithm similar to Algorithm (5.1) for solving the lagrangean dual. Details are omitted.

5.1.3 Lower Bounds from Primal Solutions

In this section we explain how feasible solutions are constructed to obtain valid lower bounds for PO_1 and PO_2. In particular, we exploit the information generated from the integer solutions to the lagrangean duals at some iterations of the corresponding subgradient optimization algorithms.

5.1.3.1 A Primal Heuristic for PO_1

Let $(\bar{s}, \bar{z}, \bar{x}, \bar{y})$ denote the solution to $L_1(\lambda, \mu)$ at the current iteration. Since in $L_1(\lambda, \mu)$ the sets of constraints (5.5) and (5.6) are relaxed, the solution $(\bar{s}, \bar{z}, \bar{x}, \bar{y})$ may not be feasible for PO_1. We next describe a simple heuristic to obtain a feasible solution $(\hat{s}, \hat{z}, \hat{x}, \hat{y})$ to PO_1.

The initial solution is the outcome of $L_1(\lambda, \mu)$ but with all routing variables at value zero, i.e., initially, $(\hat{s}, \hat{z}, \hat{x}, \hat{y}) = (\bar{s}, \bar{z}, \mathbf{0}, \bar{y})$. This solution contains a set of open hubs, a set of served nodes, and a set of active hub edges. Given that $L_y(\mu)$ and $L_h(\lambda, \mu)$ are independently solved, some hub edges could be associated with closed hub nodes. In order to guarantee the feasibility of the edge variables $\hat{y}$, we close all hub edges that do not have both end-nodes open as hubs. That is, for each $e = \{i, j\} \in E_H$ such that $\hat{z}_i = 0$ or $\hat{z}_j = 0$, we set $\hat{y}_e = 0$. Finally, we select the set of commodities to be served and their routing paths as follows. For each commodity $k \in K$ with both end-nodes activated, we identify the most "attractive" path among the ones using open hub edges (and thus open hub nodes), and route commodity k through it only if it is profitable. That is, for each $k \in K$ with $\hat{s}_{o(k)} + \hat{z}_{o(k)} = \hat{s}_{d(k)} + \hat{z}_{d(k)} = 1$, let $e(k) \in \arg\max \{R_k - F_{ek} : \hat{y}_e = 1, e \in E_H\}$. If $R_k - F_{e(k)k} > 0$, then $\hat{x}_{e(k)k} = 1$, and 0 otherwise.

5.1.3.2 A Primal Heuristic for PO_2

To obtain feasible solutions to PO_2 we apply a two phase heuristic. The first phase is an adaptation of the heuristic applied to PO_1. Since the quality of the PO_2 solutions produced by such first phase is usually quite weak, we apply a second phase to improve the outcome of the first phase.

The first phase starts with $(\hat{s}, \hat{z}, \hat{x}, \hat{y}) = (\bar{s}, \bar{z}, \mathbf{0}, \bar{y})$, and then closes all hub edges that do not have both end-nodes open as hubs. The set of commodities to be served and their routing paths are selected as follows. In order to satisfy constraints (5.11), for each commodity with both end-nodes activated we identify the best path among the ones using open hub edges, and route such commodity through it regardless if it is profitable or not. That is, for each $k \in K$ with $\hat{s}_{o(k)} + \hat{z}_{o(k)} = \hat{s}_{d(k)} + \hat{z}_{d(k)} = 1$, let $e(k) \in \arg\max \{R_k - F_{ek} : \hat{y}_e = 1, e \in E_H\}$ and set $\hat{x}_{e_k k} = 1$ (independently of the sign of $R_k - F_{e_k k}$). Let $\hat{\eta}$ denote the objective value of $(\hat{s}, \hat{z}, \hat{x}, \hat{y})$.

The second phase is a three-step procedure that aims at improving the output of Phase 1 by: (i) activating additional hub edges, (ii) adding new served nodes, and (iii) closing open hub nodes.

(i) For each non-activated hub edge $e = \{i, j\} \in E_H$ but with both endnodes open as a hubs, we compute the variation in the objective function if hub edge e were activated and the commodities re-routed accordingly. Then, the hub edge is activated if the estimation is positive. That is, we consider in an arbitrary order each $e = \{i, j\} \in E_H$ with $\hat{y}_e = 0$ and $\hat{z}_i = \hat{z}_j = 1$, and for each $k \in K$ we set

$$\Delta_k = \begin{cases} \max\{R_k - F_{ijk}, 0\} & \text{if } \sum_{(i',j') \in A} \hat{x}_{i'j'k} = 0, \\ \max\{F_{e_k k} - F_{ijk}, 0\} & \text{if } \hat{x}_{e_k k} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

If $\Gamma_e = \sum_{k \in K} \Delta_k - r_{ij} > 0$, then $\hat{y}_e = 1$ and $\hat{\eta} = \hat{\eta} + \Gamma_e$.

(ii) For each node $i \in N$ that is not served, we compute the variation in the objective function if node i was served and its associated commodities routed. The node is then served if the estimation is positive. We denote as $\hat{A} = \{(i', j') \in A \mid \hat{y}_{i'j'} = 1\}$ the set of arcs whose associated hub edges are active in the current solution. We consider in an arbitrary order each $i \in N$ with $\hat{s}_i = 0$, and for each $k \in K$ we define

$$\Delta_k = \begin{cases} \max\{R_k - \min_{(i',j') \in \hat{A}} \{F_{i'j'k}\}, 0\}, & \text{if } o(k) = i \text{ and } \hat{s}_{d(k)} + \hat{z}_{d(k)} = 1, \\ \max\{R_k - \min_{(i',j') \in \hat{A}} \{F_{i'j'k}\}, 0\}, & \text{if } d(k) = i \text{ and } \hat{s}_{o(k)} + \hat{z}_{o(k)} = 1, \\ 0 & \text{otherwise.} \end{cases}$$

If $\Gamma_i = \sum_{k \in K} \Delta_k - c_i > 0$, then $\hat{s}_i = 1$ and $\hat{\eta} = \hat{\eta} + \Gamma_i$.

(iii) For each hub $i \in H$ that is open we compute the variation in the objective function if hub i was closed and its associated commodities re-routed. The hub node is then closed if the estimation is positive. We denote as $\hat{E}(i) = \{\{i', j'\} \in E \mid \hat{y}_{i'j'} = 1, \text{ and } i' = i \text{ or } j' = i\}$ the set of active hub edges incident to i. We consider in an arbitrary order each $i \in N$ with $\hat{z}_i = 1$, and for each

$k \in K$ we define

$$\Delta_k = \begin{cases} -(R_k - F_{i(k)j(k)k}), & \text{if } \hat{x}_{i(k)j(k)k} = 1 \text{ and } \{i(k), j(k)\} \in \hat{E}(i), \\ 0, & \text{otherwise.} \end{cases}$$

If $\Gamma_i = \sum_{k \in K} \Delta_k + f_i + \sum_{(i',j') \in \hat{E}(i)} r_{i'j'} y_{i'j'} > 0$, then $\hat{z}_i = 0$, $\hat{y}_e = 0$ for all $e \in \hat{E}(i)$, and $\hat{\eta} = \hat{\eta} + \Gamma_i$.

5.2 Variable Elimination Techniques

One of the main challenges of the MIP formulations we use to model PO_1 and PO_2 are the very large number of variables and constraints that these require, even for small-size instances. By slightly increasing the size of the instances, the number of variables in the formulations becomes so large that considerable amounts of computing time and memory are required to solve them with a commercial solver. In the previous sections, we have presented LRs whose lagrangean functions can be solved efficiently in polynomial time. Still, any reduction on the size of the formulations is highly beneficial for attaining a higher efficiency. In our algorithmic framework we reduce the size of the instances by means of three effective procedures: (i) *Preprocessing*, valid only for PO_1, which is applied prior to the solution of D_1, and aims at eliminating variables and constraints; (ii) *Reduction Tests*, valid for both PO_1 and PO_2, which eliminate variables based on the information obtained from the lagrangean functions; and, (iii) *Post-processing*, which further eliminates variables, both for PO_1 and PO_2, using jointly information from the reduction tests and valid lower bounds.

5.2.1 Preprocessing

In the case of PO_1, it is possible to a priori eliminate routing variables x that will not make part of an optimal solution by using the following property.

Property 5.1 [*Alibeyg et al. [6]*] *There is an optimal solution to formulation* (5.1) – (5.10) *where* $x_{ijk} = 0$, *for all* $k \in K$ *and* $(i,j) \in A_H$, *with* $R_k - F_{ijk} \leq 0$.

The use of Property 5.1 in PO_1 allows to eliminate all routing variables with unprofitable arcs. That is, for each $k \in K$ we set $x_{ijk} = 0$ for all $(i,j) \in A_H$ such that $R_k - F_{ijk} \leq 0$. Since we are assuming that routing costs are symmetric, if $(i,j) \in A_H$ is unprofitable so is $(j,i) \in A_H$. Thus, when we set $x_{ijk} = 0$ we not only set $x_{jik} = 0$, but also eliminate the corresponding constraint (5.6), as it becomes unnecessary. Hence, for each $k \in K$ we restrict the set of potential candidate arcs for routing it to the arcs that are profitable for this commodity, $A_k = \{(i,j) \in A_H \mid R_k - F_{ijk} > 0\}$. Let also E_k denote the corresponding set of profitable hub edges for k.

Since the above elimination affects variables and constraints of PO_1, it can also be extended to the lagrangean function $L_1(\lambda, \mu)$, where only arcs and edges of A_k and E_k, respectively, will now be considered. We also note that the reduction on the number of constraints (5.6) of PO_1 causes a significant reduction on the number of lagrangean multipliers μ in $L_1(\lambda, \mu)$.

An important consequence of (5.11), is that Property 5.1 does not hold for PO_2 as all the commodities whose O/D nodes are active must be served, independently of whether or not there are profitable arcs for them.

5.2.2 Reduction Tests

Another way of reducing the size of the formulations is to develop tests to eliminate variables based on information generated from the LR. We next develop two such tests based on sufficient conditions that determine if a potential hub will be closed or if a hub edge will not be activated in an optimal solution of a given HNDPP instance. These tests are valid for both PO_1 and PO_2, since they are based on the information produced by their respective lagrangean functions L_1 and L_2. We will not distinguish the case of PO_1 from the case of PO_2, since the structure of the terms that construct the lagrangean functions $L_1(\lambda, \mu)$, and $L_2(\lambda, \mu)$ is exactly the same and the rationale of the tests is also the same in both cases. Similar reduction tests have been successfully applied to other HLPs (Contreras et al. [35, 36]).

5.2.2.1 Elimination of Potential Hub Nodes

The idea of this test is to use the lagrangean function to obtain upper bounds on the profit that would be obtained in the original problem if a given node $l \in H$ is chosen to become a hub. If this estimated profit is less than the value of the best known solution to the original problem, then node l will not be a hub in any optimal solution. Let $\hat{L}_h(\lambda, \mu, S_z)$ denote the value of $L_h(\lambda, \mu)$ when restricted to a set of potential hub nodes $S_z \subseteq H$, and its associated set of hub arcs $A_S = \{(i,j) \in A_H : i, j \in S_z\}$. That is,

$$\begin{aligned} \hat{L}_h(\lambda, \mu, S_z) = \text{maximize} \quad & \sum_{k \in K} \overline{Q}_k h_{o(k)} h_{d(k)} - \sum_{i \in N} F_i h_i \\ \text{subject to} \quad & h_i \in \{0, 1\} \qquad i \in N, \end{aligned}$$

where $\overline{Q}_k = \max\left\{0, \max_{(i,j) \in A_S} \left\{\overline{P}_{ijk}\right\}\right\}$. Let $\hat{L}^l_h(\lambda, \mu, S_z)$ denote the optimal value of $\hat{L}_h(\lambda, \mu, S_z)$ with the additional constraint that hub l is open, i.e. $z_l = 1$. The only difference between $\hat{L}_h(\lambda, \mu, S_z)$ and $\hat{L}^l_h(\lambda, \mu, S_z)$ is that, in the latter, node l is now a priori activated as an open hub. This means that now $F_l = \{f_l\}$ and $h_l = 1$. The following result can be used to perform variable elimination tests on hub location decisions.

PROPOSITION 5.2 *Let $\underline{\eta}$ be a valid lower bound on the optimal value of PO_1 (resp. PO_2), $S_z \subseteq H$ a given set of potential hub nodes, $l \in S_z$ a specific potential hub node, and (λ, μ) a multiplier vector. If $\Delta_l(\lambda, \mu, S_z) = L_y(\mu) + \hat{L}_h^l(\lambda, \mu, S_z) < \underline{\eta}$, then $z_l = 0$ in any optimal solution.*

Proof The result follows since $\Delta_l(\lambda, \mu, S_z)$ is an upper bound on the objective function value of any solution in which a hub is located at node l. Therefore, if $\Delta_l(\lambda, \mu, S_z) < \underline{\eta}$, no optimal solution will have an open hub at $l \in S_z$, so $z_l = 0$. ■

We use this result as follows. The subgradient optimization is initialized with all possible nodes as candidate hub nodes, that is $S_z = H$. Once the deviation between the upper and lower bounds becomes smaller than a given threshold ϵ_{Test} after a number of iterations of the subgradient optimization algorithm, we apply the reduction test for each $l \in S_z$ that is not active in the current subgradient optimization iteration, i.e. $\bar{s}_l = \bar{z}_l = 0$, every $niter_{\text{Test1}}$ iterations. If $\Delta_l(\lambda, \mu, S_z) < \underline{\eta}$, we eliminate l from the set of candidate hub nodes, i.e. $S_z \leftarrow S_z \setminus \{l\}$. According to Proposition 5.2, by applying the test in this way we ensure that S_z always contains an optimal set of hubs.

When some node is eliminated from S_z, not only the associated z_l variable is eliminated from the LR, but also several routing variables x_{ijk} associated with node l. This plays an important role in the computational complexity for solving $L_{z,s,x}(\lambda, \mu)$, as the running time is now dependent of the size of A_S, instead of A_H. That is, the lagrangean functions $L_1(\lambda, \mu)$ and $L_2(\lambda, \mu, \pi)$ can now be solved in $O(|K||A_S| + |N|^3)$ time. Another important consequence of eliminating one variable z_l is that we can remove $|K|$ constraints (5.5) from the solution process, which in turn significantly reduces the solution space of the lagrangean dual problems D_1 and D_2.

5.2.2.2 Elimination of Potential Hub Edges

An immediate consequence of the elimination of potential hub nodes is that if the two end-nodes of a hub edge have been eliminated, then the hub edge can also be eliminated. That is, we set $y_e = 0$ for all $e = \{i, j\} \in E_H$ where z_i and z_j have been set to zero.

Additional hub edges can be further eliminated by estimating an upper bound on the objective function value if a hub edge is activated. This bound can be easily computed after setting at value one the variable associated with the candidate edge in $L_y(\mu)$. In particular, for a set of candidate hub edges $S_y \subseteq E_H$, and a hub edge $\bar{e} \in S_y$, let $\hat{L}_y^{\bar{e}}(\mu, S_y)$ denote the optimal value of $L_y(\mu)$ restricted to S_y when hub edge $\bar{e}$ has been activated

$$\hat{L}_y^{\bar{e}}(\mu, S_y) = -\overline{r}_{\bar{e}} - \sum_{e \in S_y \setminus \{\bar{e}\}} \min\{0, \overline{r}_e\}.$$

The following result can be used to perform reduction tests on hub edge activation decisions.

Proposition 5.3 *Let $\underline{\eta}$ be a valid lower bound on the optimal value of PO_1 (resp. PO_2), $S_y \subseteq E_H$ a given set of potential hub edges, $\bar{e} \in S_y$ a specific potential hub edge, and (λ, μ) a multipliers vector. If $\Delta_{\bar{e}}(\lambda, \mu, S_y) = \hat{L}_y^{\bar{e}}(\mu, S_y) + L_{z,s,x}(\lambda, \mu) < \underline{\eta}$, then $y_{\bar{e}} = 0$ in any optimal solution.*

Proof The result follows since $\Delta_{\bar{e}}(\lambda, \mu, S_y)$ is an upper bound on the objective function value of any solution in which a hub edge $\bar{e}$ is activated. Therefore, if $\Delta_{\bar{e}}(\lambda, \mu, S_y) < \underline{\eta}$, hub edge $\bar{e}$ will not be activated in any optimal solution. ■

Reduction tests for hub edges are applied immediately after reduction tests for hub nodes. Let E_{H^0} denote the set of edges eliminated in the first phase of the hub elimination test. For the second phase, we set $S_y = E_H \setminus E_{H^0}$, and apply the elimination test to each candidate hub edge $\bar{e}$ in the updated set S_y. Then, if $\Delta_{\bar{e}}(\lambda, \mu, S_y) < \underline{\eta}$, we eliminate $\bar{e}$ from the set of candidate hub edges, i.e. $S_y \leftarrow S_y \setminus \{\bar{e}\}$. According to Proposition 5.3, applying the test in this way ensures that S_y always contains an optimal set of hub edges.

In addition, once a y_e variable has been eliminated, we can also remove $|K|$ constraints (5.6) from the solution process, which causes a considerable reduction of the solution space of the lagrangean dual problems D_1 and D_2.

5.2.3 Post-processing

This is a simple procedure where we use information obtained from the reduction tests for hub edges to update the set of candidate hub edges A_k, so as to further eliminate additional routing variables x_{ijk}. In particular, for each $k \in K$, we remove from its set of profitable edges A_k any hub edge that has been fixed to zero during the hub edge elimination test. That is, any variable x_{ijk} associated with an arc removed from A_k is permanently set at value 0. Given that the amount of time for updating this set is significant, this procedure is only applied every $niter_{\text{Test2}}$ applications of the tests.

5.3 An Exact Solution Algorithm

In this section we present the complete algorithmic framework used for solving problems PO_1 and PO_2 to optimality. Its core component is a branch-and-bound method in which, at every node of the enumeration tree, we obtain lower and upper bounds by using the subgradient optimization algorithms and the primal heuristics presented in Section 5.1. We also apply a partial enumeration phase to enhance the application of the reduction tests. This phase is applied at the beginning of the branch-and-bound procedure right after solving the root node. It is particularly useful to reduce the number of variables to branch on, and to reduce the size of the subproblems in the nodes of the tree. Contreras et al. [35, 36] provide some examples of successful implementations of branch-and-bound algorithms based on lagrangean bounds used to solve HLPs. We next describe the partial enumeration and then the overall branch-and-bound algorithm.

5.3.1 Partial Enumeration

The partial enumeration works as follows. Let H^0 and H^1 denote the set of potential hubs that have been already fixed at value 0 and 1, respectively. Since, the partial enumeration is applied after solving the root node, initially we have $H^0 = H \setminus S_z$ and $H^1 = \emptyset$. Then, for each hub not yet considered $i \in H \setminus (H^0 \cup H^1)$, we temporarily fix $z_i = 1$ and solve the resulting lagrangean dual problem using an iteration limit of $Iter_{max} = 80$. If the resulting upper bound ub_i^1 is smaller than the current best lower bound, we set $z_i = 0$ (as well as the the related y variables) and we update the set H^0, accordingly. Otherwise, we temporarily fix $z_i = 0$ and solve the resulting lagrangean function. If the obtained upper bound ub_i^0 is smaller than the current best lower bound, we set $z_i = 1$ and update the set H^1. At the end of the partial enumeration we re-optimize the lagrangean dual problem using an iteration limit of $Iter_{max} = 1,000$ to further improve the bound of the root node.

5.3.2 Branch and Bound

We now present a branch-and-bound algorithm in which valid lower and upper bounds are constructed at each node of the enumeration tree with the proposed LR. The tree is structured in three levels: the first level where we branch on the z variables (hub nodes); the second level where we branch on the s variables (served nodes); and a third level, where we branch on the y variables (hub edges). Each level is explored according to a depth first search policy in which the 1-branch is explored first. No subsequent level is explored until all the nodes of the previous level have been explored. The strategy for selecting of the branching variable at each node of the first level is guided by the output of the partial enumeration. In particular, for each potential hub node not yet fixed $i \in H \setminus (H^0 \cup H^1)$, we compute $\delta_i = \min\{ub_i^0, ub_i^1\}$. At any point during the first level, the branching variable z_j is the selected as $j \in \arg\max\{\delta_i \mid i \in H \setminus (H^0 \cup H^1)\}$.

After finishing branching on the z variables, we continue branching on the s variables. For each *active* node at the end of the first level, we set $s_i = 0$ for all $i \in H^1$, and continue branching on the remaining s_i variables with $i \in N \setminus H^1$. During the second level, branching variables are arbitrarily selected. If some nodes remain active after completing the branching on the z and s variables, then the branching on the hub edge variables y begins. For each active node at the end of the second level, we set $y_e = 0$ for all $e \in H^0 \times H^0$. During the third level, branching variables are also arbitrarily selected. Given that the lagrangean dual problems are only approximately solved with Algorithm (5.1), it may happen that there are some active nodes after finishing branching in the third level. In this case, the remaining routing subproblems can be efficiently solved to optimality as described in Section 5.1.3. Finally, at each node of the enumeration tree, we use the optimal dual solution to the lagrangean dual of its parent node, as the initial solution to the current lagrangean dual, instead of starting from scratch.

5.4 Computational Experiments

We have run extensive computational experiments to analyze and compare the performance of the lagrangean relaxation, the reduction tests and the exact algorithm, both for PO_1 and PO_2. All algorithms were coded in C and run on an HP station with an Intel Xeon CPU E3-1240V2 processor at 3.40 GHz and 24 GB of RAM under Windows 7 environment. In all the experiments the maximum CPU time was set to 86,400 seconds (one day).

The benchmark instances are the same we used in Alibeyg et al. [6]. Most of the data comes from the well-known CAB data set of the US Civil Aeronautics Board from *http://www.researchgate.net/publication/269396247_cab100_mok*. This data provides Euclidean distances d_{ij} between 100 cities in the US and the values of the service demand W_k between each pair of cities. We have considered instances with $n \in \{25, 30, 40, 50, 60, 70, 80, 90, 100\}$ and $\alpha \in \{0.2, 0.5, 0.8\}$. Since the CAB instances do not provide setup costs f_i for opening hubs, we use the ones generated by de Camargo et al. [45]. For the remaining missing information, we use the following additional data that we generated for the computational experiments of Alibeyg et al. [6]. The setup costs c_i for served nodes are $c_i = \nu f_i$, where $\nu = 0.1$ unless otherwise stated. The setup costs for activating hub edges are $r_e = \tau(f_i + f_j)/2$, where $\tau \in \{0.3, 0.6, 0.4\}$ is a parameter used to model the increase (decrease) in setup costs on the hub edges when considering smaller (larger) discount factors α. The revenues R_k for routing commodities are randomly generated as $R_k = \varphi \sum_{(i,j)\in A_H} F_{ijk}/|A_H|$, where φ is a continuous random variable following a uniform distribution $\varphi \sim U[0.25, 0.35]$. The collection and distribution factors are $\chi = \delta = 1$.

5.4.1 Implementation Details

After some fine-tuning, we set the following parameter values for the subgradient optimization algorithm. The maximum number of iterations, $Iter_{max}$, is 3,000 at the root node, 80 at each application of the partial enumeration, and 1,000 in the re-optimization after the partial enumeration. At each node of the branch and bound tree we set $Iter_{max} = 200$. The additional parameters that are used for the termination criteria of the subgradient optimization are the following: the threshold between the upper and lower bounds is $\epsilon = 10^{-6}$ (termination criterion *ii*); and the number of consecutive iterations without improvement is $niter = 1,500$ (termination criterion *iii*). We set $(\lambda^0, \mu^0, \pi^0) = (95, 85, 85)$ as the initial multipliers vector. The parameter Λ^t that is used in the computation of the step length is initialized to 7 and halved every 500 iterations, provided that the % gap is less than %50, and is reset to its initial value whenever it becomes smaller than 2. We apply the heuristics every 10 iterations of the subgradient algorithm. We use $\underline{\eta} = 0$ as the initial lower bound. This value is updated and recorded for further applications of the subgradient and the elimination tests, whenever the heuristic improves the incumbent solution. We apply the elimination tests every $niter_{\text{Test1}} = 100$ and iterations of subgradient optimization and the post-processing every $niter_{\text{Test2}} = 700$ applications of the tests. Both the tests and post-processing are only applied if the percentage gap between the upper and lower bounds is below the threshold $\epsilon_{\text{Test}} = \%5$.

5.4.2 Comparison of the Exact Algorithmic Framework and CPLEX

We next analyze and compare the performance of the general purpose solver CPLEX 12.6.3 using a traditional (deterministic) branch-and-bound algorithm and our exact algorithmic framework for PO_1 and PO_2. The application of CPLEX to PO_1 and PO_2 is referred to as $CPLEX_1$ and $CPLEX_2$, respectively, whereas our exact algorithms for PO_1 and PO_2 are referred to as BB_1 and BB_2, respectively. All parameters have been set to their default values both in $CPLEX_1$ and $CPLEX_2$. It is worth mentioning that, similar to Alibeyg et al. [6], Property 5.1 is also applied to $CPLEX_1$.

Figures 5.1 and 5.2 give performance profiles of $CPLEX_1$ (dotted line) and BB_1 (solid line), and of $CPLEX_2$ (dotted line) and BB_2 (solid line), respectively. In each figure, the horizontal axis refers to computing times while the vertical axis refers to number of instances. The points (x, y) depicted in the lines on each figure indicate the total number of instances y optimally solved within the computing time x. In general, small size instances can be solved rather fast both with CPLEX and our exact algorithms, but the performance decreases as the sizes of the instances increase. This is why in the two lines depicted in each figure the vertical values increase fast at the beginning but slow down after a while. Throughout the considered one-day time interval, BB_1 is consistently better than $CPLEX_1$. Moreover, within the time limit, BB_1 is able to optimally solve all 27 instances, while $CPLEX_1$ solves 18. The effect of the additional set of constraints (5.11) on the difficulty for solving PO_2 is evident, and both $CPLEX_2$ and BB_2 are slower than their respective counterparts for PO_1. In any case, BB_2 still outperforms $CPLEX_2$ and, within the time limit, it is able to optimally solve 21 instances instead of the 15 instances optimally solved by $CPLEX_2$.

Tables 5.1 and 5.2 give information of the bounds at the root nodes and of the complete enumeration trees of the compared solution methods for PO_1 and PO_2, respectively. The first two columns of each table give some instances data: α, the discount factor on hub edges, and $|N|$, the number of nodes. The next two columns,

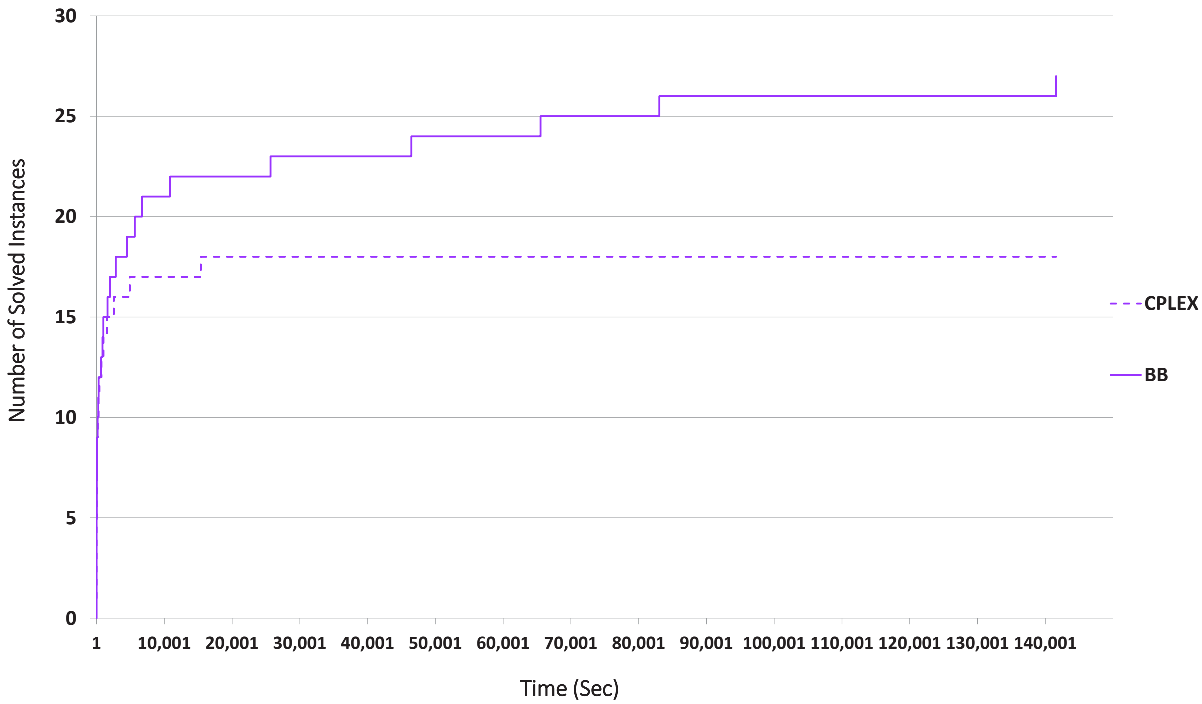


Figure 5.1: Performance profile of CPLEX and BB_1 for PO_1.

under the heading *% Dev*, give the percentage deviations of the upper bounds produced by the employed relaxations: Linear Programming (LP) in the case of CPLEX and lagrangean in our proposed solution algorithms. These deviations have been computed as $100(v_{RP} - v^*)/v^*$, where v_{RP} denotes the upper bound produced by the relaxed problem (LP or lagrangean) and v^* the optimal or best-known value. The next two columns under the header *Nodes* give the number of nodes explored in the enumeration trees. The three columns under the header *Time (sec)* give computing times in seconds. The first of these columns gives the total time consumed by CPLEX, and the other two refer to our exact solution algorithms: *LR* for the computing time for solving the lagrangean Dual at the root node and *BB* for the overall

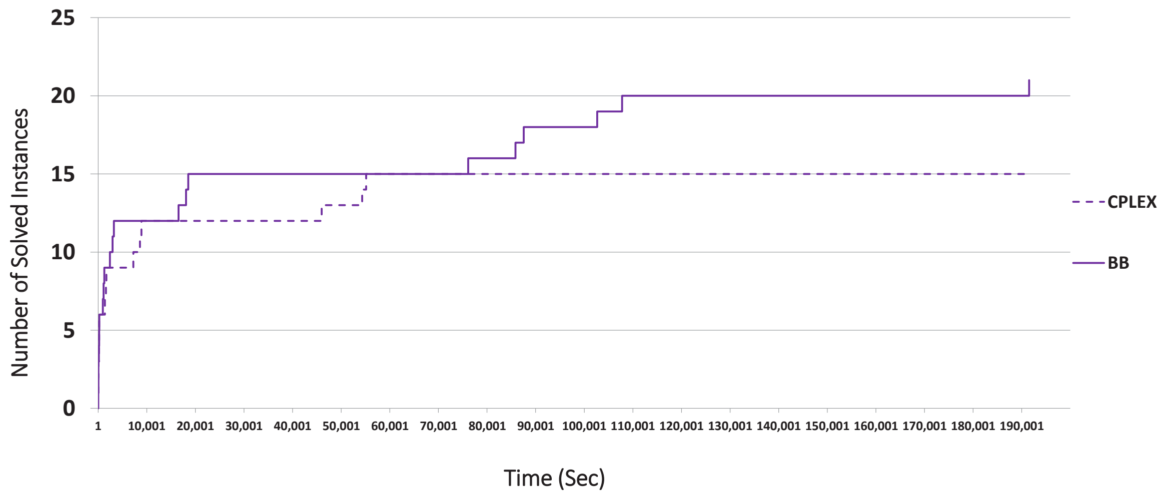


Figure 5.2: Performance profile of CPLEX and BB_2 for PO_2.

time needed to optimally solve each instance. Finally, the last two columns RT and PE give the percentage of hubs fixed with the reduction tests (see Section 5.2.2) and with the partial enumeration (see Section 5.3.1), respectively. That is, the entries of these columns are computed as $100(FH/|H|)$, where FH is the number of hubs fixed in each case. The entries corresponding to instances that could not be handled by CPLEX because of insufficient memory are filled with the text *mem*. When an instance could not be solved to optimality within the time limit, the corresponding entry in the column of the computing times is *time* followed by the percentage optimality gap at termination, in parenthesis.

The results of Table 5.1 confirm the superiority of BB_1 over $CPLEX_1$. On the one hand, even if formulation (5.1)-(5.10) produces, in general, very tight LP bounds, it has a very strict limitation in terms of the size of the instances that can be handled by $CPLEX_1$. It is true that the LP gap of $CPLEX_1$ is always % 0.00 for the 18 instances with up to 70 nodes. However, the quality of these bounds contrasts with the

α	|N|	% Dev		Nodes		Time (sec)			% Fixed hubs	
		LP	LR	CPLEX	BB	CPLEX	LR	BB	RT	PE
0.2	25	0.00	0.00	0	0	3.00	1.80	1.80	0.00	0.00
	30	0.00	0.00	0	0	9.64	5.63	5.63	0.00	0.00
	40	0.00	0.04	0	0	126.12	35.78	45.66	0.00	100.00
	50	0.00	0.11	0	0	513.20	81.94	120.31	0.00	100.00
	60	0.00	0.16	0	160	2370.97	181.69	349.34	0.00	98.33
	70	0.00	0.27	0	218	10460.44	391.55	850.60	0.00	98.57
	80	mem	0.26	mem	340	mem	736.90	1641.43	0.00	98.75
	90	mem	0.36	mem	1318	mem	1298.37	4129.42	1.11	97.78
	100	mem	0.64	mem	6738	mem	1970.64	19048.79	0.00	90.00
0.5	25	0.00	0.00	0	0	1.60	0.36	0.36	8.00	8.00
	30	0.00	0.00	0	0	4.55	3.85	3.85	13.33	13.33
	40	0.00	0.03	0	0	21.53	13.67	19.08	15.00	100.00
	50	0.00	0.10	0	0	75.90	45.63	59.83	18.00	100.00
	60	0.00	0.13	0	162	309.06	110.09	189.35	18.33	98.33
	70	0.00	0.27	0	570	1006.20	248.81	626.97	7.14	97.14
	80	mem	0.40	mem	600	mem	446.79	1170.51	12.50	92.50
	90	mem	1.46	mem	3676	mem	634.52	14824.07	0.00	67.78
	100	mem	1.28	mem	3666	mem	1161.68	17537.49	1.00	77.00
0.8	25	0.00	0.02	0	0	1.36	1.83	2.34	24.00	100.00
	30	0.00	0.01	0	0	3.62	3.42	4.71	20.00	100.00
	40	0.00	0.03	0	0	15.42	9.61	13.55	22.50	100.00
	50	0.00	0.36	0	128	44.93	22.09	43.13	24.00	94.00
	60	0.00	0.27	0	132	121.70	46.91	100.51	25.00	96.67
	70	0.00	0.51	0	166	293.20	155.21	386.88	2.86	85.71
	80	mem	0.53	mem	792	mem	267.48	1085.56	5.00	88.75
	90	mem	0.88	mem	24214	mem	471.32	20789.11	6.67	88.89
	100	mem	0.87	mem	52372	mem	698.48	58546.99	11.00	89.00

Table 5.1: Results of exact algorithm using CAB instances for PO_1

insufficiency of the 24 GB of memory available: none of the remaining nine instances with 80-100 nodes could even be uploaded to the CPLEX solver. In contrast, our lagrangean Dual D_1 is highly effective in all cases, as it is able to produce tight bounds for all 27 instances using only 2 GB of memory for the largest considered instances with up to 100 nodes. In some cases achieving convergence when solving D_1 was very difficult, and the actual upper bound Z_{D_1} could not be attained. This explains why in some cases % *Dev* is 0.00 for *LP*, but it is strictly positive for *LR*. Still, the bounds we could obtain with D_1, together with the quality of the heuristic applied within subgradient optimization, assess its effectiveness. The optimality of four out of the 27 PO_1 instances was already proven after solving D_1 at the root node. For these instances, the heuristic applied within subgradient optimization produced a feasible solution with the same value as that of the upper bound. For 16 and seven of the remaining 23 instances, the percent deviation after solving D_1 was below % 0.5 and % 1.46, respectively.

The columns under *Time (sec)* relative to D_1 and BB_1 confirm that these good results were obtained with a small computing effort. On the one hand, BB_1 is able to solve all 27 instances to proven optimality within the CPU time limit, while $CPLEX_1$ is able to solve only instances with up to 70 nodes. On the other hand, BB_1 is, in general, much faster than $CPLEX_1$ on the 18 instances that could be solved by $CPLEX_1$, particularly for the instances with the smallest discount factor $\alpha = 0.2$. Note that BB_1 is faster than $CPLEX_1$ in 15 of out of the 18 such instances. Finally, the last two columns of Table 5.1 assess the effectiveness of the reduction tests and, particularly, of the partial enumeration: in 21 benchmark instances it was possible to fix more than % 80 of the hubs. The side effect of the good performance of these tests is that no enumeration is required in 11 out of the 27 tested instances.

The results of Table 5.2 confirm that, as mentioned, solving PO_2 is more challenging than solving PO_1 both for CPLEX and for our exact algorithmic framework. In any case, the superiority of our exact algorithm over CPLEX becomes even more evident for PO_2 than for PO_1. In particular, with the 24 GB of memory available, $CPLEX_2$ could only handle the 15 instances with up to 60 nodes, all of which were optimally solved at the root node. However, it was not possible to even upload to CPLEX any of the remaining 12 instances with 70-100 nodes. The reason for which $CPLEX_2$ could handle fewer instances than $CPLEX1$ is that Property 5.1 no longer

α	$\|N\|$	*% Dev*		*Nodes*		*Time (sec)*			*% Fixed hubs*	
		LP	*LR*	*CPLEX*	*BB*	*CPLEX*	*LR*	*BB*	*RT*	*PE*
0.2	25	0.00	0.00	0	0	25.15	28.23	28.23	12.00	12.00
	30	0.00	0.07	0	30	130.19	61.34	81.31	13.33	93.33
	40	0.00	0.20	0	166	1162.89	486.23	735.45	2.50	95.00
	50	0.00	0.19	0	150	5557.70	552.99	1153.80	4.00	98.00
	60	0.00	0.71	0	872	37065.80	7015.70	13311.15	0.00	83.33
	70	mem	0.97	mem	2610	mem	7550.15	57608.85	0.00	64.29
	80	mem	1.15	mem	259	mem	15347.87	time (0.02)	0.00	0.00
	90	mem	1.40	mem	335	mem	27938.78	time (0.92)	43.33	47.77
	100	mem	1.44	mem	573	mem	14092.6	time (0.35)	0.00	64.00
0.5	25	0.00	0.04	0	0	10.24	21.04	23.32	40.00	100.00
	30	0.00	0.03	0	0	31.81	41.99	46.45	36.67	100.00
	40	0.00	0.14	0	0	216.29	138.28	162.76	37.50	100.00
	50	0.00	0.29	0	100	1364.90	379.20	530.78	42.00	94.00
	60	0.00	0.39	0	0	8339.37	1326.02	1526.58	36.67	100.00
	70	mem	0.87	mem	524	mem	3654.87	9727.55	27.14	75.71
	80	mem	0.91	mem	622	mem	7117.06	15128.05	0.00	85.00
	90	mem	4.02	mem	175	mem	10881.89	time (2.77)	0.00	12.22
	100	mem	3.74	mem	54	mem	18334.13	time (3.17)	0.00	17.00
0.8	25	0.00	0.04	0	0	7.06	18.40	19.83	44.00	100.00
	30	0.00	0.02	0	0	17.87	25.64	28.32	50.00	100.00
	40	0.00	0.05	0	34	88.42	125.79	141.74	45.00	95.00
	50	0.00	0.11	0	0	305.75	252.98	280.06	52.00	98.00
	60	0.00	0.09	0	0	805.18	420.89	453.19	53.33	100.00
	70	mem	0.38	mem	198	mem	1259.86	1690.30	50.00	97.14
	80	mem	0.68	mem	220	mem	3931.77	5099.68	42.50	93.75
	90	mem	1.05	mem	8366	mem	5122.81	83735.91	0.00	93.33
	100	mem	1.04	mem	11174	mem	8995.96	time (0.61)	42.00	88.00

Table 5.2: Results of the exact algorithm for PO_2 with CAB instances

applies to PO_2 so, for a given instance, the actual size formulation (5.1)-(5.11) is considerably larger than that of the PO_1 formulation (5.1)-(5.10). Despite the fact that Property 5.1 no longer applies to the PO_2 formulation (5.1)-(5.11), D_2 could be optimally solved for all 27 instances using only 3 GB of memory, producing percentage deviations $\%Dev$ smaller than %1 for 20 of the instances, and smaller than 4.02% for the remaining 6 instances. Moreover, BB_2 was able to solve to optimality 21 benchmark instances within the time limit of 86,400 seconds. For the remaining six instances the percentage optimality gaps at termination (given in parentheses under the column *Time (sec)*) never exceed %3.17. The effectiveness of the partial enumeration and the reduction tests is higher in PO_2 than in PO_1. This effectiveness is particularly noticeable for the instances with higher values of α. Altogether, the partial enumeration was able to fix all the hubs in 7 instances, and the reduction tests fixed more than % 40 of the hubs in 11 additional instances.

We complete the information reported and discussed above, by analyzing in detail the performance of each of the steps of the enumeration trees of BB_1 and BB_2. In particular, Tables 5.3 and 5.4 show additional information of the partial enumeration at the root node, as well as of each of the branching levels, namely branching on hubs (z variables), branching on served nodes (s variables) and branching on hub edges (y variables). The first two columns in each table give the discount factor α, and the number of nodes $|N|$ of each instance. The next three columns under the heading of *Nodes* depict the exact number of nodes explored at each of the levels of the enumeration trees: enumeration on the hub variables (z), enumeration on the served nodes variables (s), and enumeration on the hub edges variables (y). The next five columns, under the heading *Time (sec)*, indicate the computing times, in seconds, consumed at each of the following steps: root node, partial enumeration, branching on z, branching on s, and branching on y. Similarly, the last four columns under the heading *%Dev* give the percent deviation of the best-known solution at the end of each step relative to the optimal (or best-known solution). These deviations have been computed as $100(v - v^*)/v^*$ where v is the upper bound at the end of each level, and v^* denotes the optimal or best-known value for each instance.

Table 5.3 further confirms the effectiveness for PO_1 of Property 5.1 and of the partial enumeration at the root node, which allow fixing hubs and also eliminating hub edges. Note that, particularly for smaller values of α, the enumeration trees of BB_1 generate very few nodes at the first level (z) and also at the level of the hub

α	$\|N\|$	*Nodes*			*Time (sec)*					*% Dev*			
		z	s	y	*Root*	*PE*	z	s	y	*Root*	*PE*	z	s
0.2	25	0	0	0	2	0	0	0	0	0.00	0.00	0.00	0.00
	30	0	0	0	6	0	0	0	0	0.00	0.00	0.00	0.00
	40	0	0	0	36	9	0	0	1	0.04	0.00	0.00	0.00
	50	0	0	0	82	36	2	0	0	0.11	0.03	0.00	0.00
	60	2	104	54	182	106	7	36	18	0.16	0.06	0.06	0.01
	70	2	146	70	392	322	13	86	38	0.27	0.05	0.05	0.01
	80	2	220	118	737	588	21	202	92	0.26	0.11	0.11	0.02
	90	4	758	556	1298	1049	53	1097	632	0.36	0.19	0.19	0.04
	100	44	3182	3512	1971	2184	418	7607	6870	0.64	0.21	0.20	0.04
0.5	25	0	0	0	0	0	0	0	0	0.00	0.00	0.00	0.00
	30	0	0	0	4	0	0	0	0	0.00	0.00	0.00	0.00
	40	0	0	0	14	4	1	0	0	0.03	0.01	0.01	0.00
	50	0	0	0	46	12	2	0	0	0.10	0.07	0.07	0.00
	60	2	130	30	110	32	5	33	9	0.13	0.04	0.04	0.01
	70	6	452	112	249	129	16	187	46	0.27	0.13	0.13	0.06
	80	28	404	168	447	272	74	288	89	0.40	0.23	0.20	0.02
	90	454	2144	1078	635	1189	3377	6646	2977	1.46	1.02	0.36	0.03
	100	572	2480	614	1162	1428	5028	8033	1886	1.28	0.81	0.22	0.06
0.8	25	0	0	0	2	0	0	0	0	0.02	0.00	0.00	0.00
	30	0	0	0	3	1	0	0	0	0.01	0.00	0.00	0.00
	40	0	0	0	10	3	0	0	1	0.03	0.00	0.00	0.00
	50	8	120	0	22	10	4	8	0	0.36	0.10	0.10	0.00
	60	4	128	0	47	20	6	28	0	0.27	0.11	0.11	0.00
	70	52	114	0	155	108	74	50	0	0.51	0.09	0.02	0.02
	80	70	722	0	267	219	134	465	0	0.53	0.25	0.17	0.00
	90	140	20278	3796	471	473	398	16065	3382	0.88	0.63	0.45	0.45
	100	210	46960	5202	698	677	862	52043	4267	0.87	0.66	0.56	0.56

Table 5.3: Detailed results of exact algorithm using CAB instances for PO_1

edges, where only for 12 out of the 27 instances any such node was generated. As can be seen, the most consuming level is the branching on served nodes (s), but a reduction in the percent deviation can be clearly observed after each step. In any case, the majority of the instances can be solved to optimality in less than one hour of computing time (21 out of 27), including the three larger instances with $N = 80$ nodes, which highlights the efficiency of BB_1.

The results of Table 5.4 allow making similar observations about the effectiveness of BB_2 for solving PO_2. Similarly to BB_1, there are fewer nodes at the hub nodes level (z) than at the other levels. However, for the largest instances there are still quite a few hubs to branch on after the partial enumeration. Despite the difficulty of PO_2, BB_2 is still robust for solving it: nine out of the 27 instances are optimally solved

α	$\lvert N \rvert$	*Nodes*			*Time (sec)*					*% Dev*			
		z	s	y	*Root*	PE	z	s	y	*Root*	PE	z	s
0.2	25	0	0	0	28	0	0	0	0	0.00	0.00	0.00	0.00
	30	4	26	0	61	13	3	4	0	0.07	0.06	0.02	0.02
	40	6	106	54	486	86	18	91	55	0.20	0.10	0.10	0.03
	50	2	88	60	553	234	77	171	119	0.19	0.07	0.07	0.03
	60	92	466	314	7016	1536	823	2321	1615	0.71	0.39	0.31	0.05
	70	366	1392	852	7550	4313	10165	21481	14100	0.97	0.44	0.30	0.07
	80	256	3	n.a.	15348	11870	59241	561	time	1.15	1.15	0.02	0.02
	90	335	n.a.	n.a.	27939	17165	41395	time	time	1.40	0.93	0.93	0.92
	100	490	83	n.a.	14093	26171	41851	4778	time	1.44	1.02	0.35	0.35
0.5	25	0	0	0	21	1	1	0	0	0.04	0.02	0.00	0.00
	30	0	0	0	42	3	0	0	0	0.03	0.00	0.00	0.00
	40	0	0	0	138	22	0	0	0	0.14	0.00	0.00	0.00
	50	10	90	0	379	66	23	62	0	0.29	0.12	0.12	0.12
	60	0	0	0	1326	187	14	0	0	0.39	0.13	0.00	0.00
	70	170	286	68	3655	937	2424	2003	709	0.87	0.32	0.19	0.03
	80	80	474	68	7117	1580	1481	4101	848	0.91	0.86	0.24	0.02
	90	175	n.a.	n.a.	10882	22216	53576	time	time	4.02	3.56	2.85	2.85
	100	54	n.a.	n.a.	18334	34389	33754	time	time	3.74	3.27	3.27	3.27
0.8	25	0	0	0	18	1	0	0	0	0.04	0.00	0.00	0.00
	30	0	0	0	26	2	1	0	0	0.02	0.01	0.00	0.00
	40	4	30	0	126	6	4	6	0	0.05	0.01	0.01	0.01
	50	0	0	0	253	21	0	0	0	0.11	0.00	0.00	0.00
	60	0	0	0	421	25	0	0	0	0.09	0.00	0.00	0.00
	70	4	194	0	1260	138	34	258	0	0.38	0.05	0.03	0.03
	80	30	168	22	3932	357	237	473	101	0.68	0.27	0.27	0.02
	90	38	6100	2228	5123	452	783	51097	26281	1.05	0.62	0.62	0.62
	100	98	9816	1260	8996	734	1566	64718	12828	1.04	0.61	0.61	0.61

Table 5.4: Detailed results of exact algorithm using CAB instances for PO_2

without any branching, including the 60 nodes instances for $\alpha = 0.5, 0.8$. Moreover, 15 instances are optimally solved in less than an hour of computing time. For only six instances the optimality of the best-known solution could not be proven within the time limit of one day.

Chapter 6

Hub Location and Pricing Problems

As we have mentioned before, one of the main applications of the HNDPPs that we have studied in the previous chapters, arise in airline transportation, where airline companies locate hub facilities (main airports) and set prices on the flights so as to maximize their profit while having the restriction of choosing a price that is lower than the competitor in order to capture the customers' demand. That is, the ticket prices have to be low enough to capture the demand of some of the attractive customers. Without the presence of other competitors, the company is free to set the price as high as possible since the customers have no other choice. However, in practice there exist competitors that have set their own prices and the customers always prefer the company that minimizes their cost.

In this chapter we focus on joint hub location and pricing problems applicable to the design of transportation and telecommunication networks. In this class of problems the goal is to determine the set of prices, location of hub facilities and the routes for serving the commodities. The company, having information on the price of the competitor and the possible reaction of the customers to its decision, wishes to find the optimal network and price that would lead to an optimal routing decision for the costumers. Once the price and network are fixed, the customers make an optimal choice between the company and the competitor.

Dealing with such problems gives rise to *hub location and pricing problems* (HLPPs) with two levels in the decision making process: the first level is a company (the leader) that decides about the price and location of hubs with the objective of maximizing its total profit (the difference between the revenue of routing commodities and the transportation and set up cots of opening hub facilities). The second level is a set of commodities (the followers) that are looking for the service provider and the path that minimizes their routing costs. Similar to other network pricing problems described in Chapter 2, there are two options on the price setting: pricing on paths and pricing on arcs. In the *hub location and pricing problem with pricing on paths* (HLPP-P), prices are set on each commodity using a path that includes at least one hub and at most two hubs. However, in the *hub location and pricing problem with pricing on arcs* (HLPP-A), there is a price associated with each arc of the network regardless of which commodities use the arc. The latter problem is more flexible in the sense that it allows paths with more than one hub arc. In this chapter, we show how both problems can be stated as MIP bilevel programs. We also present single-level reformulations for each of them.

The results of computational experiments clearly show the difficulty of HLPP-As, as a general purpose solver is not able to solve single-level reformulations even for a ten node instance due to memory issues. We use two variants of a math-heuristic to provide feasible solutions to the HLPP-A. Both heuristics construct feasible solutions in three steps. In the first step, we select the set of open hubs. Second, we set prices on each arc of the solution network. Having the network and prices, in the last step we solve the routing problem. The two heuristics differ in the way solutions are initially constructed. The first heuristic generates many random solution networks at each iteration using a given criteria. The second heuristic randomly opens hubs and changes the solution using destroy and repair strategies. However, they both set prices and identify feasible routes in a similar way.

This chapter is organized as follows: In Section 6.1 we present the formal definition and assumptions of the considered HLPPs. Sections 6.2 and 6.3 present the bilevel program together with the single-level MIP formulation of the HLPP-P and the HLPP-A, respectively. Section 6.4 describes the two math-heuristics used to solve the HLPP-A. The chapter ends in Section 6.5 by presenting some computational experiments.

6.1 Formal Definition of the Hub Location and Pricing Problem

We can formally define the HLPP as follows. Let $G = (N, A)$ be a complete directed graph, where N=$\{1, 2, ..., n\}$ represents the set of nodes and A represents the set of arcs. For each $a = (i, j) \in A$, let C_a denote the distance or unit transportation cost between nodes i and j, which we assume to be symmetric. Let $H \subseteq N$ be the set of potential hub locations. The parameter α $(0 \leq \alpha \leq 1)$ is used as a discount factor to provide reduced unit transportation costs on hub edges to represent economies of scale.

Let K denote the set of commodities. Commodity $k \in K$ is defined as a triplet $(o(k), d(k), W_k)$, where $o(k), d(k) \in N$, respectively denote its origin and its destination, also referred to as its O/D pair, and W^k denotes its service demand, i.e., the amount of flow that must be routed from $o(k)$ to $d(k)$ if commodity k is served. We define the per unit transportation cost for routing commodity k on the path

$(o(k), i, j, d(k))$ as $F_a^k = (\chi C_{(o(k)i)} + \alpha C_{(ij)} + \delta C_{(jd(k))})$, where the parameters χ and δ reflect weights factors for collection and distribution, respectively. For each $i \in H$, f_i is the fixed setup cost for opening a hub at node i. Let q^k denote the competitor's price for commodity k. We also define $\delta(i) \subseteq A$ for each $i \in H$ as the subset of arcs incident to node i and accordingly, $\delta^-(j)$ as the subset of arcs entering node i and $\delta^+(j)$ as the subset of arcs leaving node i.

Similarly to most HLPs, we consider the following assumptions:

- All O/D paths include at least one hub node. That is, the solution network contains no direct connections between two non-hub nodes.
- Nodes can be assigned to more than one hub node, i.e. multiple assignments.
- There is no set-up cost on the arcs and thus, hubs are fully interconnected.
- Distances satisfy the triangle inequality.
- In case of equal prices, the lower level (the set of commodities) always prefers the leader to its competitor.

6.2 HLPs with Pricing on Paths

In the HLP with Pricing on Paths, profit is defined for each commodity based on each specific path. For simplicity, we require solution networks to contain at most three edges in each O/D path (at most one hub arc). This hypothesis that is common in classical hub location models may seem restrictive as compared to other network design models. Note, however, that this hypothesis is consistent with the potential applications that we mention, mainly air transportation where paths with three legs already correspond to two intermediate transfers.

6.2.1 A Bilevel Programming Formulation

We define the following set of decision variables. For each $a \in A$ and $k \in K$, we define the continuous variable P_a^k as the price of commodity k routed through hub arc a. For $i \in H$, z_i is the binary location variable equal to 1 if and only if a hub is located at node i. For $k \in K$, $a \in A$, we define routing variables x_a^k equal to 1 if and only if commodity k is routed via arc $a \in A$. Finally, for $k \in K$, we define the binary variable y^k equal to 1 if commodity k is routed by the competitor, 0 otherwise. Using these sets of variables, the HLPP-P can be formulated as follows:

$$(POP) \qquad \max_{P,z,x,y} \sum_{k \in K} W^k \sum_{a \in A} (P_a^k - F_a^k) x_a^k - \sum_{i \in H} f_i z_i \tag{6.1}$$

$$z_i \in \{0,1\} \qquad i \in H \tag{6.2}$$

$$P_a^k \geq 0 \qquad a \in A, k \in K \tag{6.3}$$

where (x,y) solves

$$\min_{x,y} \sum_{k \in K} (\sum_{a \in A} P_a^k x_a^k + q^k y^k) \tag{6.4}$$

$$\text{s.t.} \sum_{a \in A} x_a^k + y^k = 1 \qquad k \in K \tag{6.5}$$

$$\sum_{a \in \delta(i)} x_a^k \leq z_i \qquad k \in K, i \in H \tag{6.6}$$

$$x_a^k,\ y^k \geq 0 \qquad a \in A, k \in K. \tag{6.7}$$

The upper-level objective (6.1) is to maximize the difference between the revenue arising from routing commodities and the transportation costs and set-up costs. The objective of the lower-level problem (6.4) is to minimize the transportation cost by choosing the path and the company with the minimum cost. Constraints (6.5) force commodities to be routed either by the leader or by the competitor. Constraints (6.6) prohibit commodities to be routed through hubs that are not open.

6.2.2 Moving Constraints to the Upper Level

In general, we cannot freely move constraints from one level to the other if the constraint includes variables that are used in both levels. However, Brotcorne et al. [19] proved that when some conditions are satisfied, we can move a specific set of constraints to the upper level.

Proposition 6.1 (Brotcorne et al. [19]) *Assume that $P1$ admits an optimal solution and that the matrix G has nonnegative entries. Then the sets of optimal solutions of the mathematical programs $P1$ and $P2$, displayed below, are nonempty and coincide.*

$$
(P1) \quad \begin{aligned} \max_{T,x,y} \quad & Tx - cx \\ & \textit{where } (x,y) \textit{ solves} \\ \min_{x,y} \quad & Tx + dy \\ s.t. \quad & Ex + Fy = b^1 \\ & Gx \le b^2 \\ & x,\ y \ge 0 \end{aligned}
\qquad\qquad
(P2) \quad \begin{aligned} \max_{T,x,y} \quad & Tx - cx \\ & Gx \le b^2 \\ & \textit{where } (x,y) \textit{ solves} \\ \min_{x,y} \quad & Tx + dy \\ s.t. \quad & Ex + Fy = b^1 \\ & x,\ y \ge 0. \end{aligned}
$$

Proposition 6.2 *Constraints* (6.6) *of POP can be moved to the upper level.*

Proof . To prove the result, we will show that for a fixed location vector z, the resulting pricing and routing problem can be cast in the format $P1$. To aim this, we introduce x_i as the total number of commodities passing through hub node i

$x_i = \sum_{k \in K} x_a^k \quad \forall a \in \delta(i),$

and replace the individual commodity constraints $\sum_{a \in \delta(i)} x_a^k \leq z_i$ by the equivalent global constraint

$x_i \leq |K| z_i.$

The resulting bilevel program is

$$
\begin{aligned}
&\max_{P,x,y} \quad \sum_{k \in K} W^k (P^k - F^k) x^k \\
&\text{ where (x,y) solves} \\
&\min_{x,y} \quad \sum_{k \in K} (P^k x^k + q^k y^k) \\
&s.t. \quad x^k + y^k = 1 \qquad\qquad k \in K \\
&\qquad\quad x \leq |K| z \\
&\qquad\quad x = \sum_{k \in K} x^k \\
&\qquad\quad y = \sum_{k \in K} y^k \\
&\qquad\quad x^k,\ y^k \geq 0. \qquad\qquad k \in K
\end{aligned}
$$

Now by making the correspondences

$x \equiv x$

$y \equiv (y, (x^k)_{k \in K}, (y^k)_{k \in K})$

$G \equiv I$

$b^2 \equiv |K| z$

$$Ex + Fy = b^1 \equiv \begin{pmatrix} I & 0 & -I & \dots & -I & 0 & \dots & 0 \\ 0 & I & 0 & \dots & 0 & -I & \dots & -I \\ 0 & 0 & I & \dots & 0 & I & \dots & 0 \\ 0 & 0 & 0 & \ddots & 0 & 0 & \ddots & 0 \\ 0 & 0 & 0 & \dots & I & 0 & \dots & I \end{pmatrix} \begin{pmatrix} x \\ y \\ (x^k)_{k\in K} \\ (y^k)_{k\in K} \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$$

once recovers the generic form of $P1$ and the results follows.

■

6.2.3 A Single-level MIP Reformulation

We next present a single-level MIP reformulation based on the procedure described in Section 6.2.2. We recall that Labbé et al. [79] show that the single-level reformulation of a bilevel programming problem consists of : i) the objective function and constraints of the first level, ii) the primal and dual constraints of the second level and, iii) the optimality conditions of the second level. After moving constraints (6.6) to the first level, in the second level we have:

$$\text{minimize} \quad \sum_{k\in K}(\sum_{a\in A} P_a^k x_a^k + q^k y^k) \tag{6.8}$$

$$\text{subject to} \quad \sum_{a\in A} x_a^k + y^k = 1 \qquad k \in K \tag{6.9}$$

$$x_a^k,\ y^k \geq 0 \qquad a \in A, k \in K, \tag{6.10}$$

and the dual of the second level is:

$$\begin{aligned}
\text{maximize} \quad & \sum_{k\in K} \lambda^k \\
\text{subject to} \quad & \lambda^k \le P_a^k && a \in A, k \in K \quad (6.11) \\
& \lambda^k \le q^k && k \in K, \quad (6.12)
\end{aligned}$$

where λ^k is the dual variable associated with constraints (6.9). The KKT conditions associated with the primal and the dual pair of the second level are:

$$\sum_{a\in A} x_a^k + y^k = 1 \qquad k \in K \qquad (6.13)$$

$$x_a^k,\ y^k \ge 0 \qquad a \in A, k \in K \qquad (6.14)$$

$$\lambda^k \le P_a^k \qquad a \in A, k \in K \qquad (6.15)$$

$$\lambda^k \le q^k \qquad k \in K \qquad (6.16)$$

$$\sum_{k\in K} \lambda^k = \sum_{k\in K}\sum_{a\in A} P_a^k x_a^k + q^k y^k. \qquad (6.17)$$

Now, given that the second level problem can be decomposed by commodity, we have:

$$\lambda^k = \sum_{a\in A} P_a^k x_a^k + q^k y^k \qquad k \in K. \qquad (6.18)$$

From (6.18) together with (6.15) and (6.16), we obtain:

$$\sum_{a\in A} P_a^k x_a^k + q^k y^k \le P_b^k \qquad b \in A, k \in K \qquad (6.19)$$

$$\sum_{a\in A} P_a^k x_a^k + q^k y^k \le q^k \qquad k \in K. \qquad (6.20)$$

Therefore, the single-level MIP reformulation of the HLPP-P can be shown as follows:

$$\text{maximize} \quad \sum_{k\in K} W^k \sum_{a\in A}(P_a^k - F_a^k)x_a^k - \sum_{i\in H} f_i z_i \tag{6.21}$$

$$\text{subject to} \quad \sum_{a\in A} x_a^k + y^k = 1 \qquad k \in K \tag{6.22}$$

$$\sum_{a\in\delta(i)} x_a^k \leq z_i \qquad k \in K, i \in H \tag{6.23}$$

$$\sum_{a\in A} P_a^k x_a^k + q^k y^k \leq P_b^k \qquad k \in K, b \in A \tag{6.24}$$

$$\sum_{a\in A} P_a^k x_a^k + q^k y^k \leq q^k \qquad k \in K \tag{6.25}$$

$$\sum_{k\in K} \lambda^k = \sum_{k\in K}\sum_{a\in A} P_a^k x_a^k + q^k y^k \tag{6.26}$$

$$x_a^k,\ y^k \geq 0 \qquad a \in A, k \in K \tag{6.27}$$

$$z_i \in \{0,1\} \qquad i \in H \tag{6.28}$$

$$P_a^k \geq 0 \qquad a \in A, k \in K, \tag{6.29}$$

We note that the first term of the objective (6.21) and constraints (6.24) and (6.25) are nonlinear due to the multiplication of the $P_a^k x_a^k$ variables. To avoid the nonlinear term in constraints (6.25) and (6.24), we can replace them by the following constraints where M_1 is a big number:

$$P_a^k \leq P_b^k + M_1(1 - x_a^k) \qquad k \in K, a \in A, b \in A \tag{6.30}$$

$$P_a^k \leq q^k + M_1(1 - x_a^k) \qquad k \in K, a \in A, \tag{6.31}$$

To linearize the objective function we define a new set of variables. For each $a \in A$ and $k \in K$, let $R_a^k = x_a^k P_a^k$, which shows the effective price paid by commodity k on arc a and is equal to P_a^k if the commodity uses an arc a, and 0 otherwise. We also need to add the following additional constraints to make the linearizion valid:

$$R_a^k \leq P_a^k \qquad k \in K, a \in A \qquad (6.32)$$

$$R_a^k \leq M_2 x_a^k \qquad k \in K, a \in A, \qquad (6.33)$$

Where M_2 is the upper bound on the price, which can be set to the price of the competitor (q^k) since the leader cannot set a price larger than the competitor if he/she wants the customer to choose him/her. The linear MIP formulation is thus as follows:

$$(POP_L) \text{ maximize} \quad \sum_{k \in K} W^k \sum_{a \in A} (R_a^k - F_a^k x_a^k) - \sum_{i \in H} f_i z_i$$

$$\text{subject to} \quad \sum_{a \in A} x_a^k + y^k = 1 \qquad k \in K \qquad (6.34)$$

$$\sum_{a \in \delta(i)} x_a^k \leq z_i \qquad k \in K, i \in H \qquad (6.35)$$

$$\sum_{a \in A} R_a^k + q^k y^k \leq P_b^k \qquad k \in K, b \in A \qquad (6.36)$$

$$\sum_{a \in A} R_a^k + q^k y^k \leq q^k \qquad k \in K \qquad (6.37)$$

$$\sum_{k \in K} \lambda^k = \sum_{k \in K} \sum_{a \in A} R_a^k + q^k y^k \qquad (6.38)$$

$$R_a^k \leq P_a^k \qquad k \in K, a \in A \qquad (6.39)$$

$$R_a^k \leq q^k x_a^k \qquad k \in K, a \in A \qquad (6.40)$$

$$x_a^k, \ y^k \geq 0 \qquad a \in A, k \in K \qquad (6.41)$$

$$z_i \in \{0, 1\} \qquad i \in H \qquad (6.42)$$

$$P_a^k, R_a^k \geq 0 \qquad a \in A, k \in K. \qquad (6.43)$$

Preliminary computational experiments showed that, due to the huge number of variables and constraints required in POP_L, only small size instances with up to 15 nodes can be solved with a general purpose solver. However, the following results show that this problem can be actually transformed into the UHLPP introduced in Chapter 3.

Proposition 6.3 *The optimal price P_b^k to route each commodity $k \in K$ through any arc $b \in A$ corresponds to the price of the competitor q^k for routing the same commodity.*

Proof . If commodity $k \in K$ is routed, due to (6.36) there is at most one hub arc $b \in A$ on the routes of the commodity with the price of $P_b^k \geq R_a^k$. Indeed, to maximize the profit, the leader wants to choose the maximum possible price for the arc b. On the other hand, he/she has the information on the reaction of the follower, who will opt for the minimum price. If the leader sets P_b^k to any value greater than q^k, the follower reacts by choosing the competitor. On the other hand, if P_b^k is set to any value less than q^k, the follower will choose the leader but the leader is losing $(P_b^k - q^k)$ from the per unit profit. So, since it is the simultaneous optimization of both levels, the optimal value for P_b^k is q^k. ■

Proposition 6.4 *The HLPP-P is equivalent to the $UHLPP$.*

Proof . We can use Proposition 6.4 to a priori define the pricing decisions in all the paths to $P_a^k = q^k$ for each $k \in K$ and $a \in A$. Setting $c_i = 0$ for each $i \in N$ the HLPP-P reduces then to a single level problem where only the locational decisions and routing decisions need to be optimized. This corresponds to the solution of the UHLPP and the result follows. ■

6.3 HLPs with Pricing on Arcs

We now consider HLPs with pricing on arcs, where there is a price associated with every arc of the network. In this case, in order to obtain the total price for the routing of a given commodity, one needs to add the price of all the arcs on the path of that commodity. Contrary to HLPs with pricing on paths, when pricing on arcs we allow paths to be more flexible, and as a result, they may contain more than one hub arc. The main challenge in these problems is that since the pricing strategy is different for the leader and the competitor, i.e., the competitor's price focuses on commodities while the leader's price focuses on arcs, pricing is far from trivial. Moreover, as each arc may be used in the path to route several commodities, setting the prices on the arcs becomes more involved. However, similar to path pricing, the pricing restriction still exists, i.e. the total price of each commodity (sum of the prices of all the arcs that are on the path of the commodity) needs to be less than the competitor's price.

6.3.1 A Bilevel Programming Formulation

We define P_a as the price for using arc $a \in A$. In order to keep track of the arcs used for routing each commodity, we need to define the following set of decision variables. U_i^k is equal to 1, if arc $(o(k), i) \in A$ is the first arc in the path of commodity $k \in K$, where $o(k)$ is the origin of k and $i \in H$ is the first hub in the path. V_j^k is equal to 1, if arc $(j, d(k))$ is the last arc used to route commodity $k \in K$, where $d(k)$ is the destination of k and $j \in H$ is the last hub on the path. x_a^k is equal to 1 if hub arc $a \in A$ is used in the path of commodity k, 0 otherwise. Using these decision variables, the HLPP-A can be stated as:

$$
\begin{aligned}
(POA) \max_{P,U,V,x,y} \quad & \sum_{a \in A} \sum_{k \in K: a_1 = o(k)} W^k (P_a - C_a) U^k_{a_2} + \sum_{a \in A} \sum_{k \in K: a_2 = d(k)} W^k (P_a - C_a) V^k_{a_1} \\
& + \sum_{a \in A} \sum_{k \in K} W^k (P_a - \alpha C_a) x^k_a - \sum_{i \in H} f_i z_i \\
& P_a \geq 0 \qquad a \in A \qquad (6.44) \\
& z_i \in \{0,1\} \qquad i \in H \qquad (6.45)
\end{aligned}
$$

where (U,V,x,y) solves

$$
\begin{aligned}
\min_{U,V,x,y} \quad & \sum_{a \in A: o(k) = a_1} \sum_{k \in K} P_a U^k_{a_2} + \sum_{a \in A: d(k) = a_2} \sum_{k \in K} P_a V^k_{a_1} \\
& + \sum_{a \in A} \sum_{k \in K} P_a x^k_a + \sum_{k \in K} q^k y^k \qquad (6.46) \\
\text{s.t.} \quad & \sum_{a \in \delta^-(i)} x^k_a + U^k_i \leq z_i \qquad i \in H, k \in K \qquad (6.47)
\end{aligned}
$$

$$
\begin{aligned}
& \sum_{a \in \delta^+(j)} x^k_a + V^k_j \leq z_j \qquad j \in H, k \in K \qquad (6.48) \\
& \sum_{i \in H} U^k_i + y^k = 1 \qquad k \in K \qquad (6.49) \\
& \sum_{j \in H} V^k_j + y^k = 1 \qquad k \in K \qquad (6.50) \\
& U^k_i + \sum_{a \in A: a_2 = i} x^k_a = V^k_i + \sum_{a \in A: a_1 = i} x^k_a \qquad k \in K, i \in H \qquad (6.51) \\
& x^k_a \in \{0,1\} \qquad a \in A, k \in K \qquad (6.52) \\
& U^k_i, V^k_i \in \{0,1\} \qquad i \in H, k \in K \qquad (6.53) \\
& y^k \in \{0,1\} \qquad k \in K. \qquad (6.54)
\end{aligned}
$$

The objective of the leader is to maximize the difference between the revenue and cost of routing each commodity. The first term shows the profit of routing through the first leg and the second term is the profit of routing through the last leg while the third term is the profit of routing through hub arcs. Similarly, the objective function of the second level has also three parts for the transportation cost if the customer chooses the leader and the last part is in case of choosing the competitor. Constraints (6.47) indicate that if hub $i \in H$ is used to route commodity k, either as the first hub or in between, that hub should be open. Equivalently, constraints (6.48), show that to use hub $j \in H$ on the path of $k \in K$, as the last hub or in between, j should be open as a hub. Constraints (6.49) and (6.50) impose that each commodity can be routed either by the leader or the competitor. Finally, constraints (6.51) are the flow conservation constraints to ensure a feasible path is built between the origin and destination of a commodity, if routed.

6.3.2 Moving Constraints to the Upper Level

Similar to the previous model, it is possible to move some of the constraints of the lower level of POA to the upper level.

Proposition 6.5 *Constraints* (6.47) *and* (6.48) *of POA can be moved to the upper level.*

Proof . To prove the result, we will show that for a fixed location vector z, the resulting pricing and routing problem can be cast in the format $P1$. To aim this, we introduce x_i as the total number of commodities passing through hub node i

$$x_i = \sum_{k \in K} (x_a^k + U_i^k) \quad \forall a \in \delta^-(i),$$

or equivalently

$$x_i = \sum_{k \in K} (x_a^k + V_i^k) \quad \forall a \in \delta^+(i),$$

and replace the individual commodity constraints (6.47) and (6.48) by the equivalent global constraint

$$x_i \leq |K| z_i.$$

The resulting bilevel program is :

$$\max_{P,U,V,x,y} \quad \sum_{k\in K} W^k\{(P-C)U^k+(P-C)V^k+(P-\alpha C)x^k\}$$

where (U,V,x,y) solves

$$\begin{aligned}
\min_{U,V,x,y} \quad & \sum_{k\in K}\{P(U^k+V^k+x^k)+q^k y^k\} \\
\text{s.t.} \quad & x \le |K|z \\
& U^k + y^k = 1 && k\in K \\
& V^k + y^k = 1 && k\in K \\
& x^k + U^k = x^k + V^k && k\in K \\
& \\
& x = \sum_{k\in K}(x^k+U^k) \\
& x = \sum_{k\in K}(x^k+V^k) \\
& y = \sum_{k\in K} y^k \\
& U^k, V^k, x^k,\ y^k \ge 0 && k\in K.
\end{aligned}$$

Now, by making the correspondences

$$x \equiv x$$

$$y \equiv (y,(U^k)_{k\in K},(V^k)_{k\in K},(x^k)_{k\in K},(y^k)_{k\in K})$$

$$G \equiv I$$

$$b^2 \equiv |K|z$$

$$
Ex + Fy = b^1 \equiv \begin{pmatrix}
I & 0 & -I & \dots & -I & 0 & \dots & 0 & -I & \dots & -I & 0 & \dots & 0 \\
0 & I & 0 & \dots & 0 & -I & \dots & -I & 0 & \dots & 0 & 0 & \dots & 0 \\
0 & 0 & 0 & \dots & 0 & I & \dots & 0 & I & \dots & 0 & 0 & \dots & 0 \\
0 & 0 & 0 & \dots & 0 & 0 & \ddots & 0 & 0 & \ddots & 0 & 0 & \dots & 0 \\
0 & 0 & 0 & \dots & 0 & 0 & \dots & I & 0 & \dots & I & 0 & \dots & 0 \\
0 & 0 & 0 & \dots & 0 & I & \dots & 0 & 0 & \dots & 0 & I & \dots & 0 \\
0 & 0 & 0 & \dots & 0 & 0 & \ddots & 0 & 0 & \dots & 0 & 0 & \ddots & 0 \\
0 & 0 & 0 & \dots & 0 & 0 & \dots & I & 0 & \dots & 0 & 0 & \dots & I \\
0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & I & \dots & 0 & -I & \dots & 0 \\
0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \ddots & 0 & 0 & \ddots & 0 \\
0 & 0 & 0 & \dots & 0 & 0 & \dots & 0 & 0 & \dots & I & 0 & \dots & -I
\end{pmatrix}
\begin{pmatrix} x \\ y \\ (x^k)_{k\in K} \\ (y^k)_{k\in K} \\ (U^k)_{k\in K} \\ (V^k)_{k\in K} \end{pmatrix}
= \begin{pmatrix} 0 \\ 0 \\ 1 \\ \vdots \\ 1 \\ 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}
$$

one recovers the generic form of $P1$, and the result follows. ■

6.3.3 The Single-level Reformulation

We next present a single-level MIP reformulation based on the procedure described in Section (2.4.2). After moving the constraints that include first-level variables to the upper level, the next step is finding the dual of the second level. As mentioned, the reformulation consists of the constraints from the first level plus the KKT conditions of the second level. By moving constraints (6.47) and (6.48) to the first level, at the second level we have:

$$
\begin{aligned}
\text{minimize} \quad & \sum_{a\in A:o(k)=a_1}\sum_{k\in K} P_a U^k_{a_2} + \sum_{a\in A:d(k)=a_2}\sum_{k\in K} P_a V^k_{a_1} \\
& + \sum_{a\in A}\sum_{k\in K} P_a x^k_a + \sum_{k\in K} q^k y^k \\
\text{subject to} \quad & \sum_{i\in H} U^k_i + y^k = 1 && k \in K \quad (6.55) \\
& \sum_{j\in H} V^k_j + y^k = 1 && k \in K \quad (6.56) \\
& U^k_i + \sum_{a\in A:a_2=i} x^k_a = V^k_i + \sum_{a\in A:a_1=i} x^k_a && k \in K, i \in H. \quad (6.57)
\end{aligned}
$$

Let $(\lambda^k, \gamma^k, \beta_i^k)$ be the dual variables associated with constraints (6.55), (6.56) and (6.57), respectively. The dual of the second level is :

$$\text{minimize} \quad \sum_{k \in K} (\lambda^k + \gamma^k)$$

$$\text{subject to} \quad \lambda^k + \beta_{a_2}^k \leq P_a \qquad k \in K, a \in A : o(k) = a_1 \tag{6.58}$$

$$\gamma^k - \beta_{a_1}^k \leq P_a \qquad k \in K, a \in A : d(k) = a_2 \tag{6.59}$$

$$\lambda^k + \gamma^k \leq q^k \qquad k \in K \tag{6.60}$$

$$\beta_{a_2}^k - \beta_{a_1}^k \leq P_a \qquad k \in K, a \in A, \tag{6.61}$$

and the KKT conditions associated with the primal and the dual pair of the second level are:

$$\sum_{i \in H} U_i^k + y^k = 1 \qquad k \in K \tag{6.62}$$

$$\sum_{j \in H} V_j^k + y^k = 1 \qquad k \in K \tag{6.63}$$

$$U_i^k + \sum_{a \in A: a_2 = i} x_a^k = V_i^k + \sum_{a \in A: a_1 = i} x_a^k \qquad k \in K, i \in H \tag{6.64}$$

$$\lambda^k + \beta_{a_2}^k \leq P_a \qquad k \in K, a \in A : o(k) = a_1 \tag{6.65}$$

$$\gamma^k - \beta_{a_1}^k \leq P_a \qquad k \in K, a \in A : d(k) = a_2 \tag{6.66}$$

$$\lambda^k + \gamma^k \leq q^k \qquad k \in K \tag{6.67}$$

$$\beta_{a_2}^k - \beta_{a_1}^k \leq P_a \qquad k \in K, a \in A \tag{6.68}$$

$$\sum_{k \in K} (\lambda^k + \gamma^k) = \sum_{k \in K} \Big(\sum_{a \in A: o(k) = a_1} P_a U_{a_2}^k + \sum_{a \in A: d(k) = a_2} P_a V_{a_1}^k + \sum_{a \in A} P_a x_a^k + q^k y^k \Big). \tag{6.69}$$

Now, given that the second level can be decomposed by commodity, we have:

$$\lambda^k + \gamma^k = \sum_{a \in A: o(k)=a_1} P_a U^k_{a_2} + \sum_{a \in A: d(k)=a_2} P_a V^k_{a_1} + \sum_{a \in A} P_a x^k_a + q^k y^k. \tag{6.70}$$

From (6.70) together with (6.67) we obtain:

$$\sum_{a \in A: o(k)=a_1} P_a U^k_{a_2} + \sum_{a \in A: d(k)=a_2} P_a V^k_{a_1} + \sum_{a \in A} P_a x^k_a + q^k y^k \leq q^k \qquad \forall k \in K \tag{6.71}$$

Therefore, the single-level MIP reformulation of the HLPP-A is as follows:

$$\begin{aligned}
\text{maximize} \quad & \sum_{a \in A} \sum_{k \in K: a_1 = o(k)} W^k (P_a - C_a) U^k_{a_2} + \sum_{a \in A} \sum_{k \in K: a_2 = d(k)} W^k (P_a - C_a) V^k_{a_1} \\
& + \sum_{a \in A} \sum_{k \in K} W^k (P_a - \alpha C_a) x^k_a - \sum_{i \in H} f_i z_i \\
\text{subject to} \quad & \sum_{a \in \delta^-(i)} x^k_a + U^k_i \leq z_i & i \in H, k \in K \quad (6.72) \\
& \sum_{a \in \delta^+(j)} x^k_a + V^k_j \leq z_j & j \in H, k \in K \quad (6.73) \\
& \sum_{i \in H} U^k_i + y^k = 1 & k \in K \quad (6.74) \\
& \sum_{j \in H} V^k_j + y^k = 1 & k \in K \quad (6.75) \\
& U^k_i + \sum_{a \in A: a_2 = i} x^k_a = V^k_i + \sum_{a \in A: a_1 = i} x^k_a & k \in K, i \in H \quad (6.76) \\
& \lambda^k + \beta^k_{a_2} \leq P_a & k \in K, a \in A : o(k) = a_1 \quad (6.77) \\
& \gamma^k - \beta^k_{a_1} \leq P_a & k \in K, a \in A : d(k) = a_2 \quad (6.78) \\
& \beta^k_{a_2} - \beta^k_{a_1} \leq P_a & k \in K, a \in A \quad (6.79)
\end{aligned}$$

$$\sum_{a\in A:o(k)=a_1} P_a U^k_{a_2} + \sum_{a\in A:d(k)=a_2} P_a V^k_{a_1}$$

$$+\sum_{a\in A} P_a x^k_a + q^k y^k \leq q^k \qquad \forall k \in K \qquad (6.80)$$

$$\sum_{k\in K}(\lambda^k + \gamma^k) = \sum_{k\in K}(\sum_{a\in A:o(k)=a_1} P_a U^k_{a_2} + \sum_{a\in A:d(k)=a_2} P_a V^k_{a_1}$$

$$+\sum_{a\in A} P_a x^k_a + q^k y^k) \qquad (6.81)$$

$$x^k_a \in \{0,1\} \qquad a \in A, k \in K \qquad (6.82)$$

$$U^k_i, V^k_i \in \{0,1\} \qquad i \in H, k \in K \qquad (6.83)$$

$$y^k \in \{0,1\} \qquad k \in K. \qquad (6.84)$$

This formulation has nonlinear terms both in the constraints and objective function. To linearize them we need to define three sets of new variables ($|N| \times |K| \times 2 + |A| \times |K|$ in total), as the effective price of the first leg, the last leg and the hub arcs, respectively:

- $PC^k_a = P_a U^k_{a_2} \qquad k \in K, a \in A : o(k) = a_1,$
- $PD^k_a = P_a V^k_{a_1} \qquad k \in K, a \in A : d(k) = a_2,$
- $PH^k_a = P_a x^k_a \qquad k \in K, a \in A,$

Using these three new variables and their linking constraints, the linearized MIP single-level reformulation is :

$$
\begin{aligned}
(POA_L)\ \text{maximize} \quad & \sum_{a\in A}\sum_{k\in K:a_1=o(k)} W^k PC_a^k - C_a U_{a_2}^k + \sum_{a\in A}\sum_{k\in K:a_2=d(k)} W^k PD_a^k \\
& -C_a V_{a_1}^k + \sum_{a\in A}\sum_{k\in K} W^k PH_a^k - \alpha C_a x_a^k - \sum_{i\in H} f_i z_i \\
\text{subject to} \quad & (6.72) - (6.79)
\end{aligned}
$$

$$\sum_{a\in A:a_1=o(k)} PC_a^k + \sum_{a\in A:a_2=d(k)} PD_a^k + \sum_{a\in A} PH_a^k + q^k y^k \le q^k \qquad k\in K \tag{6.85}$$

$$\sum_{k\in K}(\lambda^k+\gamma^k) = \sum_{k\in K}\Big(\sum_{a\in A:o(k)=a_1} PC_a^k + \sum_{a\in A:d(k)=a_2} PD_a^k + \sum_{a\in A} PH_a^k + q^k y^k\Big) \tag{6.86}$$

$$PH_a^k \le M_a^k X_a^k \qquad k\in K, a\in A \tag{6.87}$$

$$P_a - PH_a^k \le N_a(1-X_a^k) \qquad k\in K, a\in A \tag{6.88}$$

$$PH_a^k \le P_a \qquad k\in K, a\in A \tag{6.89}$$

$$PC_a^k \le M_a^k U_{a_2}^k \qquad k\in K, a\in A: o(k)=a_1 \tag{6.90}$$

$$P_a - PC_a^k \le N_a(1-U_{a_2}^k) \qquad k\in K, a\in A: o(k)=a_1 \tag{6.91}$$

$$PC_a^k \le P_a \qquad k\in K, a\in A: o(k)=a_1 \tag{6.92}$$

$$PD_a^k \le M_a^k V_{a_1}^k \qquad k\in K, a\in A: d(k)=a_2 \tag{6.93}$$

$$P_a - PD_a^k \le N_a(1-V_{a_1}^k) \qquad k\in K, a\in A: d(k)=a_2 \tag{6.94}$$

$$PD_a^k \le P_a \qquad k\in K, a\in A: d(k)=a_2 \tag{6.95}$$

$$z_i \in \{0,1\} \qquad i\in H \tag{6.96}$$

$$x_a^k \in \{0,1\} \qquad a\in A, k\in K \tag{6.97}$$

$$U_i^k, V_i^k \in \{0,1\} \qquad i\in H, k\in K \tag{6.98}$$

$$PH_a^k, PC_a^k, PD_a^k \ge 0 \qquad a\in A, k\in K \tag{6.99}$$

$$P_a \ge 0 \qquad a\in A \tag{6.100}$$

$$y^k \in \{0,1\} \qquad k\in K. \tag{6.101}$$

Constraints (6.87) to (6.95) are the linking constraints associated with the new variables ($3 \times |A| \times |K|, 3 \times |N| \times |K| \times 2$ in total). Constraints (6.87) impose that $PH_a^k = 0$, if hub arc a is not used for commodity $k \in K$. Constraints (6.88) and (6.89) set $PH_a^k = P_a$, if hub arc $a \in A$ is used for commodity $k \in K$. Constraints (6.90) to (6.92) impose the same condition for the collection leg while Constraints (6.93) to (6.95) are for the distribution leg.

Note that routing variables have to be binary for this linearizion to be valid. Also, the choice of M_a^k and N_a must be as tight as possible to ensure a valid formulation. Indeed, the smallest value for them is the price of competitor q^k. Constraints (6.85) enforce the total price of the arcs of the path of each commodity k to be less than q^k. Using these constraints, an upper bound on the price of each arc can be q^k which refers to a single arc path. As a result, if an arc is used for each commodity k, its price cannot exceed q^k.

6.4 A Heuristic Algorithm for HLPs with Pricing on Arcs

In this section we describe two versions of a math-heuristic we have developed to obtain feasible solutions and thus, lower bounds on the optimal value of the HLPP-A. The idea of the proposed heuristic is to decompose the decision-making process into three steps and to obtain feasible solutions of each subproblem based on the information obtained from the previous step. The proposed heuristics have two phases: a *constructive phase* and *local search phase*. The constructive phase concentrates on obtaining feasible solutions and has three steps: *i)* selection of hub facilities, *ii)* determination of prices on the arcs, and *iii)* selection and routing of commodities. The local search phase improves the hub network obtained by the constructive phase by exploring four neighborhoods: *i)* open a hub node, *ii)* close a hub node, *iii)* open/close hubs, and *iv)* close two hubs. The two versions of the heuristic differ only in the first phase. Each of these use a different way to select the hubs. The first one, denoted as MH_1, selects at each iteration different sets of hubs based on a given criteria while in the second one (MH_2), at each iteration a set nodes is randomly selected to be open as hubs and then the set is modified by using a destroy/repair strategy.

In what follows, let $(\tilde{z}, \tilde{P}, \tilde{y}, \tilde{U}, \tilde{V}, \tilde{x})$ denote a feasible solution to POA and $\tilde{h} = \{i : \tilde{z}_i = 1 : i \in H\}$ represent the set of open hubs at the current iteration. We initialize $(\tilde{z}, \tilde{P}, \tilde{y}, \tilde{U}, \tilde{V}, \tilde{x}) \leftarrow (0, 0, 0, 0, 0, 0)$ and $\tilde{h} \leftarrow \emptyset$ at iteration 0. We define $\underline{\eta}$ as the current lower bound. Let $\bar{N}$ be the sorted set of all the nodes (N) in a decreasing order based on the *total flow originating at each node* ($O_i = \sum_{k \in K : o(k) = i} W^k$) and, $\underline{N}$ be the sorted set of nodes on an increasing order based on their *distance from the other nodes* ($D_i = \sum_{a \in A : a_1 = i} C_a$).

6.4.1 Constructive Phase

As mentioned, the constructive phase has three steps. The first one focuses on selecting hub facilities, the second one determines the prices on arcs, and the third one selects the routing of commodities. After finding the routes for the solution network and prices, we obtain a valid lower bound by using the objective function of the leader defined in POA. In what follows, we explain each step in detail.

6.4.1.1 Selection of Hubs

Given that there are no set-up cost on hub arcs, by selecting the set of open hubs, the hub-and-spoke network can be obtained. That is, when there is no set-up cost on the arcs, the network design decisions become trivial and the hubs are fully interconnected. As mentioned, MH_1 and HM_2 differ in the way the initial hub network is constructed. Their main difference is that MH_1 constructs many solution networks at each iteration while the output of hub selection step for MH_2 is one network.

• **Hub Selection in MH$_1$:** At each iteration, from the set $\bar{N}$ (set of ordered nodes based on the highest demand), we select the first M nodes to be candidate locations for the hubs, where $M \sim U[m_1, m_2]$ and m_1, m_2 are integer numbers less than or equal to $|N|$. Next, we define the set $\Gamma = \{\tilde{h} : \tilde{h} \subseteq M \text{ and } |\tilde{h}| = n\}$ as the set of subsets of M with cardinality Π where $\Pi \sim U[\Pi_1, \Pi_2]$ and Π_1, Π_2 are integers less than M. That it, at each iteration, we generate $|\Gamma| = \binom{M}{\Pi}$ different subsets of open hubs where the elements of each subset $\tilde{h}$ are from the set of Π-combinations of M. For each set $\tilde{h}$, we proceed with the pricing and routing steps as explained in Sections 6.4.1.2 and 6.4.1.3 to obtain a complete feasible solution. We try all the combinations until there is no improvement on the lower bound after $r < |\Gamma|$ solutions. We start the next iteration by selecting different values for M and Π while m_1, m_2, Π_1 and Π_2 do not change.

Figure 6.1 shows an example of a hub selection step for a network with $|N| = 10$ for MH_1. Let $M = 5$ and $\Pi = 3$ and the first 5 elements of the set $\bar{N}$ be $\{4, 2, 5, 7, 1\}$. So, the total number of subsets of M with the cardinality of 3 are $|\Gamma| = \binom{5}{3} = 10$. That is, we obtain 10 solution networks with 10 different sets of open hubs $\tilde{h}$.

• **Hub Selection in MH$_2$:** At each iteration from the set $\bar{N}$ (set of ordered nodes based on the highest demand), we select the first 70% of the nodes and add them to the set of open hubs $\tilde{h}$. Next, we apply destroy/ repair strategies. The repair strategy randomly selects G nodes from the set N, where $G \sim U[g_1, g_2]$ and g_1, g_2 are integer numbers less than or equal $|N|$. Then, adds the selected nodes to the set of open hubs $\tilde{h}$, if they are not already there. The destroy strategy selects the first 40% of the elements from the set $\underline{N}$ (set of ordered nodes based on the highest distance from the other nodes) and removes them from the set $\tilde{h}$, if they are already in the set. Figure

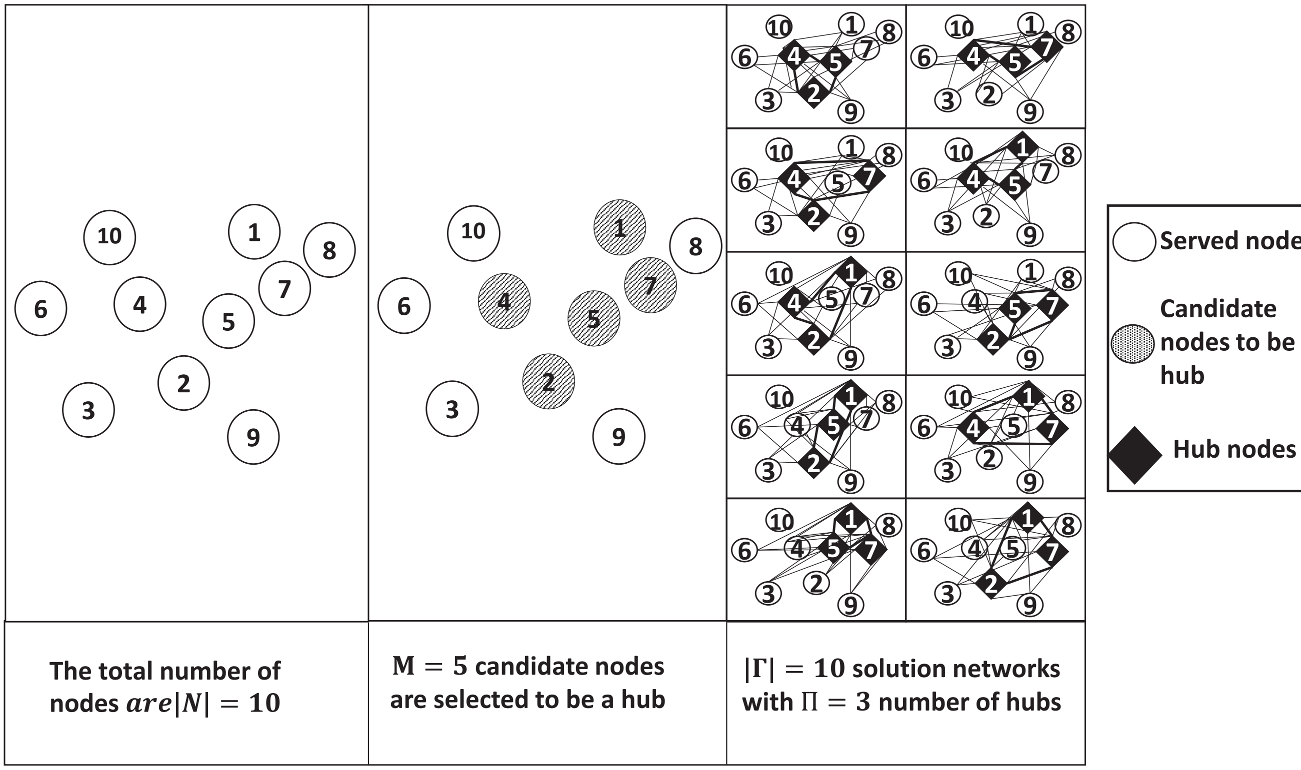


Figure 6.1: Example of a hub selection step of MH_1

6.2 shows an example of a hub selection step for a network with $|N| = 10$ number of nodes for MH_2 with $|N| = 10$ and $G = 2$. The first 7 (70%) elements of the set $\bar{N}$ are $\{4, 2, 5, 7, 1, 8, 6, 3\}$ and the first 4 (40%) elements of the set $\underline{N}$ are considered to be $\{3, 9, 6, 8\}$.

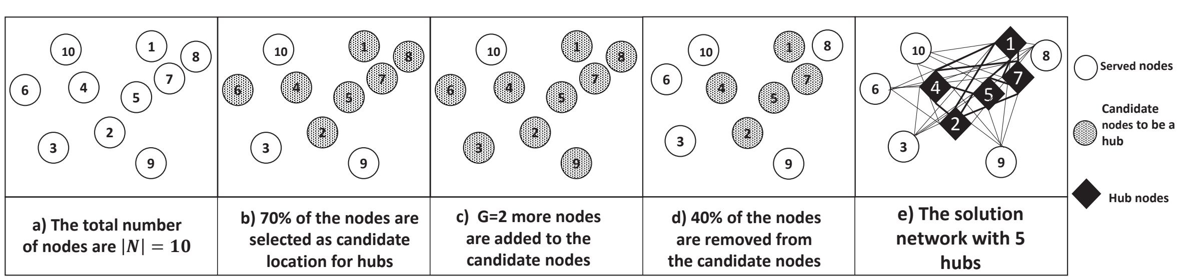


Figure 6.2: Example of a hub selection step of MH_2

6.4.1.2 Solving the Pricing Subproblem

Once an initial solution network has been obtained from the previous step, the goal of the second step is to find a good feasible price for each arc of this network. To achieve this goal, we first construct feasible temporary routes for some commodities in such a way that all the arcs are used at least for one commodity (to guarantee that a price will be set on each arc). We call them temporary because we only use them in this step to set the pricing on arcs and we disregard them afterwards. The way we construct these paths is as follows:

▷ For each $k \in K$ such that $o(k), d(k) \in \tilde{h}$: a feasible path for k where both O/D nodes are open as hubs, is $(o(k), o(k), d(k), d(k))$, which corresponds to the routing solution $(\tilde{U}^k_{o(k)},\ \tilde{x}^k_{(o(k),d(k))},\ \tilde{V}^k_{d(k)}) = (1, 1, 1)$. That is, we select a one-leg path using the hub arc $(o(k), d(k)) \in A$.

▷ For each $k \in K$ such that $o(k) \in \tilde{h},\ d(k) \notin \tilde{h}$: a feasible path for k where its origin is a hub and its destination is a non-hub node, is $(o(k), o(k), o(k), d(k))$, which corresponds to the routing solution $(\tilde{U}^k_{o(k)},\ \tilde{V}^k_{o(k)}) = (1, 1)$. That is a one-leg path using the arc $(o(k), d(k)) \in A$.

▷ For each $k \in K$ such that $o(k) \notin \tilde{h},\ d(k) \in \tilde{h}$: a feasible path for k where its origin is a non-hub and its destination is a hub node, is $(o(k), d(k), d(k), d(k))$ which corresponds to the routing solution $(\tilde{U}^k_{d(k)},\ \tilde{V}^k_{d(k)}) = (1, 1)$. That is a one-leg path using the arc $(o(k), d(k)) \in A$.

Note that there is no need to consider the commodities in which none of their OD pairs are hubs because the access arcs and the hub arcs on the route of those commodities are already considered by the above three cases. After constructing the temporary paths, to solve the pricing problem with the existing network and paths, we only need to solve an optimization problem with the objective of maximizing the total revenue of routing commodities. The only constraint that exists is to set prices of arcs in such a way that for each commodity $k \in K$, the total price of the commodity does not exceed the existing price of the competitor (q^k) and the only decision variable of this problem is the price of each arc ($\tilde{P}_a, \ \forall a \in A$). This is an LP problem that can be formulated as follows:

$$
\begin{aligned}
(PRICE) \quad \text{maximize} \quad & \sum_{a \in A} \sum_{k \in K: a_1 = o(k)} \tilde{P}_a U_{a_2}^k + \sum_{a \in A} \sum_{k \in K: a_2 = d(k)} \tilde{P}_a V_{a_1}^k \\
& + \sum_{a \in A} \sum_{k \in K} \tilde{P}_a x_a^k \\
\text{subject to} \quad & \sum_{a \in A: o(k) = a_1} \tilde{P}_a U_{a_2}^k + \sum_{a \in A: d(k) = a_2} \tilde{P}_a V_{a_1}^k \\
& + \sum_{a \in A} \tilde{P}_a x_a^k \leq q^k \qquad k \in K \qquad (6.102) \\
& \tilde{P}_a \geq 0 \qquad a \in A \qquad (6.103)
\end{aligned}
$$

This is a linear program problem that can be solved by a general solver very efficiently. After finding the prices, we reset all the routing variables to zero and proceed to the next step.

6.4.1.3 Solving the Routing Subproblem

Given the solution network and prices from the previous steps, the routing subproblem has two objectives: maximizing the total profit of routing commodities (the objective of the leader) and minimizing the total price of routing each commodity (the objective of the follower). To better clarify the idea of the routing subproblem, we provide an example in Figure 6.3. This figure shows possible routes for a commodity after fixing the network and prices. The diamonds show the hub nodes and the circles represent the served nodes. The network solution of this example includes three hubs $(1, 2, 3)$ and the OD pair of the commodity is $(15, 24)$. The numbers on the arcs are in the form $P_a(C_a)$, where P_a represents the unit price on each arc a and C_a the unit transportation cost. The price of the competitor for this commodity ($q^{(15,24)}$) is assumed to be 30 and the discount factor α on the hub arcs is assumed to be 0.5. The optimal path is the path that has the price low enough for the customer (less than or equal to the price of the competitor) and gives the maximum profit to the leader. The optimal path for this instance is highlighted by solid lines which is $(15-1-2-3-24)$ with the unit price of $4+12+5+9=30$ and unit transportation cost of $6+0.5\times 5+0.5\times 3+6 = 16$. So, the per unit profit of routing this commodity through this path is $30-16=14$ units which is the maximum profit possible among all the paths. So, any path with the price higher than 30 (such as path $15-3-1-24$) cannot be optimal because the customer prefers to pay less and chooses the competitor. Also, any path with a price less than or equal to 30 but not maximum profit, such as path $15-1-2-24$, is not optimal for the leader because he/she wants to maximize his/her profit.

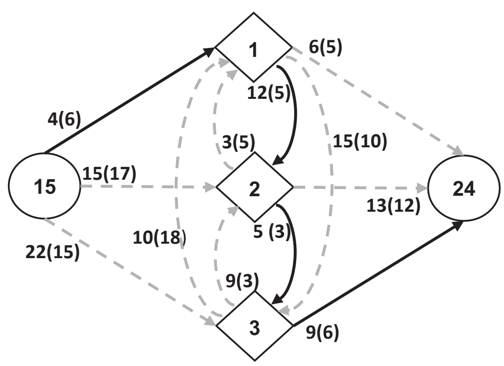


Figure 6.3: Example of possible routes for a commodity with OD pairs =(15, 24) on a given network.

Based on the above remark, the routing subproblem can be solved in two phases. First, for each commodity $k \in K$, we compute the length of the shortest path (ξ^k) (relative to the current set of prices) between its origin and destination on the existing hub-and-spoke network using the well-known Floyd–Warshall algorithm. Second, after finding the length of the shortest path for each commodity, there might be more than one path having the same length. To retrieve the path that has the minimum price and maximum profit (the difference between the revenue and the transportation cost), we need to take into account two assumptions :*i)* there is a discount factor on the transportation cost of the hub arcs which gives priority to the paths that have more hub arcs, *ii)* the leader will be chosen when the prize he offers for some commodity is the same as that of the competitor. A MIP formulation for the routing subproblem associated with each commodity $k \in K$ that reflects these assumptions is the following:

$$
\begin{aligned}
(ROUTE) \quad \text{maximize} \quad & \sum_{a \in A: a_1 = o(k)} W^k (P_a - C_a) \tilde{U}^k_{a_2} + \sum_{a \in A: a_2 = d(k)} W^k (P_a - C_a) \tilde{V}^k_{a_1} \\
& + \sum_{a \in A} W^k (P_a - \alpha C_a) \tilde{x}^k_a \\
\text{s.t.} \quad & \sum_{a \in \delta^-(i)} \tilde{x}^k_a + \tilde{U}^k_i \leq z_i \qquad i \in \tilde{h} \qquad (6.104) \\
& \sum_{a \in \delta^+(j)} \tilde{x}^k_a + \tilde{V}^k_j \leq z_j \qquad j \in \tilde{h} \qquad (6.105) \\
& \sum_{i \in \tilde{h}} \tilde{U}^k_i + \tilde{y}^k = 1 \qquad (6.106) \\
& \sum_{j \in \tilde{h}} \tilde{V}^k_j + \tilde{y}^k = 1 \qquad (6.107) \\
& U^k_i + \sum_{a \in A: a_2 = i} \tilde{x}^k_a = \tilde{V}^k_i + \sum_{a \in A: a_1 = i} \tilde{x}^k_a \qquad i \in \tilde{h} \qquad (6.108) \\
& \sum_{a \in A: o(k) = a_1} P_a \tilde{U}^k_{a_2} + \sum_{a \in A: d(k) = a_2} P_a \tilde{V}^k_{a_1} \\
& + \sum_{a \in A} P_a \tilde{x}^k_a + \sum_{k \in K} q^k \tilde{y}^k = \varsigma^k \qquad (6.109) \\
& \tilde{x}^k_a \in \{0, 1\} \qquad a \in A \qquad (6.110) \\
& \tilde{U}^k_i, \tilde{V}^k_i \in \{0, 1\} \qquad i \in \tilde{h} \qquad (6.111) \\
& \tilde{y}^k \in \{0, 1\}. \qquad (6.112)
\end{aligned}
$$

The decision variables of the above model are only the routing variables since the location and pricing decisions are known by the previous steps. The objective function maximizes the total profit when routing commodity $k \in K$. Constraints (6.104) to (6.108) take care of the feasibility of the path. Constraint (6.109) forces commodity k to be routed through the path that has the minimum price and $\varsigma^k = \min\left\{\xi^k, q^k\right\}$ is the the value of the minimum price path for a commodity which is the minimum of the price of competitor (q^k) and the length of the shortest path (ξ^k). Note that the above optimization problem identifies the commodities whose routing is profitable for the leader. That is, according to the maximization objective function the leader cannot choose commodities with negative profit and, accordingly, those commodities will be routed by the competitor.

6.4.1.4 Evaluation of the Lower Bound

By following the above steps, we can construct a feasible solution $(\tilde{z}, \tilde{P}, \tilde{y}, \tilde{U}, \tilde{V}, \tilde{x})$ at each iteration. The last step in the constructive phase is to obtain a valid lower bound. This is done by evaluation the original objective function of POA using the obtained feasible solution:

$$\underline{\eta} = \sum_{a \in A} \sum_{k \in K: a_1 = o(k)} W^k(\tilde{P}_a - C_a)\tilde{U}^k_{a_2} + \sum_{a \in A} \sum_{k \in K: a_2 = d(k)} W^k(\tilde{P}_a - C_a)\tilde{V}^k_{a_1}$$
$$+ \sum_{a \in A} \sum_{k \in K} W^k(\tilde{P}_a - \alpha C_a)\tilde{x}^k_a - \sum_{i \in H} f_i \tilde{z}_i.$$

Algorithm (6.2) summarizes the constructive phase of the heuristic we developed. Let $\tilde{\underline{\eta}}$ be the best lower bound and $\underline{\eta}^*$ be a lower bound at each iteration and let $\tilde{S} = (\tilde{z}, \tilde{P}, \tilde{y}, \tilde{U}, \tilde{V}, \tilde{x})$ denote the feasible solution at each iteration.

Algorithm 6.1 Constructive Phase

Initialize $\tilde{S} \leftarrow 0$, $\underline{\eta} \leftarrow 0$,$\underline{\eta}^* \leftarrow 0$
iteration $\tilde{t}$
  Select the set of open hubs using MH_1 *or* MH_2
  Solve $PRICE$ *to define the prices*
  Solve $ROUTE$ *to define the set of routed commodities*
  Evaluate $\underline{\eta}$
   If $\underline{\eta} > \underline{\eta}^*$
    $\underline{\eta}^* = \underline{\eta}$
$\tilde{t} \leftarrow \tilde{t} + 1$

6.4.2 Local Search Phase

We apply a *variable neighborhood search* (VNS) heuristic on the most promising solutions obtained from the constructive phase. Since both the constructive and the local search phases are expensive due to the required computational effort to solve the routing subproblem to optimality, we only apply the local search whenever we obtain a better solution on the constructive phase or on certain iterations. The proposed VNS starts from an initial solution and applies four systematic search methods to find neighbor solutions. If a neighbor solution is better than the current best solution, that solution will be accepted. If not, it will continue searching until all neighbor solutions are examined. Then, it starts the new search method. The accepted solution will be the basics of the next neighborhood search.

The neighborhoods we consider explore solutions where the set of open hubs changes. We explore four neighborhood structures which consider solutions within the feasible domain. Let $S = (z, P, y, U, V, x)$ be the best solution obtained from the constructive phase and the set h as the set of open hubs associated with the obtained solution ($h = \{i : z_i = 1 : i \in H\}$). We update the set of h to the set h' by adding or removing elements. Accordingly, the new value for the prices (P') and routing variables (y', U', V', x') are obtained by solving the routing and pricing subproblems to optimality for the updated set of hubs.

The first neighborhood we explore is *open/close* which considers a subset of feasible solutions that are obtained by opening a new hub and closing an open hub. That is:

$$N_{open\backslash close} \subset \{S' = (z', P', y', U', V', x') : h' = h \backslash \{m\} \cup \{n\},\ m \in h,\ n \in N \backslash h,$$
$$P' \in \arg\max\ PRICE,\ and\ (y', U', V', x') \in \arg\max\ ROUTE\}.$$

The second type of neighborhood we explore is *close-close* where closes two hubs that are open in the current best solution S.

$$N_{close-close} \subset \{S' = (z', P', y', U', V', x') : h' = h \backslash \{m, n\},\ m, n \in h,$$
$$P' \in \arg\max\ PRICE,\ and\ (y', U', V', x') \in \arg\max\ ROUTE\}.$$

The third type of neighborhood considers a subset of feasible solutions that are obtained from the current best solution S.

$$N_{open} \subset \{S' = (z', P', y', U', V', x') : h' = h \cup \{m\},\ m \in N \backslash h,$$
$$P' \in \arg\max\ PRICE,\ and\ (y', U', V', x') \in \arg\max\ ROUTE\}.$$

We explore all the neighborhoods and perform all improvement moves.

Finally, the last neighborhood we try is *close* neighborhood which considers a subset of feasible solutions that are obtained from the current best solution S by closing one hub. Then,

$$N_{close} \subset \{S' = (z', P', y', U', V', x') : h' = h \backslash \{m\},\ m \in h,$$
$$P' \in \arg\max\ PRICE,\ and\ (y', U', V', x') \in \arg\max\ ROUTE\}.$$

To provide diversity, we also apply VNS every t_{ls} iterations on the current solution of the constructive phase.

Algorithm 6.2 Local Search Phase

Initialization
    Select the set of neighborhood structure N_k*, for* $k = 1, ..., 4$
    Find an initial solution $S = (z, P, y, U, V, x)$
while Stopping criteria not satisfied **do**
    $k \leftarrow 1$
    for $(k \leq 4)$ **do**
        Generate a feasible solution $S' \in N_k$
        If S' *is better than the incumbent,* $S \leftarrow S'$ *and continue the search with* N_k
        If all moves are examined, $k \leftarrow k + 1$
    end for
end while

The stopping criteria that we chose is one of the following: (i) the maximum CPU time ($time_{max}$) or (ii) the maximum number of iterations t_{max}.

6.5 Computational Results

In this section we describe the computational experiments we have run in order to analyze the performance and various aspects of the HLPs with profit on arcs. We do not perform experiments for HLPs with pricing on paths since as we explained in Section 6.2, this class of HLPPs are a special case of UHLPPs introduced in Chapter 3.

Our computational Experiments focus on two aspects. In Section 6.5.2, we first study the empirical computational complexity of the problem and compare and analyze the performance of our two heuristics. In Section 6.5.3, we next study the structure of the solution networks as well as economical aspects of the problem by performing a sensitivity analysis on the parameters of the model.

All experiments were run on an HP station with an Intel Xeon CPU E3-1240V2 processor at 3.40 GHz and 24 GB of RAM under Windows 7 environment. All formulations were coded in C and solved using the callback library of CPLEX 12.6.3. We use a traditional (deterministic) branch-and-bound solution algorithm with all CPLEX parameters set to their default values. In all experiments the maximum computing time was set to 86,000 seconds (one day).

The benchmark instances we have used for the experiments are the well-known CAB data set of the US Civil Aeronautics Board, with additional data that we generated for the missing information. 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 between each pair of cities, W^k, where $o(k) \neq d(k)$. We have considered instances with $n \in \{5, 7, 10, 25, 30, 40, 50, 60, 70, 80, 90, 100\}$ and $\alpha \in \{0.2, 0.5, 0.8\}$. Since CAB instances do not provide the setup costs for opening facilities, we use as the setup cost of opening hubs, i.e. f_i, generated by de Camargo et al. [45]. To have more reasonable solutions, we used $f_i \times \zeta$ where ζ is set to 2. The price of the competitor q^k, $k \in K$, for routing commodities are randomly generated as $q^k = \varphi \sum_{(i,j) \in A_H} F_a^k / |A|$, where φ is a continuous random variable following a uniform distribution $\varphi \sim U[0.3, 0.4]$. The collection and distribution factors are $\chi = \delta = 1$.

6.5.1 Implementation Details of the Heuristics

After some fine-tuning, we set the following parameter values for MH_1. The maximum number of iterations, $t_{max} = 25$. The additional parameters that are used are the following: the number of consecutive solutions with the same combinations without improvement is $r = 10$. We set $M \sim U[3, |N|]$ and $\Pi \sim U[1, 8]$ as the parameters of the network design phase while $\Pi < M$. We use $\underline{\eta} = 0$ as the initial lower bound.

This value is updated whenever the heuristic improves the incumbent solution. We apply the local search phase every $t_{ls} = 10$ or any time there is an improvement on the lower bound.

For MH_2, the maximum number of iterations, t_{max}. We set $G \sim U[1, 8]$ as the initial number of hubs at the beginning of each iteration. We apply the local search phase every $t_{ls} = 40$ or any time there is an improvement on the lower bound. Other parameters are similar to the ones used in MH_1.

6.5.2 Numerical Results for Hub Location Problems with Pricing on Arcs

Table 6.1 provides detailed information of the lower bounds obtained by the two heuristics and the optimal solutions found by CPLEX. The first two columns of each table give some instances data: $|N|$, the number of nodes and α, the discount factor on hub arcs. The next three columns, under the heading *Time (sec)* give computing times in seconds. The three columns under the heading *Profits* indicate the optimal profit found by CPLEX and the lower bounds on the profits obtained by MH_1 and MH_2, respectively. The two columns under *% Dev*, give the percentage deviations of the lower bound produced by each heuristic with respect to the best known solution. These deviations have been computed as $100(v_{best} - v_h)/v_{best}$, where v_h denotes the lower bound by each heuristic and v_{best} is the optimal or best-known value. The next two columns under the header *Hubs* give the number of hubs open in the solution of each heuristic. Finally, the last two columns are the percentage of the market share of the leader obtained from the solutions of each heuristic, which is computed as $100(x/|K|)$ where x is the total commodities routed by the leader and $|K|$ is the total number of commodities. The entries corresponding to instances that could not be handled by CPLEX because of insufficient memory are filled with the sign "$*$".

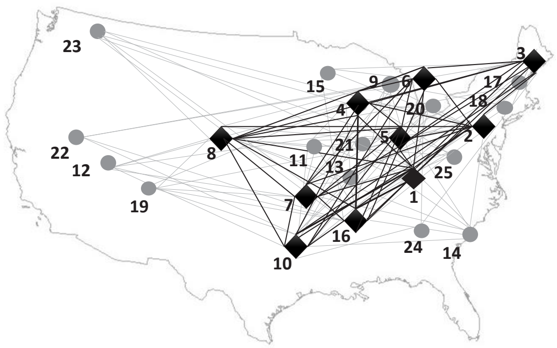
		Time			*Profits*			*% Dev*		*Hubs*		*% Market Share*	
$\|N\|$	α	CPLEX	MH_1	MH_2	CPLEX	MH_1	MH_2	MH_1	MH_2	MH_1	MH_2	MH_1	MH_2
5	0.2	0.08	0.87	0.90	662629.26	662629.26	662629.26	0.00	0.00	2	2	40.00	40.00
	0.5	0.08	0.85	0.77	574776.94	574776.94	574776.94	0.00	0.00	1	1	30.00	30.00
	0.8	0.08	0.77	0.78	574776.94	574776.94	574776.94	0.00	0.00	1	1	30.00	30.00
7	0.2	4.43	5.02	1.81	1935568.70	1935568.70	1935568.70	0.00	0.00	4	4	59.52	59.52
	0.5	0.28	4.86	1.74	1329601.59	1329601.59	1329601.59	0.00	0.00	2	2	30.95	30.95
	0.8	0.28	4.64	1.76	1250039.70	1250039.70	1250039.70	0.00	0.00	1	1	19.05	19.05
10	0.2	*	10.95	7.69	*	2186167.51	2186167.51	0.00	0.00	5	5	60.00	60.00
	0.5	*	10.83	6.68	*	1509353.47	1509353.47	0.00	0.00	2	2	23.33	23.33
	0.8	*	10.66	6.54	*	1359764.72	1359764.72	0.00	0.00	2	2	23.33	23.33
25	0.2	*	452.46	212.98	*	4817486.83	4761101.00	0.00	1.17	10	9	43.50	39.33
	0.5	*	444.69	201.09	*	3305241.83	3305241.83	0.00	0.00	5	5	22.50	22.50
	0.8	*	409.31	201.39	*	2854727.73	2854727.73	0.00	0.00	3	3	11.67	11.67
30	0.2	*	831.42	511.25	*	5187146.44	5187146.44	0.00	0.00	9	9	33.45	33.45
	0.5	*	814.50	489.94	*	3650768.07	3650768.07	0.00	0.00	4	4	15.40	15.40
	0.8	*	816.07	480.99	*	3235522.62	3235522.62	0.00	0.00	4	4	14.71	14.71
40	0.2	*	4951.18	4683.55	*	5755605.01	5683600.99	0.00	1.25	14	12	39.36	33.78
	0.5	*	4477.03	4319.75	*	3794763.42	3794763.42	0.00	0.00	4	4	11.35	11.35
	0.8	*	4469.05	5833.87	*	3455431.93	3309762.88	0.00	4.22	4	4	10.96	10.77
50	0.2	*	17800.58	46969.14	*	6355614.54	6355614.54	0.00	0.00	14	14	31.67	31.67
	0.5	*	17133.33	46595.79	*	4095398.73	4095398.73	0.00	0.00	7	7	16.04	16.04
	0.8	*	17018.97	46490.46	*	3631810.39	3428770.69	0.00	5.59	4	4	8.98	8.65
60	0.2	*	26061.56	74949.34	*	7465693.33	7465693.33	0.00	0.00	15	15	28.25	28.25
	0.5	*	25917.78	73334.61	*	4920486.10	4920486.10	0.00	0.00	8	8	15.90	15.90
	0.8	*	25944.06	73320.78	*	4297049.02	4297049.02	0.00	0.00	5	5	9.52	9.52
70	0.2	*	76450.86	86400	*	7901256.32	7901256.32	0.00	0.00	16	16	25.30	25.30
	0.5	*	75602.29	86400	*	5131974.57	5159831.57	0.54	0.00	10	8	15.78	13.56
	0.8	*	86400	86400	*	4402958.34	4402958.34	0.00	0.00	5	5	8.07	8.07
80	0.2	*	86400	86400	*	8102227.05	7152094.18	0.00	11.73	16	9	22.36	11.88
	0.5	*	86400	86400	*	4738415.02	5193828.53	8.77	0.00	5	8	6.95	10.60
	0.8	*	86400	86400	*	4269732.78	0.00	0.00	100.00	5	0	6.84	0.00
90	0.2	*	86400	86400	*	5442408.79	0.00	0.00	100.00	4	0	4.97	0.00
	0.5	*	86400	86400	*	4711548.67	0.00	0.00	100.00	4	0	4.96	0.00
	0.8	*	86400	86400	*	4318638.14	0.00	0.00	100.00	4	0	4.89	0.00
100	0.2	*	86400	86400	*	3792366.98	0.00	0.00	100.00	2	0	4.97	0.00
	0.5	*	86400	86400	*	3642778.23	0.00	0.00	100.00	2	0	4.97	0.00
	0.8	*	86400	86400	*	3493189.47	0.00	0.00	100.00	2	0	4.37	0.00

Table 6.1: Computational experiments for MH_1 and MH_2 using CAB instances.

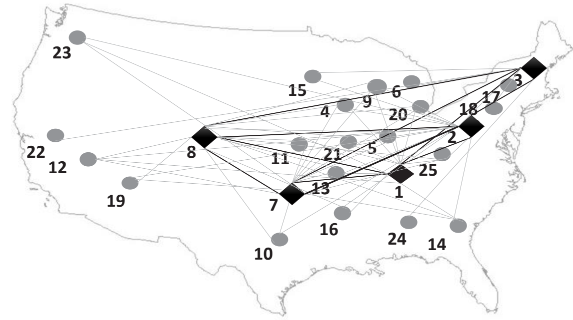
The results shown in Table 6.1 clearly confirms the complexity of the problem since CPLEX is only able to solve very small instances with 5 and 7 nodes. After that, it runs out of memory. Among the two heuristics, MH_2 appears to be faster for instances up to 40 nodes, however, for 50 nodes and more MH_1 is much faster. MH_1 finds a solution for 26 out of 36 instances within the time limit of one day CPU Time while MH_2 is able to find a solution for 24 out of 36 instances. In terms of the quality of the lower bounds, it is confirmed that both of the heuristics provide optimal solutions for the 6 instances that CPLEX is able to solve to optimality. However, for larger instances the optimal solution is not known and the $\%Dev$ presented are with respect to the best known solution. The instances with zero gap for both heuristics are the ones that reach to the same lower bounds using both heuristics (22 out of 36). Among the 14 remaining, MH_1 provides the best known solutions for 12 instances. Moreover, MH_2 is not able to provide a lower bound more than zero for large instances with 80, 90 and 100 nodes because the larger the instance, the harder it is to find the right locations for the hubs. This confirms that MH_1 performs better than MH_2. Comparing the number of hubs and the market share values for both heuristics, it is shown that better bounds are from the results with larger number of open hubs and higher values of market share. Also, the smaller the discount factor, the more hubs are open and the higher market share will be obtained.

6.5.3 Sensitivity Analysis

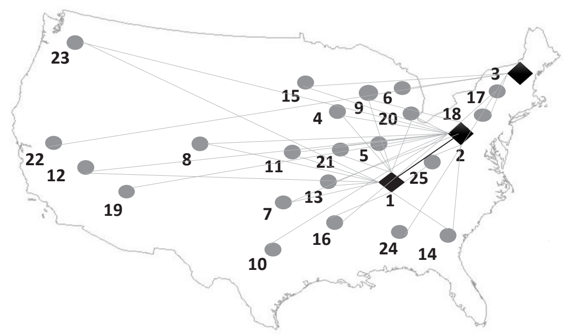
In this section we present a sensitivity analysis of the presented model with respect to some of the parameters. Figure 6.4 allows to compare the effect of the discount factor α on the solution networks. It shows the networks produced by MH_1 (the best known solution) for the CAB instance with $n = 25$ and three different values of the discount factor α. Note that the network design decisions in this problem are trivial



a) $\alpha = 0.2$
Number of hubs= 10
Profit= 4817486.83
Market Share = 43%



b) $\alpha = 0.5$
Number of hubs= 5
Profit= 3305241.83
Market Share = 22%



c) $\alpha = 0.8$
Number of hubs= 3
Profit= 2854727.73
Market Share = 12%

Figure 6.4: Network solution for POA obtained by MH_1 for different discount factors α and $N = 25$.

since there is no set-up costs on the arcs. However, the arcs presented here denotes the arcs that are used by the customer and the rest of the arcs that are not shown are the ones that are not used to transport any commodity. The network for $\alpha = 0.2$ (Figure 6.4a) consists of 10 hubs, and uses 45 hub edges, and serves 15 other nodes with 43% of the market share. However, by increasing the discount factor to $\alpha = 0.5$ (Figure 6.4b), the solution network consists of only 5 hubs and commodities travel from 10 hub edges which causes a considerable reduction in the market share and profit. Figure 6.4c shows the solution network for the highest value $\alpha = 0.8$. Now the number of hub nodes has further decreased to three and the leader only captures 12% of the market share.

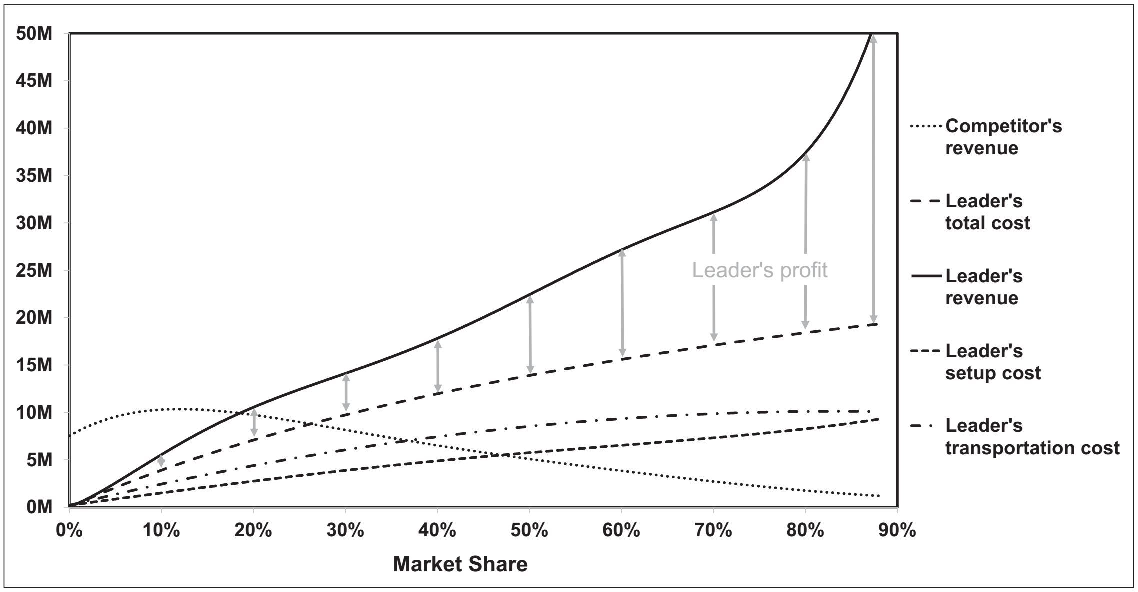


Figure 6.5: The Leader's and the competitor's economical behavior in the market by increasing the market share of the leader, $N = 25, \alpha = 0.5$.

Figure 6.5 represents the economical behavior of the leader and the competitor by changing the market share of the leader. The horizontal axis shows the percentage of the market share of the leader and the vertical axis shows revenue/cost and the gray arrows represents the profit of the leader (revenue-cost). To increase the market share of the leader (decreasing the market share of the competitor) we increased the price of the competitor (q^k) for each commodity $k \in K$. Starting from the zero market share, the revenue of the competitor is high $(8, 021, 28)$ because he has all the market and the leader's profit is zero. By increasing q, the competitor's revenue increases up to 11,576,43 but he is losing the market and accordingly, the leader's market share and revenue increases (4,466,14). However, when q reaches to approximately twice its

Scenario	α	*Hubs*	*%Market share*	*Set-up cost (M)*	*Transportation cost (M)*	*Leader's revenue (M)*	*Leader's profit (M)*	*Competitor's revenue (M)*
S_1	0.5	4	24.00	4.19	6.87	17.70	6.65	12.38
S_2	0.2	5	29.17	5.59	5.98	19.17	7.61	10.91
S_3	0.8	4	23.83	4.19	7.16	17.31	5.96	12.77

Table 6.2: Sensitivity analysis on the discount factor for the CAB instance with $N = 25$.

original value, the revenue of the competitor starts decreasing, and around 8% of the customers are attracted by the leader. As it is shown on the graph, the slope of the reduction on the revenue of the competitor is small at the beginning and that explains why the profit of the leader increases slowly. When the leader has captured %18.75 of the market, the revenues of the competitor and the leader are equal (9,684,41). As the competitor gets weaker, the leader gets stronger and his profit increases more rapidly.

Figure 6.5 also shows the trend of the transportation and set-up costs of the leader. When the leader captures more demand, the number of hubs increases to be able to route more commodities. Accordingly, the transportation cost also increases but with a lower speed and this is due to the discount factor α on the hub arcs which reflects economies of scale.

Table 6.2 to 6.4 give the sensitivity analysis on different parameters of the model: the discount factor (α), price of the competitor (q), and set-up cost on the hubs (f), respectively. κ is a coefficient we used to change the values of the q and ζ is used to change values of f. We study seven different scenarios where S_1 is used as the base scenario. We use both heuristics to obtain a feasible solution for each scenario and compare them. Both of heuristics produce the same solution.

Table 6.2 studies the effect of changing the discount factor. By decreasing α from 0.5 in S_1 to 0.2 in S_2, the number of hubs increases from 4 to 5 and accordingly

Scenario	κ	*Hubs*	*%Market share*	*Set-up cost (M)*	*Transportation cost (M)*	*Leader's revenue (M)*	*Leader's profit (M)*	*Competitor's revenue (M)*
S_1	1.5	4	24.00	4.19	6.87	17.70	6.65	12.38
S_4	1	2	10.00	1.80	2.90	6.64	1.94	13.41
S_5	2	9	59.17	10.75	10.49	35.10	13.85	5.01

Table 6.3: Sensitivity analysis on the competitor's price for the CAB instance with $N = 25$.

Scenario	ζ	*Hubs*	*%Market share*	*Set-up cost (M)*	*Transportation cost (M)*	*Leader's revenue (M)*	*Leader's profit (M)*	*Competitor's revenue (M)*
S_1	4	4	24.00	4.19	6.87	17.70	6.65	12.38
S_6	3	7	39.00	6.25	8.36	22.65	8.03	7.43
S_7	5	3	17.83	4.34	6.40	16.35	5.61	13.73

Table 6.4: Sensitivity analysis on the set-up cost of the hubs for the CAB instance with $N = 25$.

the market share from 24% to %29. Capturing more commodities will increase more the revenues than the transportation costs since commodities benefit more from the discount factor on the hub arcs. Although the set-up costs increase, the reduction on the transportation costs and the increase on the revenues help the leader attain a higher profit, which leads to less revenue for the competitor. On the contrary, in S_3 by increasing α to 0.8, the number of hubs does not change as compared with S_1 while the transportation cost increases and this causes capturing less customers from the market.

Table 6.3 studies the effect of changing the competitor's price. In S_4, 33% reduction on the price of the competitor will give more power to the competitor and and the produce a loss in market capture of 14%.. This incredibly decreases the leader's revenue (62%) while the competitor only experiences a moderate increase on his revenue due to selling commodities at a lower price. S_5 increases the competitor's price to 33% compared to S_1 and this results a huge number of customers (59%) being attracted by the leader and leaving the competitor. Although the competitor is selling products 33% more expensive, his revenue decreases $1,5\%$ while the leader's revenue increases 1%.

Table 6.4 studies the effect for changing the set-up cost of opening the hubs. Obviously, due to the 25% reduction on the set-up cost in S_6, more hubs will be open as compared with S_1, which helps the leader to route more commodities and have a 60% increase in the market share as compared with S_1. Although there is a reduction on the unit set-up cost, the total set-up cost increases due to the 75% increase on the number of open hubs. Accordingly, the revenue and profit of routing more commodities will increase while the revenue of the competitor decreases due to losing part of the market share. On the other hand, in S_7 by increasing the set-up cost to 25%, the leader loses 30% of his market share which leads to a 7.6% reduction on its revenue.

Chapter 7

Conclusions

This dissertation introduced a new class of *hub location problems* (HLPs) called *hub network design problems with profits* (HNDPPs) and seven different alternative models of increasing complexity. Each of these involve interesting features, assumptions and applications that add new avenues of research to the current state-of-the-art in hub location research. HNDPPs are different from the studied HLPs, in their profit oriented objectives which leads to incorporate different types of locational decisions (on the hubs, served nodes), network design decisions (on the hub edges, access edges and bridge edges), operational decisions (on demand levels), and pricing decisions. Moreover, the flexibility arising from the profit maximization objectives, allow these problems to offer different service commitments. All these and still more, deserve to be explored in this new line of research.

We started with the most basic variant of HNDPPs, denoted as the *uncapacitated hub location problem with profits* (UHLPP) in which the selection of a set of nodes to be served is part of the decision process. Potential applications appear in the design of airline and ground transportation networks. We presented a MIP formulation for the problem and we solved it by means of Lagrangean relaxation. A primal heuristics is used to obtain feasible solutions for the problem. Computational results confirm the efficiency of the proposed approach. Benchmark instances involving up to 75 nodes were solved with small optimality gaps.

We further proposed and introduced the foundations of alternative HNDPPs of increasing complexity. These models incorporate different features and relax different unrealistic assumptions usually considered in classical HLPs such as fully connected networks and serving all the nodes. To do so, HNDPPs integrate several locational and network design decisions such as the selection of origin/destination nodes, a set of commodities to serve, and a set of access, bridge and hub edges.

We compared profit oriented models to their traditional cost-oriented hub arc location problems in which all nodes and associated commodities must be served. The solutions of the profit-oriented models do not only outperform the cost-oriented counterparts in terms of the total collected profit, specially for medium and high values of the discount factor α, but also the topology of the obtained solutions networks are substantially different. This clearly illustrates the added-value of integrating within the decision-making process additional strategic decisions on the nodes and the commodities that have to be served. We also compared HNDPPs among them in terms of the profit per served O/D pair and profit per routed unit of flow. Results indicate that the quality of the models, measured in terms of their ability to produce solutions with a better trade-off between their profit and the service level attained, is inversely proportional to their sophistication.

The results of computational experiments also pointed out the significant influence of the discount factor on the design of optimal networks of all HNDPPs. The value of α affects not only the number of hub edges but also number of hubs opened and non-hub nodes activated in solution networks. The higher the value of α, fewer hub edges and hubs are activated as well as fewer nodes are served. Moreover, discount factor also affects the CPU times. It seems clear that the computing times are considerable higher in all the models with smaller values of α. Given the inherent difficulty of the considered models, CPLEX was only able to solve small to medium-size problems.

We next developed an exact algorithmic framework for primary hub network design problems with profits. We considered two variants, which differ from each other in only one set of constraints that forces to route all the commodities with their two end-nodes activated.

We proposed a Lagrangean relaxation that exploits the structure of the problems and can be solved efficiently. In particular, the Lagrangean functions can be decomposed in two independent subproblems: one of them is trivial and the other one can be transformed into a *quadratic boolean problem*, which can be solved efficiently as a max-flow problem. The Lagrangean dual problems were solved with a subgradient optimization algorithm that applied simple primal heuristics, which produced valid lower bounds. The Lagrangean relaxation was embedded within exact branch-and-bound algorithms for each of the considered problems. Moreover, reduction tests were applied at the root node, which helped to considerably reduce the number of variables and constraints. These tests were enhanced with the application of a partial enumeration phase to reduce the number of branches of the enumeration phase.

The results from computational experiments with benchmark instances with up to 100 nodes assessed the efficiency of the proposed framework, and its superiority over CPLEX. On the one hand, because of memory limitations CPLEX was not able to solve instances with more than 60 or 70 nodes, depending on the version of the problem, whereas our proposed solution algorithms did not have this limitation. On the other hand, for the instances where both types of methods could be compared, our algorithms consistently outperformed CPLEX.

In the last part of this dissertation, we went further by introducing hub location problems which consider joint location, pricing and routing decisions that we denoted as *hub location and pricing problems* (HLPPs). In all the considered problems, prices are exogenous. i.e., are defined as part of the inputs of the problem. However, HLPPs incorporate pricing decisions within the decision making process. These problems are interesting in the sense that they are presenting a different environment in HLPs that involves a leader and a follower. The leader (first level) wants to maximize its profit while in the second level, customers want to minimize their objectives. Our proposed models are able to consider both optimization problems by means of bilevel program. Two types of pricing are considered: pricing on paths and pricing on arcs. We showed how both problems can be stated as MIP bilevel programs. We also presented single-level reformulations for these problems. We studied the complexity of each type as well as their features and capabilities in modeling the real world behaviors of firms in the market. We showed that HLPs with pricing on paths are equivalent to UHLPPs.

We developed two variants of a math-heuristic to provide feasible solutions to the HLP with pricing on arcs. The two heuristics differ in the way feasible solutions are constructed. We also improved the solutions by applying a *variable neighborhood search* (VNS). We run some computational experiments on a set of CAB instances with up to 100 nodes. The results showed that CPLEX is not able to solve instances with more than 10 nodes due to memory issues. Moreover, MH_1 performs better than MH_2 both in the quality of the solutions and the CPU time since MH_2 is not able to obtain any feasible solution with a positive objective value after one day of CPU time for instances with 80, 90 and 100 nodes. In the second part of our experiments we analyzed the sensitivity of our model to the parameters such as discount factor α, competitor's price q and, set-up cost of opening hubs f. Solution networks with smaller values of α include more hubs, have higher market share and lead to higher profit while the ones with higher discount factor clearly benefit less from the market. Moreover, we showed how the leader's profit and the competitor's revenue are sensitive to the price of the competitor. Higher prices from the competitor attract more customers to the leader while lower prices give more power to the competitor. Furthermore, increasing set-up cost of the leader decreases its number of hubs and makes it weaker in the market.

Naturally, there exist several aspects related to this work worth to be further investigated that are unfortunately out of the scope of this dissertation. For this reason we briefly describe some future research avenues in which we are interested. We are currently working on more challenging extensions of HNDPPs in which O/D paths can use several hub and bridge arcs. Moreover, improving the proposed heuristic for the HLP with pricing on arcs by applying a more efficient way to solve approximately the routing subproblems instead of optimally solving them is worth studying. Although pricing problems appear to be very complex, they are very applicable and there is a lot of room to develop these models by considering other features. Example of these features are considering triangle inequality on the prices, including more network design decisions and allowing HLPs with pricing on paths to include several hub arcs, among others.

Bibliography

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

[2] Adler, N. and K. Smilowitz (2007). Hub-and-spoke network alliances and mergers: Price-location competition in the airline industry. *Transportation Research Part B 41*(4), 394–409.

[3] Ahuja, R. K., T. L. Magnanti, and J. B. Orlin (1994). Network flows: theory, algorithms, and applications. *Transportation Science 28*(4), 354–356.

[4] Alibeyg, A., I. Contreras, and E. Fernández (2014). A lagrangean relaxation for uncapacitated hub location problems with profits. In *Conference Proceedings of the CLAIO*, pp. in press.

[5] Alibeyg, A., I. Contreras, and E. Fernández (2016). An exact algorithm for hub network design problems with profits. *Submitted to European Journal of Operational Research*.

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

[7] Alumur, S. and B. Y. Kara (2008). Network hub location problems: The state of the art. *European Journal of Operational Research 190*(1), 1–21.

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

[9] Álvarez-Miranda, E., I. Ljubić, and P. Toth (2013). Exact approaches for solving robust prize-collecting steiner tree problems. *European Journal of Operational Research 229*(3), 599–612.

[10] Amaldi, E., M. Bruglieri, and B. Fortz (2011). On the hazmat transport network design problem. In *Network Optimization*, pp. 327–338. Springer.

[11] Aráoz, J., E. Fernández, and O. Meza (2009). An LP based algorithm for the privatized rural postman problem. *European Journal of Operational Research 196*(5), 866–896.

[12] Aras, N., D. Aksen, and M. Tuğrul Tekin (2011). Selective multi-depot vehicle routing problem with pricing. *Transportation Research Part C 19*(5), 866–884.

[13] Asgari, N., R. Z. Farahani, and M. Goh (2013). Network design approach for hub ports-shipping companies competition and cooperation. *Transportation Research Part A 48*, 1–18.

[14] Aykin, T. (1995). Networking policies for hub-and-spoke systems with application to the air transportation system. *Transportation Science 29*(3), 201–221.

[15] Bouhtou, M., A. Grigoriev, S. v. Hoesel, A. F. Van Der Kraaij, F. C. Spieksma, and M. Uetz (2007). Pricing bridges to cross a river. *Naval Research Logistics 54*(4), 411–420.

[16] Bouhtou, M., S. van Hoesel, A. F. van der Kraaij, and J.-L. Lutton (2007). Tariff optimization in networks. *INFORMS journal on computing 19*(3), 458–469.

[17] Bracken, J. and J. T. McGill (1973). Mathematical programs with optimization problems in the constraints. *Operations Research 21*(1), 37–44.

[18] Brotcorne, L., M. Labbé, P. Marcotte, and G. Savard (2000). A bilevel model and solution algorithm for a freight tariff-setting problem. *Transportation Science 34*(3), 289–302.

[19] Brotcorne, L., M. Labbé, P. Marcotte, and G. Savard (2008). Joint design and pricing on a network. *Operations Research 56*(5), 1104–1115.

[20] Bryan, D. L. and M. E. O'Kelly (1999). Hub-and-spoke networks in air transportation: An analytical review. *Journal of Regional Science 39*(2), 275–295.

[21] Campbell, J. F. (1994a). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research 72*(2), 387–405.

[22] Campbell, J. F. (1994b). A survey of network hub location. *Studies in Locational Analysis 6*(6), 31–49.

[23] Campbell, J. F. (2013). A continuous approximation model for time definite many-to-many transportation. *Transportation Research Part B 54*, 100–112.

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

[25] Campbell, J. F., A. T. Ernst, and M. Krishnamoorthy (2002). Hub location problems. In *Facility location: applications and theory*, Volume 1, pp. 373–407. Springer-Verlag, New York, NY.

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

[27] Candler, W. and R. Norton (1977). *Multilevel programming. Technical report 20*. World Bank Development Research Center, Washington, DC.

[28] Church, R. and C. R. ReVelle (1974). The maximal covering location problem. *Papers in Regional Science 32*(1), 101–118.

[29] Colson, B., P. Marcotte, and G. Savard (2005). Bilevel programming: A survey. *4OR 3*(2), 87–107.

[30] Colson, B., P. Marcotte, and G. Savard (2007). An overview of bilevel optimization. *Annals of Operations Research 153*(1), 235–256.

[31] Contreras, I. (2015). Hub location problems. In G. Laporte, F. Saldanha da Gama, and S. Nickel (Eds.), *Location Science*, pp. 311–344. Springer.

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

[33] Contreras, I., J.-F. Cordeau, and G. Laporte (2011b). The dynamic uncapacitated hub location problem. *Transportation Science 45*(1), 18–32.

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

[35] Contreras, I., J.-F. Cordeau, and G. Laporte (2011a). The dynamic uncapacitated hub location problem. *Transportation Science 45*(1), 18–32.

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

[37] Contreras, I. and E. Fernández (2012). General network design: A unified view of combined location and network design problems. *European Journal of Operational Research 219*(3), 680–697.

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

[39] Contreras, I., E. Fernández, and A. Marín (2009b). Tight bounds from a path based formulation for the tree of hub location problem. *Computers and Operations Research 36*(12), 3117–3127.

[40] Contreras, I., E. Fernández, and A. Marín (2010). The tree of hubs location problem. *European Journal of Operational Research 202*(2), 390–400.

[41] Contreras, I., E. Fernández, and G. Reinelt (2012). Minimizing the maximum travel time in a combined model of facility location and network design. *Omega 40*(6), 847–860.

[42] Contreras, I., M. Tanash, and N. Vidyarthi (2016). Exact and heuristic approaches for the cycle hub location problem. *Annals of Operations Research*. DOI 10.1007/s10479-015-2091-2.

[43] Croxton, K. L., B. Gendron, and T. L. Magnanti (2007). Variable disaggregation in network flow problems with piecewise linear costs. *Operations research 55*(1), 146–157.

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

[45] de Camargo, R. S., G. Miranda, and H. P. Luna (2008b). Benders decomposition

for the uncapacitated multiple allocation hub location problem. *Computers and Operations Research 35*(4), 1047–1064.

[46] Dempe, S. (2002). *Foundations of bilevel programming.* Springer Science & Business Media.

[47] Dewez, S. and M. Labbé (2004). *On the toll setting problem, PhD thesis.* Université libre de Bruxelles, Bruxelles.

[48] Dewez, S., M. Labbé, P. Marcotte, and G. Savard (2008). New formulations and valid inequalities for a bilevel pricing problem. *Operations research letters 36*(2), 141–149.

[49] Drexl, M. and M. Schneider (2015). A survey of variants and extensions of the location-routing problem. *European Journal of Operational Research 241*(2), 283–308.

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

[51] Eiselt, H. A. and V. Marianov (2009). A conditional p-hub location problem with attraction functions. *Computers and Operations Research 36*(12), 3128–3135.

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

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

[54] Feillet, D., P. Dejax, and M. Gendreau (2005). Traveling salesman problems with profits. *Transportation Science 39*(2), 188–205.

[55] Fisher, M. L. (1981). The lagrangian relaxation method for solving integer programming problems. *Management science 27*(1), 1–18.

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

[57] Garey, M. R. and D. S. Johnson (2002). *Computers and intractability*, Volume 29. wh freeman New York.

[58] Gelareh, S., R. N. Monemi, and S. Nickel (2015). Multi-period hub location problems in transportation. *Transportation Research Part E: Logistics and Transportation Review 75*, 67–94.

[59] Gelareh, S., S. Nickel, and D. Pisinger (2010). Liner shipping hub network design in a competitive environment. *Transportation Research Part E 46*(6), 991–1004.

[60] Gendron, B., T. G. Crainic, and A. Frangioni (1999). *Multicommodity capacitated network design*. Springer.

[61] Goldman, A. (1969). Optimal locations for centers in a network. *Transportation Science 3*(4), 352–360.

[62] Gouveia, L. and F. Saldanha-da Gama (2006). On the capacitated concentrator location problem: a reformulation by discretization. *Computers & operations research 33*(5), 1242–1258.

[63] Grove, P. G. and M. E. O'Kelly (1986). Hub networks and simulated schedule delay. *Papers in Regional Science 59*(1), 103–119.

[64] Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Operations Research 12*(3), 450–459.

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

[66] Hansen, P., B. Jaumard, and G. Savard (1992). New branch-and-bound rules for linear bilevel programming. *SIAM Journal on scientific and Statistical Computing 13*(5), 1194–1217.

[67] Heilporn, G., M. Labbé, P. Marcotte, and G. Savard (2010a). A parallel between two classes of pricing problems in transportation and marketing. *Journal of Revenue & Pricing Management 9*(1), 110–125.

[68] Heilporn, G., M. Labbé, P. Marcotte, and G. Savard (2010b). A polyhedral study of the network pricing problem with connected toll arcs. *Networks 55*(3), 234–246.

[69] Heilporn, G., M. Labbé, P. Marcotte, and G. Savard (2011). Valid inequalities and branch-and-cut for the clique pricing problem. *Discrete Optimization 8*(3), 393–410.

[70] Hwang, Y. H. and Y. H. Lee (2012). Uncapacitated single allocation p-hub maximal covering problem. *Computers and Industrial Engineering 63*(2), 382–389.

[71] Jaillet, P., G. Song, and G. Yu (1996). Airline network design and hub location problems. *Location science 4*(3), 195–212.

[72] Jeroslow, R. G. (1985). The polynomial hierarchy and a simple model for competitive analysis. *Mathematical programming 32*(2), 146–164.

[73] Joret, G. (2011). Stackelberg network pricing is hard to approximate. *Networks 57*(2), 117–120.

[74] Kantorovich, L. V. (1942). On the transfer of masses. In *Dokl. Akad. Nauk. SSSR*, Volume 37, pp. 227–229.

[75] Kara, B. Y. and M. R. Taner (2011). Hub location problems: The location of interacting facilities. In *Foundations of Location Analysis*, pp. 273–288. Springer.

[76] Klincewicz, J. G. (1996). A dual algorithm for the uncapacitated hub location problem. *Location Science 4*(3), 173–184.

[77] Kuehn, A. A. and M. J. Hamburger (1963). A heuristic program for locating warehouses. *Management science 9*(4), 643–666.

[78] Labbe, M., G. Laporte, I. R. Martín, and J. J. S. Gonzalez (2004). The ring star problem: Polyhedral analysis and exact algorithm. *Networks 43*(3), 177–189.

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

[80] Labbé, M. and A. Violin (2013). Bilevel programming and price setting problems. *4OR 11*(1), 1–30.

[81] Labbé, M. and H. Yaman (2006). Polyhedral analysis for concentrator location problems. *Computational optimization and applications 34*(3), 377–407.

[82] Labbé, M. and H. Yaman (2008). Solving the hub location problem in a star–star network. *Networks 51*(1), 19–33.

[83] Laporte, G., S. Nickel, and F. S. da Gama (2015). *Location science.* Springer.

[84] Lin, C.-C. and S.-C. Lee (2010). The competition game on hub network design. *Transportation Research Part B 44*(4), 618–629.

[85] Lowe, T. and T. Sim (2012). The hub covering flow problem. *Journal of the Operational Research Society 64*(7), 973–981.

[86] Lüer-Villagra, A. and V. Marianov (2013). A competitive hub location and pricing problem. *European Journal of Operational Research 231*(3), 734–744.

[87] Magnanti, T. L., P. Mirchandani, and R. Vachani (1995). Modeling and solving the two-facility capacitated network loading problem. *Operations Research 43*(1), 142–157.

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

[89] Marianov, V., D. Serra, and C. ReVelle (1999). Location of hubs in a competitive environment. *European Journal of Operational Research 114*(2), 363–371.

[90] Marín, A. (2005a). Formulating and solving splittable capacitated multiple allocation hub location problems. *Computers & operations research 32*(12), 3093–3109.

[91] Marín, A. (2005b). Uncapacitated euclidean hub location: Strengthened formulation, new facets and a relax-and-cut algorithm. *Journal of Global Optimization 33*(3), 393–422.

[92] Marín, A., L. Cánovas, and M. Landete (2006). New formulations for the uncapacitated multiple allocation hub location problem. *European Journal of Operational Research 172*(1), 274–292.

[93] Martins de Sá, E., I. Contreras, and J.-F. Cordeau (2015a). Exact and heuristic

algorithms for the design of hub networks with multiple lines. *European Journal of Operational Research 246*(1), 186–198.

[94] Martins de Sá, E., I. Contreras, J.-F. Cordeau, R. Saraiva de Camargo, and G. de Miranda (2015b). The hub line location problem. *Transportation Science 49*(3), 500–518.

[95] Melkote, S. and M. S. Daskin (2001). An integrated model of facility location and transportation network design. *Transportation Research Part A: Policy and Practice 35*(6), 515–538.

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

[97] Monge, G. (1781). *Mémoire sur la théorie des déblais et des remblais.* De l'Imprimerie Royale.

[98] Nagy, G. and S. Salhi (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research 177*(2), 649–672.

[99] O'Kelly, M. E. (1986). Activity levels at hub facilities in interacting networks. *Geographical Analysis 18*(4), 343–356.

[100] O'Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research 32*(3), 393–404.

[101] O'Kelly, M. E. (1992). Hub facility location with fixed costs. *Papers in Regional Science 71*(3), 293–306.

[102] O'Kelly, M. E. (1998). A geographer's analysis of hub-and-spoke networks. *Journal of transport Geography 6*(3), 171–186.

[103] O'Kelly, M. E. (2012). Fuel burn and environmental implications of airline hub networks. *Transportation Research Part D: Transport and Environment 17*(7), 555–567.

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

[105] O'Kelly, M. E. and H. J. Miller (1994). The hub network design problem: a review and synthesis. *Journal of Transport Geography 2*(1), 31–40.

[106] Orlin, J. B. (2012). Max flows in O(nm) time or better. In *Proceedings of the 2013 Symposium on the Theory of Computing*, pp. 765–774.

[107] Picard, J.-C. and H. D. Ratliff (1975). Minimum cuts and related problems. *Networks 5*(4), 357–370.

[108] Pirkul, H. and D. A. Schilling (1998). An efficient procedure for designing single allocation hub and spoke systems. *Management Science 44*(12), 235–242.

[109] Roch, S., P. Marcotte, and G. Savard (2003). *Design and analysis of an approximation algorithm for Stackelberg network pricing.* Citeseer.

[110] Saberi, M. and H. S. Mahmassani (2013). Modeling the airline hub location and optimal market problems with continuous approximation techniques. *Journal of Transport Geography 30*, 68–76.

[111] Sasaki, M., J. F. Campbell, A. Ernst, and M. Krishnamoorthy (2014). A Stackelberg hub arc location model for a competitive environment. *Computers and Operations Research 47*, 27–41.

[112] Sasaki, M., J. F. Campbell, M. Krishnamoorthy, and A. T. Ernst (2014). A stackelberg hub arc location model for a competitive environment. *Computers & Operations Research 47*, 27–41.

[113] Sasaki, M., A. Suzuki, and Z. Drezner (1999). On the selection of hub airports for an airline hub-and-spoke system. *Computers and Operations Research 26*(14), 1411–1422.

[114] Shioda, R., L. Tunçel, and T. Myklebust (2011). Maximum utility product pricing models and algorithms based on reservation price. *Computational Optimization and Applications 48*(2), 157–198.

[115] Smith, H. K., G. Laporte, and P. R. Harper (2009). Locational analysis: highlights of growth to maturity. *Journal of the Operational Research Society 60*, S140–S148.

[116] Toregas, C., R. Swain, C. ReVelle, and L. Bergman (1971). The location of emergency service facilities. *Operations Research 19*(6), 1363–1373.

[117] Vicente, L., G. Savard, and J. Júdice (1994). Descent approaches for quadratic bilevel programming. *Journal of Optimization Theory and Applications 81*(2), 379–399.

[118] Winter, P. (1987). Steiner problem in networks: a survey. *Networks 17*(2), 129–167.

[119] Yaman, H. (2004). Concentrator location in telecommunications. *Quarterly Journal of the Belgian, French and Italian Operations Research Societies 2*(2), 175–177.

[120] Zanjirani Farahani, R., M. Hekmatfar, A. B. Arabani, and E. Nikbakhsh (2013).

Hub location problems: A review of models, classification, solution techniques, and applications. *Computers & Industrial Engineering 64*(4), 1096–1109.