Hub Network Design and Discrete Location: Economies of Scale, Reliability and Service Level Considerations

Moayad Tanash

A Thesis

In the Department

of

Mechanical, Industrial and Aerospace Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy (Industrial Engineering) at

Concordia University

Montréal, Québec, Canada

April 2018

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Moayad Tanash**

Entitled: **Hub Network Design and Discrete Location: Economies of Scale, Reliability and Service Level Considerations**

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 Committee:

______________________ Chair
Dr. Andrea Schiffauerova

______________________ External Examiner
Dr. Maria Albareda-Sambola

______________________ External to Program
Dr. Ali Akgunduz

______________________ Examiner
Dr. Onur Kuzgunkaya

______________________ Examiner
Dr. Satyaveer Chauhan

______________________ Thesis Co-Supervisor
Dr. Ivan Contreras

______________________ Thesis Co-Supervisor
Dr. Navneet Vidyarthi

Approved by ______________________
Dr. Ali Dolatabadi, Graduate Program Director

11 June, 2018 ______________________
Dr. Amir Asif
Dean, Faculty of Engineering and Computer Science

Abstract

Hub Network Design and Discrete Location: Economies of Scale, Reliability and Service Level Considerations

Moayad Tanash, Ph.D.

Concordia University, 2018

In this thesis, we study three related decision problems in location theory. The first part of the dissertation presents solution algorithms for the cycle hub location problem (CHLP), which seeks to locate p-hub facilities that are connected by means of a cycle, and to assign non-hub nodes to hubs so as to minimize the total cost of routing flows through the network. This problem is useful in modeling applications in transportation and telecommunications systems, where large setup costs on the links and reliability requirements make cycle topologies a prominent network architecture. We present a branch-and-cut algorithm that uses a flow-based formulation and two families of mixed-dicut inequalities as a lower bounding procedure at nodes of the enumeration tree. We also introduce a greedy randomized adaptive search algorithm that is used to obtain initial upper bounds for the exact algorithm and to obtain feasible solutions for large-scale instances of the CHLP. Numerical results on a set of benchmark instances with up to 100 nodes confirm the efficiency of the proposed solution algorithms. In the second part of this dissertation, we study the modular hub location problem, which explicitly models the flow-dependent transportation costs using modular arc costs. It neither assumes a full interconnection between hub nodes nor a particular topological structure, instead it considers link activation decisions as part of the design. We propose a branch-and-bound algorithm that uses a Lagrangean relaxation to obtain lower and upper bounds at the nodes of the enumeration tree. Numerical results are reported for benchmark instances with up to 75 nodes. In the last part of this dissertation we study the dynamic facility location problem with service level constraints (DFLPSL). The DFLPSL seeks to locate a set of facilities with sufficient capacities over a planning horizon to serve customers at minimum cost while a service level requirement is met. This problem captures two important sources of stochasticity in facility location by considering known probability distribution functions associated with processing and routing times. We present a nonlinear mixed integer programming formulation and provide feasible solutions using two heuristic approaches. We present the results of computational experiments to analyze the impact and potential benefits of explicitly considering service level constraints when designing distribution systems.

Acknowledgments

Alhamdulil-Allah, all praises are belong to Allah who has given me the patience and perseverance to successfully accomplish my Ph.D. study. May the blessing and peace of Allah be upon the messenger, Mohammed S.A.A.W.

My deep gratitude goes to Dr. Ivan Contreras for the continuous support of my Ph.D study and related research, frothier patience, motivation, and immense knowledge. Also, I would like to thank my co-supervisor Dr. Navneet Vidyarthi for his constant guidance, suggestions, and cooperation in helping to finish this thesis.

I would like to express my special gratitude towards my family for the encouragement which helped me in completion of this thesis. My beloved and supportive wife, ASEEL ALQUDAH who always be my side when times I need her most. My parents, Omar and Eman have been extremely supportive of me throughout this entire process and has made countless sacrifices to help me get to this point.

I must express my very profound gratitude to Carlos Armando Zetina for providing me with unfailing support and continuous encouragement throughout of writing this thesis. would like also to thank my friends Rami Assad, Othman Alsmadi, Masoud Madani and Ayman Oweida for all their support. Without such friends behind me, I doubt that I would be in this place today.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Location analysis and network design are challenging classes of NP-hard combinatorial optimization problems. Such problems arise in many practical application areas such as: public transportation, telecommunication, emergency service facilities, production planning, and computer system among others. Location problems aim to select a set of nodes to locate facilities and to design the allocation pattern of customers to facilities. Network design problems aim to connect customers to facilities and facilities between themselves by activating a set of links in the underlying network. Well known examples from location analysis are weber problems, p-center, and p-median problems which do not explicitly consider the design of the interconnecting networks. On the other hand, network design problems such as multicommodity fixed-charge network design and network loading problems do not consider the location of facilities at nodes. Although the separate study of location analysis and network design have led to fruitful theoretical and applied advances in the literature, combining them bring additional complexity and unique challenges to these class of problems.

Discrete location problems are concerned with the selection of one or more facilities (hubs, plants, routers, warehouses...) from a discrete set of possible locations to serve a given set of customers, such that the total associated fixed and operational costs are minimum. Discrete location models have been studied within different disciplines such as Operations Research, Management Science, Economics, Geography, Industrial Engineering, Computer Sciences, and Mathematics. For more details about discrete location, readers may refer to Smith, Laporte, and Harper (2009), Daskin (2011), Eiselt and Marianov (2011), and Laporte, Nickel, and da Gama (2015).

Network design problems involve selecting a set of links from a given underlying network to connect customers with facilities and facilities between themselves. These problems can be divided into single-commodity and multi-commodity. Single-commodity network design problems consider only one type of good to be routed through the network, whereas multi-commodity network design problems considers more than one type of goods. Fixed-charge network design and network loading problems are well-known examples of single-commodity network design problems. Hub-and-spoke and capacitated multi-commodity network design problems are well known examples for multi-commodity network design problems. Interested readers are referred to Magnanti and Wong (1984), Costa (2005), Frangioni and Gendron (2009), and Yaghini and Akhavan (2012).

Combining location and network design decisions is useful for modeling situations in which tradeoffs between network design costs, facility costs, and operational costs have to be made. Such situations arises in the design of telecommunication networks, airline network, less-than-truckload distribution systems, just to name a few. One important class of these problems is *hub location problems* (HLPs), which seek to select a set of nodes as hubs to consolidate and centralize traffic between many origin/destination (O/D) nodes and to allocate demand nodes to hubs in order to route commodities through hub network. HLPs have four classical assumptions: commodities have to be routed via at least one hub, hubs are fully interconnected, economies of scale exists in the form of a constant discount factor that is applied to the flow costs between hub nodes, and distance between nodes satisfy the triangle inequality.

One important goal of this dissertation is to study more realistic models that relax some of the above mentioned assumption in hub location. We focus on the design of hub networks that are not necessary complete, instead, we approach hub location problems from a network design perspective. In addition to location decisions, we consider arc selection decisions. The location decisions involve the selection of a set of nodes at which hub facilities can be located; the arc selection decision addresses the design of the hub network by selecting the links to connect origins, destinations, and hubs, establishing a framework for the routing of commodities through the network.

In the first part of this dissertation, we study the *cycle hub location problem* (CHLP). The CHLP seeks to locate p-hub facilities that are connected with a set of hub arcs by means of a cycle. Each O/D node must be allocated to exactly one hub and flows between pair of nodes have to be routed through the cycle network so as to minimize the total flow cost. The CHLP is a challenging NP-hard problem that combines location and network design decisions. The location decisions focus on the selection of the set of nodes to locate facilities, while the network design decisions focus on the design of the cycle-star network, by selecting the access and hub arcs as well as the routing of flows through the network. This problem was first introduced in Contreras and Fernández (2012) in the context of general network design problems, but to the best of our knowledge, there is no paper in the literature dealing with approximate or exact solution methods for solving it.

Potential applications of models where cycle start networks are used arise when the setup cost of the arcs of the network are very high. When minimizing such setup cost, tree-star topologies are particularly attractive as they minimize the number of links on the network at the expense of containing exactly one path between pair of nodes. However, in the design of reliable networks, cycle topologies are preferred to tree topologies as they offer an alternative path between any pair of nodes when a link connecting two nodes fails for some reason. Applications of these problems arise in the design of telecommunication networks and rapid transit systems planning. Xu, Chiu, and Glover (1999) offer an example in digital data service design and Laporte and Rodríguez Martín (2007) provide additional applications considering cycle-star structures.

We present two mixed integer programming formulations for the CHLP and computationally compare them with a general solver. One of the formulations is used in a branch-and-cut algorithm to obtain optimal solutions for small to medium size instances and to provide lower bounds for larger instances. It uses two families of valid inequalities to improve the linear programming relaxation bounds at some nodes of the enumeration tree. We develop separation heuristics to find violated inequalities efficiently. Moreover, we introduce a metaheuristic based on a greedy randomized adaptive search procedure to obtain initial upper bounds for the branch-and-cut algorithm and to obtain feasible solutions for large-scale instances of the CHLP. In order to evaluate the efficiency and limitations of our algorithms, extensive computational experiments were performed on benchmark instances with up to 100 nodes and 8 hubs.

Fully interconnected networks and the independence of flow discounted costs can be an oversimplification in applications where the costs represent the economies of scale due to the bundling of flows on the hub arcs. Full interconnection between hub nodes could lead to solutions in which hub arcs carry considerably less flow than access arcs, yet the transportation costs are discounted on the hub arcs. It may also be the case that the amounts of flow that are routed on various hub arcs are different, yet the same discount factor is applied across the board. Under the assumption of flow-independent costs and the use of fully interconnected hubs, the overall transportation cost may be miscalculated, and the set of hub nodes selected and the corresponding allocation pattern of O/D nodes to hubs may be suboptimal.

In the second part of this dissertation, we study the *modular hub location problem*

(MHLP). The MHLP considers explicitly the flow dependence of transportation costs based on modular arc costs. Thus, the total transportation cost is estimated not in terms of the per unit flow cost, as in the classical HLPs, but in terms of the number of facility links used on each arc. Our approach can be interpreted in terms of its ability to incorporate multiple capacity levels on the arcs. Another advantage is that it neither assumes a fully interconnected hub network nor a particular topological structure, instead it considers the design of the hub network as part of the decision process.

The assumption of modular (or stepwise) transportation costs is consistent with applications in freight transportation and telecommunications networks. In the case of ground transportation, trucking companies send commodities (e.g., goods, express packages, ordinary mail) along hub arcs between break bulk terminals, and along access arcs between an end-of-line terminal and a break bulk terminal, using one or more trucks. The number and capacity of the trucks and the distance traveled can be used to obtain an accurate estimate of the transportation cost between terminals. An analogous situation is the use of regional and hub airports by air cargo companies to efficiently route commodities between many origins and destinations. The transportation cost between airports can be estimated based on the number and capacity of the cargo planes, together with the distance.

We present two mixed integer programming formulations for the MHLP. The first formulation uses flow variables to compute the flow through hub arcs, whereas the second formulation uses path variables to determine whether a specified hub arc lies on the path between a pair of nodes. We propose a Lagrangean relaxation for the path-based formulation of MHLP by relaxing the linking constraints of the location/allocation and routing variables. This makes it possible to decompose the Lagrangean function into two independent subproblems which can be solved efficiently. We also propose a heuristic algorithm to obtain feasible solutions. To prove optimality, we develop a branch-and-bound algorithm that uses the Lagrangean relaxation and a heuristic to obtain lower and upper bounds at the nodes of the enumeration tree. Computational results on benchmark instances with up to 75 nodes confirm the efficiency and the robustness of the proposed algorithms.

In today's transition into a more competitive economy, companies now need to consider customer preferences and service levels within their strategic decisions. Classically, facility location decisions are studied within a static environment where they remain unchanged throughout a time horizon. However, demand often change in a random manner depends on different factors, such as population shift, and price. If facilities cannot cope with heavy increase in demands, then facilities are called congested. A queue usually builds up in the system resulting in long waiting times. A long waiting time is an important concern of decision makers in FLPs.

In the last part of this dissertation, we study this decision-making process as a *dynamic facility location problem with service level constraints* (DFLPSL). The DFLPSL seeks to locate a set of facilities with sufficient capacities through a planning horizon to satisfy customer demand with a minimal cost while ensuring a minimum service level requirement. We consider three sources of uncertainties: demand level, service time, and travel time. Facilities operate in an $M/M/1$ queue setting and travel time from a facility to a costumer is assumed to follow a gamma distribution. One advantage of our model is that customers are not necessarily served from their closest facility, rather, they are served from the facility that provides a minimum average travel and waiting time. To the best of our knowledge, the service level constraint to be presented in this dissertation is the first to consider the variability of travel time, in addition to service time and demands levels in a closed form expression.

Applications of the DFLPSL can be found in build-to-order systems. In these systems, companies receive confirmed orders from customer's ends, assemble the finished products from multiple components and finally ship the finished products to the customer's ends.

One of the great advantages of the build-to-order systems is the ability to customize products to meet customers expectations without carrying finished good inventory. However, if demands for a given time is large, then facilities may have a substantial lead time. Additionally, considering the stochastic nature of the travel time in delivering finished products might add more delay to total lead time. A long lead time may cause the company to lose potential sales and thereby reduce its profitability.

We present a nonlinear mixed integer programming formulation for the DFLPSL. This formulation capture two important sources of stochasticity in facility location by considering known probability distribution functions associated with processing and routing times. The resulting nonlinear formulation cannot be solved using commercial solvers, therefore, we develop two heuristic approaches to obtain good quality solutions. A series of computational experiments are performed to compare the efficiency of the two proposed solution algorithms.

This dissertation is organized as follows: Chapter 2 presents the literature review related to network hub location and facility location problems with uncertainty. In chapter 4, we present mathematical programming formulations and an exact solution algorithm for the cycle hub location problem. In Chapter 5, we present integer programming formulations and an exact solution algorithm for the single assignment modular hub location problem. Chapter 6 introduces the dynamic facility location problem service level constraints and two heuristic solution algorithms for solving it. Finally, in Chapter 7 we derive some conclusions and comments on future research.

Chapter 2

Literature Review

Facility location and network design are two classes of NP-hard problems which lie at the heart of network optimization. Facility location problems aim to select a set of nodes to locate facilities while network design problems aim to activate a set of links to connect facilities between them. The aim in both facility location and network design problems is to satisfy customers demands at minimal cost. *General network design problems* (GNDPs) offer a unified view of these research directions (Contreras & Fernández, 2012). GNDPs include *design decisions* in which the decisions of selecting facilities and activating links in the underlying network; and *operational decisions*, in which the decisions of allocating customers to facilities and routing there demands through the network have to be made.

A well-known classical location problem is the *uncapacitated facility location problem* (UFLP), which involves locating a set of facilities and allocating customers to facilities in such a way that the total setup and allocation costs are minimized. Facilities are assumed to have unlimited capacity to serve customers, therefore, any facility can satisfy all demands. Another well-known discrete location problem is the *p*-median problem, which aims to select the location of *p* facilities to serve a set of customers so as to minimize the total transportation cost. *Covering location problems*, and *p-center problem* are other well-known facility location problems that involve location-allocation decisions. For more details, readers are referred to Laporte et al. (2015).

Network design problems mainly focus on network design decisions in which a set of links have to be selected from a given underlying network to connect customers to facilities and facilities among themselves. Because of their importance in many practical areas, much research has been carried out in this field. *Network loading problems* (Magnanti, Mirchandani, & Vachani, 1995) and *fixed charge network problems* (Costa, 2005; Holmberg & Hellstrand, 1998; Sun, Farris, Cote, Shoults, & Chen, 1982) are examples in which consider link activation decisions. Other network optimization problems deal with both location and network design decisions. For instance, *location-vehicle routing problems* (Nagy & Salhi, 2007) seek to locate a set of depots and design the route to visit customers. In some other problems, routing decisions have to be made to indicate the route that is used to send flow between customers. However, the routing and network design decisions are closely related, since the network design decisions activate the links that will be used in the path between any pair of nodes.

Generally speaking, GNDPs tackle the design and operational decisions. These problems are useful for modeling a number of applications in which a tradeoff between network design costs, allocation cost, transportation cost and facility setup costs have to be made. Facility location problems, hub location problems, and location-routing problems are well-known examples of GNDPs. Although these problems share most essential aspects, a small difference between them has a significant impact on the type of formulations and the difficulty level of the problems.

GNDPs can be categorized based on the type of customer demand into *user-facility demands* (GNDPs-UF) and *user-user demands* (GNDPs-UU). In GNDPs-UF, facilities provide service to users. Therefore, demands are routed between facilities and users. In fact, these problems are strategic decision problems concerned with locating facilities and allocating users to facilities such that the setup and transportation costs are minimized. Whenever links' setup costs are zero, the network design decisions will allocate each customer to its facility by a direct connection. Nevertheless, when the links' setup costs are non-zero, the network design decisions become non-trivial. For instance, *location-network design problems* (Contreras, Fernández, & Marín, 2010; Melkote & Daskin, 2001), and *location-vehicle routing problems* (Nagy & Salhi, 2007).

In GNDPs-UU, facilities provide connections between users where commodities are consolidated from different origins, transfered demands through the network, and distributed to their final destinations. GNDPs-UU, involve location, allocation, network design and routing decisions. The network design and routing decisions influence the optimal solution structure by selecting the way to connect users to facilities and facilities to each other. Classical HLPs can be seen as GNDPs-UU. Some particular problems consider non-trivial location decisions. *Concentrator location problems* (Labbé & Yaman, 2006; Yaman, 2004), *tree-star location problems* (Contreras et al., 2010), *ring-star problems* (Labbé, Laporte, Martín, & Salazar-González, 2004), and *star-star location problems* (Labbé & Yaman, 2008; Yaman, 2008), are well-known examples of these problems.

GNDPs-UU can be classified based on the type of allocation scheme, i.e. single and multiple allocation problems. In single allocation problems, users must be allocated to exactly one facility whereas multiple allocation problems allow users to be allocated to more than one facility. GNDPs-UU can be also classified based on the topological structure induced by the facilities and the edges of the network used to connect them, i.e., fully interconnected, tree networks, cycle networks, etc.

2.1 Hub Location Problems

HLPs arise in the design of hub-and-spoke networks. They have a wide variety of applications in airline transportation, freight transportation, rapid transit systems, trucking industries, postal operations, and telecommunications networks. These systems serve demand for transportation of passengers, commodities, and/or transmission of information (data, voice, video) between multiple origins and destinations. Instead of connecting every origin/destination (O/D) pair directly, hub-and-spoke networks serve customers via a small number of links, where hub facilities consolidate the flows from many origins, transfer them through the hub level network, and eventually distribute them to their final destinations (see Figure 2.1). The use of fewer links in the network concentrates flows at the hub facilities, allowing economies of scale to be applied on routing costs, besides helping to reduce setup costs and to centralize commodity handling and sorting operations. Broadly speaking, HLPs consider the location of a set of hubs and the design of the hub-and-spoke network so as to minimize the total flow cost.

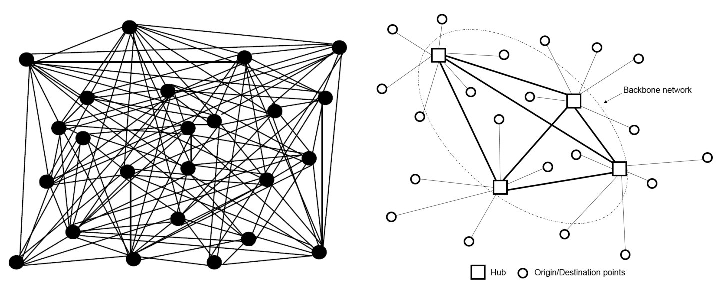


Figure 2.1: Network with direct links vs. hub-and-spoke network

Typical examples of applications of hub-and-spoke networks appear in the design of transportation and telecommunication networks. In the case of transportation networks (i.e., air and freight transportation), the demand corresponds to commodities, such as passengers and goods, that are routed in a transportation mode (i.e., airplanes, truck). Hubs correspond to sorting centers or transportation terminals. The use of hub-and-spoke network encourage these networks to consolidate flows at hub facilities and hub arcs, allowing economies of scale to be applied on the transportation costs (D. L. Bryan & O'Kelly, 1999). In the case of telecommunication networks, applications of hub-and-spoke networks appear in the design of a broad range of networks such as telephone networks, computer communication, etc (J. Klincewicz, 1998). In these networks, demand corresponds to transmission of information (data, voice video) that are transmitted via a physical media (i.e. fiber optic cables) or wireless signals through the air (i.e. microwave, radio, infrared). Hub facilities correspond to electronic devices such as multiplexors, switches, concentrators, etc. Economies of scale in information transmission and network utilization, along with the large set-up cost for hub facilities and communication links encourage the use of hub-and-spoke architecture.

Since the seminal work by O'Kelly (1986b), several classes of fundamental discrete HLPs, such as *p*-hub median problems, uncapacitated hub location problems, *p*-hub center problems, and hub covering problems, have been studied in the literature. For a detailed classification and review of discrete HLPs, the readers are referred to S. Alumur and Kara (2008), Campbell and O'Kelly (2012), Farahani, Hekmatfar, Arabani, and Nikbakhsh (2013), and Contreras (2015). Even though these problems are different on a number of characteristics, mainly due to their particular applications, the vast majority of them share in common four assumptions. The first one is that flows have to be routed via hubs and thus, paths between O/D nodes must include at least one hub. Second, it is possible to connect hubs with more effective pathways that allow a constant discount factor to be applied to the flow cost between hubs. The third assumption is that hub arcs have no setup cost and thus, hub facilities can be connected at no additional cost. The last one is that distances between nodes satisfy the triangle inequality.

The above mentioned assumptions simplify the network design decisions. For example, the last two assumptions allow the backbone network to be fully interconnected (i.e. a complete graph), whereas the access network is determined by the allocation pattern of O/D nodes to hub facilities. Moreover, the combination of the first, third and fourth assumptions results in O/D paths containing at least one hub or at most two hub nodes. This results in HLPs having a number of attractive theoretical features, which have given rise to various mathematical models (Campbell, 1994; Ernst & Krishnamoorthy, 1998a; Hamacher, Labbé, Nickel, & Sonneborn, 2004; Labbé & Yaman, 2004), and sophisticated solution algorithms (Contreras, Cordeau, & Laporte, 2011a; Contreras, Díaz, & Fernández, 2011; Ilić, Urošević, Brimberg, & Mladenović, 2010; Labbé, Yaman, & Gourdin, 2005).

The idea of hub-and-spoke networks was initially addressed in Goldman (1969), then followed by O'Kelly (1986a, 1986b). O'Kelly (1987) introduced the first quadratic mixed integer programming formulation for the *single allocation p-hub median problem* (*p*-HLP). Campbell (1994) proposed the first linear mixed integer programming formulation for the same problem using a set of four indices binary variables. The main idea of this formulation is to track the path of routing commodities between any O/D node. Skorin-Kapov, Skorin-Kapov, and O'Kelly (1996) introduced an important family of formulations called *path-based formulations*. They formulated the *p*-HLP using an additional set of constraints that model the relation between the location$\backslash$allocation and routing variables. An attractive feature of this formulation is that it has a very tight linear programming relaxation bound, yielding integer optimal solutions in most of the tested instances.

Ernst and Krishnamoorthy (1996) proposed another important family of formulations called *flow-based formulations* for the $p-$HLP. The main idea of this formulation is to replace the four indices variables by treating the inter-hub transfers as a multi-commodity flow problem where each commodity represents the outgoing flow from a particular node. In comparison to Skorin-Kapov's et al. (1996) formulation, a flow-based formulation reduces the number of required constraints and variables by a factor n. It is worth noting that both path-based and flow-based formulations have been successfully combined with sophisticated solution algorithms to obtain optimal solutions for large-scale instances

(Cetiner, Sepil, & Sural, 2010; Contreras, Cordeau, & Laporte, 2011a; Labbé, Yaman, & Gourdin, 2005). They have also been adapted to model more realistic variants of HLPs including additional features of real applications (S. Alumur & Kara, 2009; Martins de Sá, Contreras, Cordeau, Saraiva de Camargo, & de Miranda, 2015a; Yaman, 2008; Yaman & Elloumi, 2012).

The earliest reviews on hub location are by Campbell (1994) and O'Kelly and Miller (1994). Campbell (1994) provided a comprehensive survey of hub network problems and presented a classification scheme for different models. O'Kelly and Miller (1994) presented a structured review of research on the hub network design problem. Later, J. Klincewicz (1998) presented key aspects in modeling hub location problems in the particular applications of communications and transportation networks. D. L. Bryan and O'Kelly (1999) provided an analytical review for hub-and-spoke networks in the airline industry. In their survey paper, S. Alumur and Kara (2008) provided a classification of hub location problems, in which they include some of the most recent trends for these problems. Contreras and Fernández (2012) presented a unified framework for the general network design problems and combined location problems with network design problems, in which HLPs are a particular case. Campbell and O'Kelly (2012) introduced some observations of the present status in hub location problems and provided some visions into early motivations in this field. Moreover, they suggested some promising research directions for future work. Farahani et al. (2013) reviewed solution algorithms and applications for several classes of HLPs. As a most recent book chapter in this area, Contreras (2015) reviewed key features, assumptions and properties that are commonly consider in HLPs.

As mentioned earlier, HLPs assume that each pair of hubs are connected via a hub arc. As a result, hub arcs form a complete graph. Although the fully interconnected assumption simplifies the network design decisions, it may be prohibitive in applications where there is a considerable setup cost associated with the hub arcs (see, for instance, J. Klincewicz, 1998; O'Kelly & Miller, 1994). To overcome this deficiency, several models considering incomplete hub networks have been introduced. Campbell, Ernst, and Krishnamoorthy (2005a, 2005b) relaxed the assumption of full interconnection between hubs and incorporate hub arc selection decisions. S. A. Alumur, Kara, and Karasan (2009), and Calık, Alumur, Kara, and Karasan (2009) studied the design of incomplete hub networks with single assignments in which no network structure other than connectivity is imposed on the backbone network. Other works have also proposed different models that do not consider a complete backbone network but rather, a particular topological structure. Contreras, Fernández, and Marín (2009), Contreras et al. (2010), and Martins de Sá, de Camargo, and de Miranda (2013) studied the design of tree-star hub networks in which the hubs have to be connected by means of a tree and each of the O/D nodes follow a single allocation pattern to hubs. These papers focus on the minimization of the total flow cost whereas Kim and Tcha (1992), Y. Lee, Lim, and Park (1996), and Y. Lee, Lu, Qiu, and Glover (1993) focused on the setup costs associated with the design of tree-star networks. Labbé and Yaman (2008) and Yaman (2008) considered the design of star networks in which hub nodes are directly connected to a central node (i.e. star backbone network) and the O/D nodes are assigned to exactly one hub node. Yaman (2009) studied the problem of designing a three-layer hub-and-spoke network, where the top layer consists of a complete network connecting the central hubs, and the second and third layers are unions of start networks connecting the remaining hubs to central hubs and the O/D nodes to hubs, respectively. Yaman and Elloumi (2012) considered the design of two-level star networks, while taking into consideration the service quality in terms of the length of paths between pair of O/D nodes. Martins de Sá, Contreras, Cordeau, Saraiva de Camargo, and de Miranda (2015b) studied the problem of designing a hub-line network in which hubs are connected by means of a line and the aim is to minimize the total service time between pairs of nodes. Other studies on incomplete hub networks include Campbell (2009); Contreras and Fernández

(2014); and Davari, Zarandi, and Turksen (2013).

The independence of flow discounted costs is appropriate in applications in which the links between hubs are associated with faster transportation modes, but it can be an over-simplification in applications where the costs represent the economies of scale due to the bundling of flows on the hub arcs. Several papers have already pointed out that the discount factor should be regarded as function of the flow volume (see, D. L. Bryan & O'Kelly, 1999; Campbell, 2013; O'Kelly, 1998; O'Kelly & Bryan, 1998). O'Kelly and Bryan (1998) were among the first to develop a hub location model that expresses the discount factor on hub arcs as a function of flow. It was later extended by D. Bryan (1998), J. G. Klincewicz (2002) and de Camargo, de Miranda Jr, and Luna (2009). However, their models use a nonlinear cost function to compute the transportation costs on a hub arc as a function of its flow. This function is approximated by a piecewise linear function to obtain a linear integer programming formulation for the problem. Horner and O'Kelly (2001) proposed a nonlinear cost function based on link performance functions; it is designed to reward economies of scale in all arcs in the network. Podnar, Skorin-Kapov, and Skorin-Kapov (2002) formulated a network design model in which the discount factor applies only on arcs that have flows larger than a given threshold; however, the model focuses on the design of the network rather than on the location of the hub facilities. Racunica and Wynter (2005) introduced a nonlinear concave cost hub location model that determines the optimal location of intermodal freight hubs. The cost function models the flow-dependent discounted cost only on origin-to-hub and hub-to-destination legs.

Yoon and Current (2006) and O'Kelly, Campbell, Camargo, and Miranda (2015) adopted a different approach for modeling economies of scale on all the arcs in a hub-and-spoke network. Rather than relying on a nonlinear cost function, they use linear cost functions which combine variable transportation costs for flows on arcs and fixed costs for activating those arcs. In these approaches, the use of fixed costs of arcs allows the link costs per unit of flow to decrease as the flow increases on that link, resulting in the economies of scale. Kimms (2006) presented three different models for hub location problems with fixed and variable costs. In one of the models, the goal is to determine the optimal number of vehicles used on each arc to route flow through the fully interconnected network. Cunha and Silva (2007) designed a hub-and-spoke network for a less-than-truckload trucking company in Brazil based on a nonlinear cost function that allows the discount factor on hub arcs to vary according to the total amount of freight between hubs. Hoff, Peiró, Corberán, and Martí (2017) studied the capacitated single assignment hub location problem in which the capacity of the edges between hubs is increased in a modular way. They assumed that hubs are fully interconnected and does not consider O/D paths containing more than two hub nodes and one hub arc. Campbell et al. (2005b) and Campbell et al. (2005a) study hub arc location problems, in which the goal is to locate a set of hub arcs, therefore, no longer considering a fully interconnected hub network. To some extent, this mitigates the limitations of flow-independent costs.

2.2 Facility Location Problems with Uncertainty

Facility location problems have a wide variety of applications in both the private and public sectors. Whether choosing the locations of manufacturing facilities or locating new retail stores, decision makers are always challenged by difficult tradeoffs between cost and efficiency. Thus, they have to identify the best potential locations, the appropriate capacity levels and allocate a large amount of capital. Because of the investment in any location project incurs in high costs, facilities are expected to remain in operation for an extended time period, therefore, determining facility locations that will continue to be profitable for a lifetime is an important strategic decision.

Since the pioneering work of Hakimi (1964), who introduced median and center location problems, an extraordinary number of generalizations and extensions have been proposed in the literature such as fixed charge facility location problems, multi-objective problems, location-routing problems and hierarchical facility location models. Readers may refer to the books by Mirchandani and Francis (1990) and Daskin (2011) Laporte et al. (2015) for an overview of facility location problems and extensions. Although these models differ from each other according to various assumptions, the vast majority assume that input parameters, including demands, travel time, transportation and operation costs are known with certainty. Therefore, once facilities have been located, they are assumed to remain open for an extended period regardless of the future changes of any of the input parameters. These models are known as a single period or static. Over time, however, some values of the input parameters might vary in an unpredictable manner. Changes in parameters over the time can radically change the attractiveness of a particular location, making the optimal site today, the blunder of investing tomorrow. Thus, it maybe more appropriate to account for parameter variations during the design process by embedding uncertainty directly into the mathematical models.

Dynamic and stochastic approaches are used to deal with parameters uncertainties in location models. The dynamic approach considers a planning horizon that is divided into several time periods each of which has independent deterministic input parameters. The dynamic approach allows changes in the network structure between consecutive time periods. The stochastic approach, on the other hand, assumes that values of the uncertain parameters are random variables with known probability distributions. The two above approaches are different in the sense that the latter considers design parameters as stochastic rather than only subject to known changes from one period to another as in the former.

As mentioned before, classical facility location problems assume that the values of input parameters are certain. If changes in these values can be predicted, it may be more appropriate to plan in advance for future modifications to the network structure. Thus, the question becomes not only “where” to locate facilities but also “when”. For instance, If the total demand for a given commodity or service varies over time, then it might be necessary to locate new facilities, close some of the existing facilities, relocate and/or adjust the total available capacity to meet the upcoming changes. For example, an increase in the total demand in a given period might require the decision makers to increase the total available capacities, relocate facilities, or open new facilities to meet the additional demands. Indeed, such decisions require a capital, however, the extra cost associated with such decisions might be compensated by the reduction in transportation costs. Similarly, a decrease in the total demand for a given period might lead to closure of some existing facilities and/or reduction in the total available capacity to achieve saving on facility operational costs at the expanse of the associated closing costs.

The first paper pointing out the important of considering the effect of the future time dimension in location analysis was presented by Ballou (1968). He considered the location of a single warehouse over a finite planning horizon such that the total profit is maximized. Since then, several classes of the dynamic facility location problems have been studied in the literature. Roodman and Schwarz (1975), Khumawala and Clay Whybark (1976), Roodman and Schwarz (1977), Van Roy and Erlenkotter (1982), Kelly and Marucheck (1984), Galvão and Santibanez-Gonzalez (1992) and Chardaire, Sutter, and Costa (1996) study the uncapacitated version of the problem where a set of uncapacitated facilities should be located dynamically in time to satisfy customer demands so that the total costs are minimized. C. Fong and Srinivasan (1986); C. O. Fong and Srinivasan (1981), S.-B. Lee and Luss (1987), Syam (2000), Correia, Melo, and Saldanha-da Gama (2013), Delmelle, Thill, Peeters, and Thomas (2014), Cortinhal, Lopes, and Melo (2015) and Castro, Nasini, and Saldanha-da Gama (2017) study the capacitated version of dynamic facility location problems where the goal is to identify in which periods the capacity of a given set of facilities has to be expanded to meet the additional demands. Luss (1982), Canel, Khumawala, Law, and Loh (2001), Antunes and Peeters (2001), Melo, Nickel, and Da Gama (2006), Dias, Captivo, and Clímaco (2007), Behmardi and Lee (2008), and Correia and Melo (2017) consider the possibility of both expanding and reducing the available capacities through the planning horizon. Wilhelm, Han, and Lee (2013) study the case in which facilities can be opened, closed, expanded and/or contracted their capacities over time. Other studies such as Wesolowsky and Truscott (1975), Min and Melachrinoudis (1999), Brotcorne, Laporte, and Semet (2003) and Melo et al. (2006) consider the case in which facilities are allowed to be relocated in a given time period, while other studies considered temporarily closing existing facilities (Canel et al., 2001; Chardaire et al., 1996; Dias, Captivo, & Clímaco, 2006; Van Roy & Erlenkotter, 1982). Chrissis, Davis, and Miller (1982), Gunawardane (1982) and Rajagopalan, Saydam, and Xiao (2008) study the dynamic version of the set covering facility location problem.

Recently, Jena, Cordeau, and Gendron (2015) introduced a dynamic facility location model with generalized modular capacities. This model considers not only the location decisions, but also the possibility of relocating existing facilities and dynamically adjusting the available capacities of existing facilities based on the demand changes and the costs involved in capacity changes. Moreover, the authors consider the possibility of temporarily closing and reopening the existing facilities multiple times through the planning horizon. Therefore, this model generalizes a large set of existing dynamic models.

Dynamic facility location problems focus solely on timing issues of locating facilities over a planning horizon. The input parameters for these problems can be forecasted for each period. Since facilities are expected to be in operation for an extended time period, input parameters such as costs and demands are likely to change in a random manner. Therefore, including the probabilistic information that describe the uncertain parameters in location models lead to achieve more realistic models. To this end, uncertainty is described using a probability distribution on the parameters that can be used to minimize the total expected cost using stochastic programming techniques (Berman & Larson, 1985; Larson, 1974; Marianov & Serra, 1998).

Traditionally, most of the facility location models assume facilities have sufficient capacities to respond immediately to demands. In real life problems, however, customers generate streams of stochastic demands for service and facilities have limited capacity and stochastic service times. This combination usually leads to congestion. Thus, customers might need to wait, if they can, until the facility is free to serve them. To address this issue, facility location problems with congestion have been studied in the literature. Usually, a waiting time or a queue length formula derived from a queuing model is introduced into an optimization framework. This brings additional complexity and unique challenges to this class of location problems. For a recent book chapter in this direction, readers may refer to Berman and Krass (2015).

There are three families of approaches to model congestion at facilities. In the first family, congestion is explicitly limited to a target level by including constraints derived from a queuing system. For instance, Marianov and Serra (1998, 2001, 2002) and Marianov and Ríos (2000) present location-allocation models that incorporate congestion at facilities by imposing constraints on the probability that either waiting time or queue length is at least a specific threshold value. They model each facility as multiple servers queuing system. Marianov and Serra (1998), Marianov and Ríos (2000) and Marianov and Serra (2002) assume that the number of servers at each facility is predetermined and the same for all facilities, however, in Marianov and Serra (2001), the number of servers at each facility is considered to be a part of the design process. Silva and De la Figuera (2007) study the capacitated facility location problem with constrained backlogging probabilities. In this model, each facility is assumed to behave as an M/M/1 queue that is required to maintain backlogging probability below a certain level. Indeed, embedding probability constraints into facility location problems result in locating facility with high service quality. However, the complexity of the problems, due to the resulting nonlinear models, limited researchers to heuristic methods for solving the problems.

In the second family of approaches, congestion cost is attached to the objective function to be minimized. Amiri (1997), Amiri (1998), Wang, Batta, and Rump (2002, 2004), Aboolian, Berman, and Drezner (2008), Elhedhli (2006), Syam (2008), Baron, Berman, and Krass (2008) and Castillo, Ingolfsson, and Sim (2009) model each facility as a queuing system where congestion cost is included in the objective function. In Amiri (1997), Amiri (1998), Wang et al. (2002, 2004), Elhedhli (2006) and Syam (2008), the objective function includes waiting time cost of customers, while in Aboolian et al. (2008) and Castillo et al. (2009), the objective function includes the travel time cost and waiting time cost. To ensure a minimum level of service quality at each service facility, Wang et al. (2004) impose an upper bound on the server utilization rate and Wang et al. (2002) ensure that the expected waiting time at any facility does not exceed a predefined threshold value. All of the previous works assume that customers are served from the closest facility. Berman and Drezner (2007) assume that customers do not necessarily travel to the closest facility, but to the facility that provides the minimum travel and waiting time. The objective is to minimize the total travel time and expected waiting time spent at the selected server for all customers. Aboolian, Berman, and Drezner (2009) study a center location model with multiple servers in which customers travel to their closest facility with the objective of minimizing the total travel and waiting time at the server for all customers.

Finally, congestion effects can be included into other factors, such as demand level or population size, where the objective is to minimize lost demand or to maximize covered population. For example, Marianov (2003) studied the congested p-median facility location problem in which the objective is to maximize the total expected demands that is elastic to distance and congestion at the facilities. Berman and Drezner (2006) studied a facility location problem with $M/M/1$ queuing framework where the objective is to maximize the expected demands served by facilities subject to a budget constraint on the average waiting time of customers.

In today's transition into a more competitive economy, companies need to consider customer preference and service level within there strategical decisions. One of the most important service quality measures in facility location problems is the order lead time of products or service. The order lead time is usually defined as the time between placing an order and the delivery time. To ensure satisfaction of different customers regarding delivery time, facility location models usually include distance, time constraints and/or other costs as service level requirements. The so-called set covering problem aims to minimize the total cost of locating a set of facilities such that no customer will be farther than a maximum service radius from a facility (Moon & Chaudhry, 1984), while the maximal covering location problem aims to locate a set of facilities in such a way that coverage is maximized within a predefined coverage radius (Church & ReVelle, 1974). Moore and ReVelle (1982) studied the hierarchical covering problem where the objective is to minimize the number of uncovered population subject to budget and coverage radius constraints. Goldman (1969) followed by Hakimi and Maheshwari (1972) studied the k-center problem where the objective is to locate k facilities to minimize the maximum service distance (or the travel time) between any demand point and its closest facility. In these studies, the service level requirement is modeled using the maximum distance between customer and facility.

Kolen (1983) use a different approach to model service level requirements by relaxing the distance constraint in the covering problem and introducing a penalty cost in the objective function for not serving some of the demand points. Geoffrion and Graves (1974) extended a multicommodity plant and distribution location problem to include customer service level constraints by inducing an upper bound on the average delivery lead time between each demand point and a facility. Cheong, Bhatnagar, and Graves (2005); Correia and Melo (2016) studied the impact of customer sensitivity to the delivery lead-time. They divided customers into two segments based on their sensitivity to the lead time. The first segment consists of customers who require timely demand satisfaction, whereas the second segment comprises customers who accept delayed deliveries. In Cheong et al. (2005), the objective is to minimize the fixed and variable costs in addition to the lost sales cost, while in Correia and Melo (2016), the objective function aims to minimize, in addition to the fixed costs, the variable costs that consider the tardiness costs resulting from delayed deliveries.

In all the above mentioned studies, travel time is assumed to be a deterministic parameter that is determined by the distance between service facility and demand node. In real-life problems, however, travel times are another source of uncertainty, due to traffic and weather conditions, which might influence the location/allocation decisions (Berman & Odoni, 1982; Mirchandani & Odoni, 1979). In this dissertation, we study a more realistic model that considers travel time, demand and service times as uncertain parameters with known probability distributions.

Chapter 3

Exact and Heuristic Approaches for the Cycle Hub Location Problem

In this chapter we study the CHLP, which consist of locating exactly p hub facilities that are connected with a set of hub arcs with a cycle topology. Each O/D node must be allocated to exactly one hub (i.e. single assignment) and flows between pair of nodes have to be routed through the cycle-star network so as to minimize the total flow cost. This problem is useful in modeling applications in transportation and telecommunications systems, where large setup costs on the links and reliability requirements make cycle topologies a prominent network architecture.

The CHLP is a challenging NP-hard problem that combines location and network design decisions. The locational decisions focuses on the selection of the set of nodes to locate facilities, whereas the network design decisions deals with the design of the cycle-star network, by selecting the access and hub arcs as well as the routing of flows through the network. To the best of our knowledge, the CHLP was first introduced in Contreras and Fernández (2012) in the context of general network design problems, but there is no paper in the literature dealing with approximate or exact solution methods for solving it. Most of the material presented in this chapter is published in Contreras, Tanash, and Vidyarthi

(2016).

The CHLP shares some similarities with other network design problems in which a cycle-star network is sought. The so-called ring-star problem (RSP), introduced by Labbé et al. (2004), aims to locate a simple cycle through a subset of nodes with the objective of minimizing the sum of setup costs of the cycle and assignment costs from the vertices not in the cycle to their closest vertex on the cycle. Another closely related problem is the median cycle problem (MCP), studied by Labbé, Laporte, Rodríguez Martín, and Salazar-González (2005). This problem arises in the design of ring-shaped infrastructures and consists of finding a simple cycle that minimizes the setup costs for the cycle, such that the total assignment cost of the non-visited nodes do not exceed a given budget constraint. Current and Schilling (1994) and Gendreau, Laporte, and Semet (1997) study covering versions of the RSP in which all nodes must be within a prespecified distance from the cycle. Baldacci, Dell'Amico, and González (2007) present the capacitated m-ring star problem, which deal with the location of m cycles that pass through a central node and the assignment of nodes to cycles. Y. Lee, Chiu, and Sanchez (1998) and Xu et al. (1999) study the Steiner ring-star problem, in which the cycle only contains Steiner nodes chosen from a given set. Current and Schilling (1994) consider the median tour problem, where a cycle with p nodes has to be located. It is a bicriterion problem, which consists of minimizing the setup cost of the cycle and of minimizing the total assignment cost of nodes to their closest facilities. Liefooghe, Jourdan, and Talbi (2010) study a bi-objective ring-star problem, in which the setup cost of the cycle and the assignment costs are considered. See Labbé, Laporte, and Rodríguez-Martín (1998) and Laporte and Rodríguez Martín (2007) for additional models related to the location of cycle structures on a network.

All the above mentioned problems focus on the minimization of the setup cost for the design of the network and on the assignment of nodes to facilities. Service is given at or from the facilities, so that service demand occurs at nodes. In the case of HLPs, and in particular the CHLP, service demand is between pairs of nodes and the facilities are used as intermediate locations in the routes that connect node pairs. Therefore, in addition to the network design and assignment decisions considered in the above problems the CHLP considers additional routing decisions and focuses on the minimization of the total flow cost between many node pairs. This makes the problem more challenging as the O/D paths need to be known to compute the routing cost. In fact, even if the location of the hub facilities and the assignment of O/D nodes to hubs is known, the problem remains NP-hard as it reduces to the minimum flow cost Hamiltonian cycle problem (see Ortiz-Astorquiza, Contreras, & Laporte, 2015).

The aim of this chapter to present two mixed integer programming formulations for the CHLP and computationally compare them with a commercial solver. The first formulation is based on flow variables that compute the amount of flow that passes through particular hub arcs, whereas the second formulation is based on path variables that determine is a hub arc is used on the path between two pair of nodes. Moreover, we propose exact and heuristic approaches for the CHLP. In particular, the flow based formulation is used in a branch-and-cut (B&C) algorithm to obtain optimal solutions for small to medium size instances and to provide lower bounds for large instances. It uses two families of valid inequalities, which can be seen as an extension of the ixed-dicut inequalities for multicommodity network design problems, to improve the linear programming relaxation bounds at some nodes of the enumeration tree. We develop separation heuristics to find violated inequalities efficiently. Moreover, we introduce a metaheuristic based on a greedy randomized adaptive search procedure (GRASP) to obtain initial upper bounds for the BC algorithm and to obtain feasible solutions for large-scale instances of the CHLP. Finally, to evaluate the efficiency and limitations of our algorithms, extensive computational experiments were performed on benchmark instances with up to 100 nodes and 8 hubs.

The remainder of the chapter is organized as follows. Section 2 provides a formal definition of the problem and presents two MIP formulations. The B&C algorithm and the metaheuristic are presented in Sections. 3 and 4, respectively. The computational results and the analysis are presented in Section. 5.

3.1 Problem Description and Formulations

Let $G = (N, A)$ be a complete digraph, where $N = \{1, 2, \dots, n\}$ represents the set of nodes as well as the potential sites for locating hubs and $A = N \times N$ is the set of arcs. For each ordered pair $(i, j) \in A$, let W_{ij} denote the amount of flow between origin i and destination j. Thus, $O_i = \sum_{j \in N} W_{ij}$ is the total flow originating at node $i \in N$, and $D_i = \sum_{j \in N} W_{ji}$, is the total flow with destination node $i \in N$. The distances, or flow costs d_{ij} between nodes i and j are assumed to be symmetric, however, they may not satisfy the triangle inequality property. Given that hub nodes are no longer fully interconnected, O/D paths on the solution network may contain more than two hub nodes. The per unit flow cost is then given by the length of the path between an origin and a destination, where the discount factor $0 < \alpha < 1$ is applied to all hub arcs contained on the path.

The CHLP seeks to determine the location of exactly p hubs which are connected by means of a cycle, and the routing of flows through the hub-and-spoke network. Each node has to be allocated to exactly one hub and if a node is selected as a hub, then it is self-assigned. The objective is to minimize the total flow cost. In every feasible solution to the CHLP: i) there exist p hub arcs; ii) every hub node is connected with exactly two other hub nodes; iii) the graph induced by the hubs does not contain subtours, and iv) there are exactly two paths between every pair of nodes on the solution network. This makes the CHLP more difficult to formulate and solve than classical HLPs, as the shortest path between O/D nodes, containing an undetermined number of hub nodes and hub arcs, needs to be determined to evaluate the objective function. Note that when $p \in \{1, 2, 3\}$, the CHLP reduces to a classical p-hub median problem in which hubs are fully interconnected.

Before presenting mixed integer programming formulations, we first define the graph of flows $G_F = (N, E_F)$, as the undirected graph with node set N and an edge associated with each pair $(i, j) \in N \times N$ such that $W_{ij} + W_{ji} > 0$. We assume that G_F is made up of a single connected component since otherwise the problem can be decomposed into several independent CHLPs, one for each connected component in G_F. If a particular application requires a single cycle and the graph of flows contains more than one connected component, we can replace these flows of value zero with $W_{ij} = \epsilon > 0$ sufficiently small. In what follows, we present two MIP formulations for the CHLP. The first one uses path variables to determine the set of arcs on each O/D paths, whereas the second one uses ow variables to compute the amount ow routed through a particular arc.

3.1.1 Path-Based Formulation

For each $i, k \in N; i \neq k$, we define binary location/allocation variables,

$$z_{ik} = \begin{cases} 1 & \text{if non hub } i \text{ is allocated to hub } k, \\ 0 & \text{otherwise.} \end{cases}$$

When $z_{kk} = 1$, node k is selected as a hub and assigned to itself. For each $k, m \in N$, $k < m$, we also introduce binary hub arc variables

$$y_{km} = \begin{cases} 1 & \text{if hub arc } (k, m) \text{ is selected,} \\ 0 & \text{otherwise.} \end{cases}$$

Finally, we define for each $i, j, k, m \in N$, we define binary routing variables variables as follows

$$X_{ijkm} = \begin{cases} 1 & \text{if the the flow from } i \text{ to } j \text{ traverses arc } (k,m), \\ 0 & \text{otherwise,} \end{cases}$$

The CHLP can be stated as follows:

$$(PF) \ \min \ \sum_{i \in N} \sum_{k \in N} (c_{ik} O_i + c_{ki} D_i) z_{ik} + \sum_{i \in N} \sum_{j \in N} \sum_{k \in N} \sum_{\substack{m \in N \\ m \neq k}} \alpha W_{ij} c_{km} X_{ijkm}$$

$$\text{s.t.} \quad \sum_{k \in N} z_{ik} = 1 \qquad \forall i \in N \tag{1}$$

$$\sum_{k \in N} z_{kk} = p \tag{2}$$

$$\sum_{k \in N} \sum_{m > k} y_{km} = p \tag{3}$$

$$\sum_{k < m} y_{km} + \sum_{k > m} y_{mk} = 2 z_{kk} \qquad k \in N \tag{4}$$

$$\sum_{\substack{m \in N \\ m \neq k}} X_{ijkm} + z_{jk} - \sum_{\substack{m \in N \\ m \neq k}} X_{ijmk} - z_{ik} = 0, \quad \forall i,j,k, i \neq j, \ k \neq j \tag{5}$$

$$X_{ijkm} + X_{ijmk} \leq y_{km} \qquad \forall i,j,k \in N, \forall m > k \tag{6}$$

$$X_{ijkm} \geq 0 \qquad \forall i,j,k,m \in N, \ k \neq m \tag{7}$$

$$z_{ik} \in \{0,1\} \qquad \forall i,k \in N \tag{8}$$

$$y_{km} \in \{0,1\} \qquad \forall k \in N, \forall m > k. \tag{9}$$

The first and second terms of the objective function represent the transportation cost between access arcs and hub arcs, respectively. Constraints (1) ensure that each node is allocated to one hub. Constraint (2) is a cardinality constraint on the number of hubs that can be opened, whereas constraint (3) state that the number of hub arcs required to define the cycle is equal to p. Constraints (4) guaranties that each hub node must be connected to exactly to other hub nodes. Constraints (5) are the well-known flow conservation constraints used to model O/D paths. Constraints (6) ensure that paths between origins and destinations will use open hub arcs. Finally, constraints (7)–(9) are the integrality contraints. The combination of constraints (1)–(9) will create paths between all pair of nodes and will form a cycle-star topology with a single connected component. As a consequence, classical subtour elimination constraints, commonly used to model cycles, are not necessary to describe the set of feasible solutions to the CHLP.

3.1.2 Flow-Based Formulation

In order to keep track of the path that is used to send the flow between O/D nodes, we use flow variables commonly used in the hub location literature (see, for instance S. A. Alumur, Nickel, Saldanha-da Gama, & Seçerdin, 2016; Contreras et al., 2010; Ernst & Krishnamoorthy, 1998b). For each $i \in N$ and $(k, m) \in A$, we define x_{ikm} equal to the amount of flow with origin in node $i \in N$ that traverses hub arc (k, m). For each $k, m \in N$, $k < m$, we introduce binary hub arc variables y_{km} equal to one if and only if hub arc (k, m) is selected. We also use the z_{ik} and y_{ik} binary variables for the location allocation and network design decisions. Following Contreras and Fernández (2012), the CHLP can be formulated as the following mixed integer program:

$$
\begin{aligned}
(FF)\ \min \quad & \sum_{i\in N}\sum_{k\in N}(c_{ik}O_i + c_{ki}D_i)z_{ik} + \sum_{i\in N}\sum_{k\in N}\sum_{\substack{m\in N\\ m\neq k}} \alpha c_{km}x_{ikm} \\
\text{s.t.} \quad & \sum_{k\in N} z_{ik} = 1 \qquad i \in N && (10)\\
& \sum_{k\in N} z_{kk} = p && (11)\\
& \sum_{k\in N}\sum_{m\in N} y_{km} = p && (12)\\
& \sum_{m>k} y_{km} + \sum_{k>m} y_{mk} = 2z_{kk} \qquad k \in N && (13)\\
& O_i z_{ik} + \sum_{\substack{m\in N\\ m\neq k}} x_{imk} = \sum_{\substack{m\in N\\ m\neq k}} x_{ikm} + \sum_{m\in N} W_{im}z_{mk} \quad i,k \in N; k \neq i && (14)\\
& z_{km} + y_{km} \leq z_{mm} \qquad k,m \in N; m > k && (15)\\
& z_{mk} + y_{km} \leq z_{kk} \qquad k,m \in N; m > k && (16)\\
& x_{ikm} + x_{imk} \leq O_i y_{km} \qquad i,k,m \in N; m > k && (17)\\
& x_{ikm} \geq 0 \qquad i,k,m \in N && (18)\\
& z_{km}, y_{km} \in \{0,1\} \qquad k,m \in N && (19)
\end{aligned}
$$

The first term of the objective function represents the flow cost on the access arcs whereas the second term evaluates the reduced flow cost on the hub arcs. Constraints (10) ensure that each node is assigned to exactly one hub. Constraint (11) is a cardinality constraint on the number of hubs that must be opened, whereas constraint (12) state that the number of hub arcs in the cycle is equal to p. Constraints (13) guarantee that each hub node must be connected to exactly two other hub nodes. Constraints (14) are the flow conservation constraints. Constraints (15) and (16) ensure that both end nodes of a hub arc are opened hubs and also, they ensure that non-hub nodes are assigned to an open hub.

Constraints (17) state that the flow between hubs moves through the hub cycle. Finally, constraints (18) and (19) are the standard nonnegativity and integrality constraints. As mentioned in Contreras and Fernández (2012), the assumption that the graph of flows G_F contains a single connected component, together with constraints (10)–(10), eliminates the need for subtour elimination constraints.

3.2 An Exact Algorithm

In this section we present an exact branch-and-cut algorithm that uses the linear programming (LP) relaxation of formulation IP as a lower bounding procedure at nodes of the enumeration tree. The LP bounds from the formulation are strengthened with the incorporation of two families of valid inequalities that exploit the structure of the CHLP.

3.2.1 Valid Inequalities

The first set of inequalities is an adaptation of the so-called *mixed-dicut inequalities* first introduced by Ortega and Wolsey (2003) for the fixed-charge, single commodity, network flow problem and later extended to the multi-commodity case for the tree of hubs location problem by Contreras et al. (2010). Let Z denote the set of feasible integer solutions of (10)–(19). The mixed-dicut inequalities can be defined as follows.

Proposition 1 *For $i, m \in N$, $F \subseteq N \setminus \{m\}$, $J \subseteq N \setminus \{i, m\}$, and $Q = \sum_{j \in J \cup \{m\}} W_{ij}$, the inequality*

$$\sum_{k \in N \setminus (F \cup \{m\})} x_{ikm} + Q \left(\sum_{\substack{k \in F \\ k < m}} y_{km} + \sum_{\substack{k \in F \\ k > m}} y_{mk} \right) \geq \sum_{j \in J \cup \{m\}} W_{ij} \left(z_{jm} - z_{im} \right) \tag{20}$$

is valid for Z.

Constraints (20) state that if the O/D nodes in set J are assigned to hub m and node i is not assigned to m, then the amount of flow entering on m via the hub arcs incident to m has to be greater or equal to the sum of the flows with origin in i and with destination in $J \cup \{m\}$, i.e. $\sum_{j \in J \cup \{m\}} W_{ij}$. Given that in any feasible solution to the CHLP the flow originated at i and with destination m and any nodes allocated to m will be routed using the shortest path between these nodes, the flow will enter m via another hub $\hat{k}$ (possibly node i) using hub arc $(\hat{k}, m)$. Thus, depending on the set F, the flow will be counted either using the flow variables of the first term of the left-hand-side of (20) or using the design variables of the second term.

We can generalize the mixed-dicut inequalities (20) by considering now a set of candidate hub nodes $M \subseteq N$ and the set of O/D nodes assigned to them as follows. Let $\delta^-(M) = \{(i,j) \in A : i \in N \setminus M, j \in M\}$ denote the set of arcs entering the set M.

Proposition 2 *For $i \in N$, $M \subseteq N \backslash \{i\}$, $J \subseteq N \backslash (M \cup \{i\})$, $F \subseteq \delta^-(M)$, and $Q_0 = \sum_{j \in (J \cup M)} W_{ij}$, the generalized mixed-dicut inequality*

$$\sum_{(k,m) \in \delta^-(M) \backslash F} x_{ikm} + Q_0 \left(\sum_{\substack{(k,m) \in F \\ k > m}} y_{mk} + \sum_{\substack{(k,m) \in F \\ k < m}} y_{km} \right) \geq \sum_{j \in (J \cup M)} W_{ij} \left(\sum_{m \in M} z_{jm} - \sum_{m \in M} z_{im} \right) \quad (21)$$

is valid for Z.

Proof:

Observe that when $m \in M$ are open hubs and node i is not allocated to any node in M, the right-hand-side $\sum_{j \in (J \cup M)} W_{ij} (\sum_{m \in M} z_{jm} - \sum_{m \in M} z_{im})$, denotes the total flow coming from i and destined to either the hub nodes $m \in M$ or the non-hub nodes $j \in J$ assigned to some hub $m \in M$. The right-hand-side of (21) is thus a lower bound on the total flow arriving to the set of hub nodes M from i. Note that this right-hand-side can only be non-negative when there is one or more nodes $m \in M$ which are open hubs and i is not assigned to any of them, otherwise the right-hand-side would be less than or equal to zero. In the case of the left-hand-side, we note that in any feasible solution in which node i is not allocated to a hub $m \in M$, any amount of flow routed from i to nodes $m \in M$ will arrive via a subset of hub arcs in the cut $\delta^-(M)$. If at least one open hub arc is in F, then the second term of the left-hand-side provides an upper bound on the total amount of flow originated at i with destination $M \cup J$. If all hub arcs in F are closed, then the first term of the left-hand-side provides an upper bound on the total amount of flow originated at i with destination $M \cup J$ entering via a subset of open hub arcs in $\delta^-(M) \setminus F$ and the result follows.

3.2.2 Separating Mixed-dicut Inequalities

Given a fractional solution $(\bar{x}, \bar{y}, \bar{z})$ of the LP relaxation of formulation (10)-(19), the separation problem of inequalities (20) and (21) must be solved to determine whether there exist a violated inequality at $(\bar{x}, \bar{y}, \bar{z})$.

In the case of (20), for each pair of nodes $i, m \in N$, we want to find sets F and J such that

$$\sum_{k \in N \setminus (F \cup \{m\})} \bar{x}_{ikm} + Q \left(\sum_{\substack{k \in F \\ k < m}} \bar{y}_{km} + \sum_{\substack{k \in F \\ k > m}} \bar{y}_{mk} \right) - \sum_{j \in J \cup \{m\}} W_{ij} \left(\bar{z}_{jm} - \bar{z}_{im} \right) < 0.$$

Contreras et al. (2010) present an exact algorithm for solving the separation problem of constraints (20) for the tree of hubs location problem. Given that for each $i, m \in N$, the proposed algorithm requires the solution of several 2-dimensional knapsack problems, the optimal solution of the separation problem requires a considerable amount of time, especially for large-scale instances. Therefore, we next present a fast heuristic to approximately solve the separation problem so as to find violated inequalities (20).

Note that the set $J \subseteq N \setminus \{i, m\}$ affects both the left-and right-hand-side of the inequality, whereas the set $F \subseteq N \setminus \{m\}$ affects only the left-hand-side. Moreover, given a set J and its associated $Q = \sum_{j \in J \cup \{m\}} W_{ij}$, we can efficiently select the set F that minimizes the value of

$$L(Q) = \min_{F \subseteq N \setminus \{m\}} \sum_{k \in N \setminus (F \cup \{m\})} \bar{x}_{ikm} + Q \left(\sum_{k \in F : k < m} \bar{y}_{km} + \sum_{k \in F : k > m} \bar{y}_{mk} \right),$$

using the following result.

Proposition 3 *(Contreras et al., 2010) Let $i, m \in N$, $Q \geq 0$, and $(\bar{x}, \bar{y}, \bar{z})$ be given. Then, a set $\overline{F} \subseteq N \setminus \{m\}$ that minimizes the value of $L(Q)$ is given by $\overline{F} = F_< \cup F_>$, where*

$$F_< = \{k \in N : k < m \textit{ and } \frac{\bar{x}_{ikm}}{\bar{y}_{km}} \geq Q\},$$

and

$$F_> = \{k \in N : k > m \textit{ and } \frac{\bar{x}_{ikm}}{\bar{y}_{mk}} \geq Q\}.$$

The proposed heuristic works by iteratively evaluating different subsets $J \subseteq N \setminus \{i, m\}$ and computing $L(Q)$ to check whether the associated inequality is violated or not. First of all, it constructs an initial set J_0 by considering all $j \in N$ such that $(\bar{z}_{jm} - \bar{z}_{im}) > 0$. Then, it modifies the set J_0 by removing elements from it (one at a time) and evaluating the magnitude of the (possible) violation of the inequality. Let δ denote the smallest difference between the left-hand-side and right-hand-side of the constraint. If the output of the algorithm gives $\delta < 0$, it means that a violated inequality has been found. The steps of the procedure are outlined in Algorithm 1.

In the case of inequalities (21), for each $i \in N$, we want to find sets M, J and F such

Algorithm 1: Separation of inequalities (20) for (i, m)

$J \leftarrow \emptyset$
for $(j \in N)$ **do**
  if $(\bar{z}_{jm} - \bar{z}_{im} > 0)$ **then**
    $J \leftarrow J \cup \{j\}$
  end if
end for
$\delta \leftarrow L(Q) - \sum_{j \in J \cup \{m\}} W_{ij}(\bar{z}_{jm} - \bar{z}_{im})$
for $(l \in J)$ **do**
  $J \leftarrow J \setminus \{l\}$
  if $\left(\delta > L(Q) - \sum_{j \in J \cup \{m\}} W_{ij}(\bar{z}_{jm} - \bar{z}_{im})\right)$ **then**
    $\delta \leftarrow L(Q) - \sum_{j \in J \cup \{m\}} W_{ij}(\bar{z}_{jm} - \bar{z}_{im})$
  else
    $J \leftarrow J \cup \{l\}$
  end if
end for

that

$$\sum_{(k,m) \in \delta^-(M) \setminus F} \bar{x}_{ikm} + Q_0 \left(\sum_{\substack{(k,m) \in F \\ k > m}} \bar{y}_{mk} + \sum_{\substack{(k,m) \in F \\ k < m}} \bar{y}_{km} \right) - \sum_{j \in (J \cup M)} W_{ij} \left(\sum_{m \in M} \bar{z}_{jm} - \sum_{m \in M} \bar{z}_{im} \right) < 0.$$

Observe that, sets $M \subseteq N \setminus \{i\}$ and $J \subseteq N \setminus M \cup \{i\}$ affect both the left and right-hand-sides, whereas set $F \subseteq \delta^-(M)$ only affects the left-hand-side. Therefore, for given sets M and J, we can efficiently select the set F that minimizes the value of

$$R(Q_0) = \min_{F \subseteq \delta^-(M)} \sum_{(k,m) \in \delta^-(M) \setminus F} \bar{x}_{ikm} + Q_0 (\sum_{\substack{(k,m) \in F \\ k > m}} \bar{y}_{mk} + \sum_{\substack{(k,m) \in F \\ k < m}} \bar{y}_{km})$$

using a similar approach as in the case of constraints (20).

Proposition 4 *Let $i \in N$, $Q_0 \geq 0$, and $(\bar{x}, \bar{y}, \bar{z})$ be a given LP solution. Then, a set $\overline{F} \subseteq \delta^-(M)$ that minimizes the value of $R(Q_0)$ is given by $\overline{F} = F_< \cup F_>$, where*

$$F_< = \{(k,m) \in \delta^-(M) : k < m \textit{ and } \frac{\bar{x}_{ikm}}{\bar{y}_{km}} \geq Q_0\},$$

and

$$F_> = \{(m,k) \in \delta^-(M) : k > m \textit{ and } \frac{\bar{x}_{ikm}}{\bar{y}_{mk}} \geq Q_0\}.$$

The proposed heuristic uses an iterative procedure to construct different subsets of $M \subseteq N \backslash \{i\}$ and $J \in N \backslash \{i, m\}$ and computes the associated $R(Q_0)$. We first order the variables $\bar{z}_{kk}$ by non-increasingly and denote k_r the r-th element according to that ordering. That is, $\bar{z}_{k_1 k_1} \geq \bar{z}_{k_2 k_2} \geq \cdots \geq \bar{z}_{k_n k_n}$. We then construct the set M by adding one element at a time with respect to this ordering. Every time a new element is added to M, an associated set J_0 is constructed by considering all $j \in N$ such that $(\sum_{m \in M} \bar{z}_{jm} - \sum_{m \in M} \bar{z}_{im}) > 0$, and $R(Q_0)$ is computed to check whether the associated inequality is violated or not. If the violation obtained from the addition of the new element to M is higher that the violation at the previous iteration, the element is permanently added to M. Otherwise, it is removed and the next element in the sequence is selected as candidate. Once all candidates with $\bar{z}_{k_r k_r} > 0$ are considered, the algorithm tries to modify J_0 by removing elements from it one at a time and evaluating the corresponding δ. The outline of the procedure is depicted in Algorithm 2.

3.2.3 A Branch-and-Cut Algorithm

We present an exact branch-and-cut method for solving the CHLP. The idea is to solve the LP relaxation of IP with a cutting-plane algorithm by initially including only constraints (10)–(16) and (18)–(19) at the root node and iteratively adding constraints (17), (20), and (21) only when violated by the current LP solution. When no more violated

Algorithm 2: Separation of inequalities (21) for (i)

$M \leftarrow \emptyset$, $\delta_{min} \leftarrow 0$, $r \leftarrow 1$
Sort the values $\bar{z}_{kk}$ non-increasingly
while$(\bar{z}_{k_r k_r} > 0)$ **do**
    $J \leftarrow \emptyset$
    $M \leftarrow M \cup \{k_r\}$
    for $(j \in N)$ **do**
        if $(\sum_{m \in M} \bar{z}_{jm} - \sum_{m \in M} \bar{z}_{im} > 0)$ **then**
            $J \leftarrow J \cup \{j\}$
        end if
    end for
    $\delta \leftarrow R(Q_0) - \sum\limits_{j \in J \cup \{m\}} W_{ij}(\sum_{m \in M} \bar{z}_{jm} - \sum_{m \in M} \bar{z}_{im})$
    if $(\delta < \delta_{min})$ **then**
        $\delta_{min} \leftarrow \delta$
    else
        $M \leftarrow M \setminus \{k_r\}$
    end if
    $r \leftarrow r + 1$
end while
for $(l \in J)$ **do**
    $J \leftarrow J \setminus \{l\}$
    $\delta \leftarrow R(Q_0) - \sum\limits_{j \in J \cup \{m\}} W_{ij}(\sum_{m \in M} \bar{z}_{jm} - \sum_{m \in M} \bar{z}_{im})$
    if $(\delta < \delta_{min})$ **then**
        $\delta_{min} \leftarrow \delta$
    else
        $J \leftarrow J \cup \{l\}$
    end if
end for

inequalities are found, we resort to CPLEX for solving the resulting formulation by enumeration, using a call-back function for generating additional violated constraints (17), (20) and (21) at the nodes of the enumeration tree. The separation problem of inequalities (17) is solved by inspection at every node of the tree. The separation of inequalities (20) is carried out using Algorithm 1 at the root node and at every other node for which the depth is multiple of 25. The separation problem of inequalities (21) is carried out using Algorithm 2 and only at the root node of the enumeration tree. For constraints (20) and (21), we limit the number of generated cuts at every iteration of the separation phase by using a threshold value ϵ for the minimum violation required for a cut to be added. We use a branching strategy in which the highest priority is given to the location variables (z_{kk}), followed by the hub arc variables (y), and least priority to the allocation variables (z_{ik}).

3.3 A Heuristic Algorithm

In this section we propose a GRASP for the CHLP. GRASP is a multi-start metaheuristic used for solving combinatorial optimization problems (Festa & Resende, 2011). Each iteration consists of two phases: a constructive phase and a local search phase.

For the CHLP we propose a constructive phase with three steps. In the first step, a set of p hubs is randomly selected among a set of candidate nodes. The remaining nodes are then assigned to their closest open hub. Finally, a set of p hub arcs, associated with the selected hub nodes, are then chosen in such a way that they form a cycle on the backbone network. A local search phase is later used to improve the initial solution. In particular, a variable neighborhood descent (VND) method is used to systematically explore a set of neighborhoods that modify the structure of the network.

In what follows, solutions are represented by a set of hub nodes H, a set of hub arcs E, and an assignment mapping a. Therefore, solutions are designated by the form $S = (H, E, a)$,where $H \in N$ denotes the set of selected sites to locate hubs, i.e., $H(i) = 1$ if node $i \in N$ is selected to be a hub, $E : e \to R$ represents the set of arcs between hub nodes, i.e., $E(e) = 1$ if hub arc e exists, and $a : N \to H$ is the assigning mapping, i.e., $a(j) = m$ if node $j \in N$ is assigned to hub node $m \in H$.

Constructive Phase

Let $S = (H, E, a)$ be a partial solution where $H(i) = null$, $E(e) = null$ and $a(j) = null$. To generate a feasible solution, three steps are considered; the selection of a set of hubs, the assignment of nodes to hubs and the connection of hubs so as to construct a cycle structure. A restricted candidate list (RCL) is built using a greedy function, where, at each iteration t a sub-region $N_i^t(r) = \{j \in N^t : d_{ij} \leq r\}$ of candidate nodes N^t around a node i with a radius of size r is considered. We define the greedy function as

$$\psi_i^1 = \sum_{j \in N_i^t(r)} (W_{ij} + W_{ji}),$$

and

$$\psi_i^t = \sum_{j \in N_i^t(r)} (W_{ij} + W_{ji}) + \sum_{j \in N_i^t(r)} \sum_{k \in \{1,\ldots,t-1\}} \sum_{m \in N_{i(k)}^k(r)} (W_{jm} + W_{mj}),$$

for $t > 1$, where $i(k)$ denotes the node selected as a hub at iteration k. The first term of ψ_i^t represents the flow originated at node i with destination $N_i^t(r)$, and the total flow going into node i coming from nodes in $N_i^t(r)$. That is, node i is considered as a potential hub to serve nodes $j \in N_i^t(r)$. The second term of ψ_i^t represents the amount of flow that needs to be routed between nodes inside the sub-region $N_i^t(r)$ of a candidate hub node i and the nodes inside the sub-regions $N_{i(k)}^k(r)$ of the open hubs $i(k)$ selected in previous iterations $k = 1, \ldots, t-1$.

In order to achieve a trade-off between quality and diversity, a partially randomized greedy procedure is considered. At each iteration, one element is randomly selected from the RCL to become a hub. The RCL is updated at each iteration of the construction phase and contains the best candidate nodes N^t with respect to the greedy **function.** Let $\psi_{min}^t = \min\{\psi_i^t : i \in N^t\}$ and $\psi_{max}^t = \max\{\psi_i^t : i \in N^t\}$, then

$$RCL = \{i : \psi_i^t \leq \psi_{min}^t + \beta \left(\psi_{max}^t - \psi_{min}^t\right)\},$$

where $0 \leq \beta \leq 1$. Once a hub is located at a candidate node i, we remove all nodes in $N_i^t(r)$ from the set of candidate nodes N^{t+1}. When p hubs are opened, all the non-hub nodes are simply assigned to their closest opened hub. In order construct a feasible solution, a *nearest neighbor algorithm* (see, Cook, Cunningham, Pulleybank, & Schrijver, 1998) is applied to connect the set of selected hubs by means of a cycle. It works by arbitrarily selecting a hub node and connecting it to the nearest hub not yet connected. The process continues until all hubs are connected. The constructive phase is outlined in Algorithm 3.

Algorithm 3: Constructive Phase of GRASP

$H \leftarrow \phi, t \leftarrow 1, N^0 \leftarrow N$
while $(\mid H \mid \neq p)$ **do**
  Evaluate ψ_i^t for all $i \in N^t$
  $RCL = \{i : \psi_i^t \leq \psi_{min}^t + \beta\,(\psi_{max}^t - \psi_{min}^t)\}$
  Select randomly $i^* \in RCL$
  $H \leftarrow H \cup i^*$
  $N^t \leftarrow N^{t-1} \backslash \{i^* \cup N_{i^*}^t(r)\}$
  $t \leftarrow t + 1$
end while
Assign each node $j \in N^{t-1}$ to its closest hub in H.
Connect hubs using the Nearest Neighbor Algorithm.

Local Search Phase

The local search phase is used to improve the initial solution obtained from the constructive phase. We use a local search procedure based on a VND method for the CHLP. The VND was initially proposed by Brimberg and Mladenovic (1996) and is based on a systematic search in a set of k neighborhoods, $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_k$. The VND works by performing a local search in a neighborhood $\mathcal{N}_1$ until a local optimal solution is found. After that, the algorithm switches to neighborhoods $\mathcal{N}_2, \ldots, \mathcal{N}_k$, sequentially, until an improved solution is found. Each time the search improves the best known solution, the procedure restarts using the neighborhood $\mathcal{N}_1$. Our implementation of the VND algorithm explores three types of neighborhood structures. The first type consist of the classical shift and swap neighborhood. The latter one tries to improve the current solution by swapping the assignment of two nodes, whereas the former one considers all solutions that differ from the current one by changing the assignment of one node. Let $S = (H, E, a)$ be the current solution, then

$$\mathcal{N}_1(s) = \{s^{'} = (H, E, a^{'}) : \exists! j \in N,\ a^{'}(j) \neq a(j)\},$$

and

$$\mathcal{N}_2(s) = \{s^{'} = (H, E, a^{'}) : \exists!(j_1, j_2),\ j_1^{'} = a(j_2),\ j_2^{'} = a(j_1),\ \forall j \neq j_1, j_2\}.$$

To explore $\mathcal{N}_1(s)$, all pairs of the form (i, j) are considered, where $a(j) \neq i$ and for $\mathcal{N}_2(s)$ all pairs of the form (j_1, j_2) are considered, where $a(j_1) = a(j_2)$. In both cases we use a best improvement strategy.

The second type of neighborhood structure affects the current set of open hubs. Let $S = (H, E, a)$ be the current solution and let $i \in N \backslash H$ be the nodes which are candidate to replace the open hubs located at site $m \in H$, then

$$\mathcal{N}_3(s) = \{S^{'} = (H^{'}, E^{'}, a^{'}) : S^{'} = H^{'} \backslash \{m\} \cup \{i\}, m \in S^{'},\ i \in N \backslash H\}.$$

To explore $\mathcal{N}_3(s)$ all nodes $m \in N \backslash H$ are considered, and a set of solutions is obtained from the current one by interchanging an open hub by a closed one and reassigning all the non-hub nodes to their closest open hub.

The third type of neighborhood structure is the so-called *2-opt* (Cook et al., 1998), commonly used in other optimization problems in which cycle structures are sought. The procedure works by deleting two hub arcs and reconnecting the network with a new cycle. Let $S = (H, E, a)$ be the current solution, then

$$\mathcal{N}_4(s) = \{S = (H, E^{'}, a) : E^{'} = E \backslash \{(i_1, j_1), (i_2, j_2)\} \cup \{(i_1, i_2), (j_1, j_2)\}\}.$$

In this neighborhood, a best improvement strategy is also considered.

3.4 Results

We conduct computational experiments to analyze and compare the performance of the formulations presented in this Chapter using the commercial solver CPLEX as well as the proposed solution approaches—the exact branch-and-cut algorithm and the GRASP meta-heuristic. The formulations and solution algorithms were coded in C and run on a single possessor of 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 integer programs were solved using the callback library of CPLEX 12.4. The numerical tests were performed using the Australian Post (AP) instances obtained from the OR library *(http://mscmga.ms.ic.ac.uk/jeb/orlib/phubinfo.html)*. These instances comprise of postal flow and Euclidean distances between 200 postal districts in an Australian city. In our experiments, we have selected instances with $|N| =$ 10, 20, 25, 40, 50, 60, 70, 75, 90, and 100 nodes. The number of hubs to be opened was set to $p =$ 4, 6 and 8, and the value of discount factor was varied from $\alpha =$0.2, 0.5 to 0.8.

Our preliminary computational results focus on the comparison between the path-based (PB) formulation and flow-based (FB) formulation when solved with CPLEX. The detailed results of this comparison on a set of instances ranging from 10 to 40 nodes are given in Table 1. The first column lists the problem parameters such as the number of nodes $|N|$, the number of hubs to be opened p and the discount factor α for each instance.. The next columnns report the linear programming relaxation gap $(\%LP)$, the percent deviation between final upper and lower bound $(\%Gap)$, the CPU time (CPU) in seconds, and the number of explored nodes in the branching tree $(Nodes)$, for both formulations. The $\%LP$ Gap is computed as $(UB - LP)/(UB) \times 100\%$, where UB denotes the best upper bound (or optimal solution value) obtained with CPLEX and LP is the optimal value of the LP relaxation. The final gap $(\%Gap)$ is computed as $(UB - LB)/(UB) \times 100\%$, where UB and LB denote the best upper and lower bounds obtained by CPLEX at termination, respectively. Throughout this experiment, the maximum time limit is set to 14,400 seconds of CPU time on CPLEX. Whenever CPLEX cannot optimally solve an instance within the time limit, we write $time$.

Instance			*Path-Based Formulation(PF)*				*Flow-Based Formulation(FF)*			
$\|N\|$	p	$alpha$	% LP	% Gap	CPU	Nodes	% LP	%Gap	CPU	Nodes
10	4	0.2	2.25	0.0	1.5	4	3.37	0.0	2.08	25
10	4	0.5	0.00	0.0	0.48	0	5.34	0.0	0.67	43
10	4	0.8	0.00	0.0	0.5	0	6.93	0.0	0.97	16
10	6	0.2	0.00	0.0	51	0	5.62	0.0	1.14	73
10	6	0.5	0.00	0.0	0.52	0	8.58	0.0	1.26	261
10	6	0.8	0.00	0.0	0.52	0	10.62	0.0	1.54	877
20	4	0.2	1.05	0.0	890.13	9	1.70	0.0	3.78	36
20	4	0.5	0.29	0.0	470.92	2	4.33	0.0	17.69	485
20	4	0.8	0.00	0.0	346.68	0	5.11	0.0	29.16	1024
20	6	0.2	0.78	0.0	511.5	4	5.60	0.0	30.48	753
20	6	0.5	0.00	0.0	320.8	0	8.26	0.0	350.89	9693
20	6	0.8	0.60	0.0	614.46	8	9.68	0.0	1127.92	32563
20	8	0.2	1.37	0.0	753.16	15	7.35	0.0	180.73	4837
20	8	0.5	1.93	0.0	2159.55	56	12.84	0.0	2736.87	58631
20	8	0.8	1.17	0.0	883.06	20	12.66	0.0	8516.44	245771
25	4	0.2	4.89	4.0	time	6	1.79	0.0	13.65	66
25	4	0.5	0.05	0.0	2661.86	0	3.15	0.0	46.69	248
25	4	0.8	0.00	0.0	707.68	0	4.50	0.0	118.65	928
25	6	0.2	0.00	0.0	2658.85	0	3.46	0.0	32.26	312
25	6	0.5	0.00	0.0	1289.08	0	6.35	0.0	327.76	3435
25	6	0.8	0.25	0.0	7554.94	3	8.87	0.0	4065.04	37955
25	8	0.2	1.08	1.1	time	32	7.51	0.0	3169.96	22421
25	8	0.5	0.45	0.0	7285.53	12	10.12	0.0	7711.25	68475
25	8	0.8	0.98	0.0	7393.37	50	11.09	1.1	time	110994
40	4	0.2	time	time	time	time	1.65	0.0	71.232	127
40	4	0.5	time	time	time	time	3.40	0.0	3445.22	1816
40	4	0.8	time	time	time	time	5.29	1.7	time	7195
40	6	0.2	time	time	time	time	4.10	0.0	4119.21	3332
40	6	0.5	time	time	time	time	8.56	7.0	time	4056
40	6	0.8	time	time	time	time	9.21	7.5	time	4929
40	8	0.2	time	time	time	time	6.54	4.6	time	3217
40	8	0.5	time	time	time	time	12.42	11.2	time	4600
40	8	0.8	time	time	time	time	14.82	14.1	time	5093

Table 3.1: Comparison between path-based and flow-based formulation.

In Table 1, we can observe that formulation PF is able to solve 22 out of the 33 considered instances to optimality within the time limit. The % LP gap for the instances that were solved is relatively small, as is always less than 2.25% and in 11 instances it is equal to zero. However, CPLEX is not able to solve the LP relaxation of all 40-node instances in four hours of CPU time. In the case of formulation FF, CPLEX is able to solve 26 out of the 33 instances to optimality. As expected, the % LP gap for the instances that were solved is always larger than the one of the PF. Nevertheless, given that there is a considerable smaller number of variables and constraints in FF, it is able to solve three 40-node instances and one 25-node instance that the PF cannot solve.

Now, we focus on analyzing the improvement of the linear programming (LP) relaxation bounds obtained when adding the cuts automatically generated by CPLEX and the two families of valid inequalities (20) and (21) introduced in Section 3.2 for the formulation FB. In particular, we compare the results of the following experiments:

In the first part of the computational experiments we focus on analyzing the improvement of the linear programming (LP) relaxation bounds obtained when adding the cuts automatically generated by CPLEX and the two families of valid inequalities (20) and (21) introduced in Section 3.2 for the formulation IP. In particular, we compare the results of the following experiments:

(1) We solve the LP relaxation of formulation IP and we do not allow CPLEX to add cuts.

(2) We solve the LP relaxation of formulation IP and we allow CPLEX to add cuts to improve the initial LP bounds. All the cuts parameters are set to their default settings.

(3) We solve the LP relaxation of formulation IP and we dynamically add the mixed-dicut inequalities (20) using the separation heuristic presented in Section 3.2 to find violated inequalities. We set $\epsilon = 0.001$ for the minimum violation required for a cut

to be added.

(4) We solve the LP relaxation of formulation IP and we dynamically add the generalized mixed-dicut inequalities (21) using the separation heuristic presented in Section 3.2 to find violated inequalities. We set $\epsilon = 0.01$ for the minimum violation required for a cut to be added.

The detailed results of these experiments are shown in Table 1. The first column lists the problem parameters such as the number of nodes $|N|$, the number of hubs to be opened p and the discount factor α for each instance. The second set of columns under the heading ***CPLEX*** reports the LP gap ($\%LP$), the LP gap after adding CPLEX cuts ($\%LP_{cuts}$), the number of cuts added by CPLEX ($\#cuts$), and the CPU time in seconds (CPU) to solve the LP and to add the cuts. The $\%LP$ gap is computed as $(UB-LP)/(UB)\times 100\%$, where UB denotes the best upper bound (or optimal solution value) and LP is the optimal value of the LP relaxation. The third set of columns under the heading ***MDI*** shows the results when adding the mixed-dicut inequalities (20) to IP. The results include the LP gap (*%LP*) after adding inequalities (20), the number of violated cuts added *(Cuts)*, and the CPU time. The last set of columns reports the LP gap (*%LP*) after adding the generalized mixed-dicut inequalities (21), the number of violated inequalities *(Cuts)*, and the CPU time. In all cases, the CPU time include the time for separating and adding violated inequalities.

Results in Table 1 show that the average percent LP gap of formulation IP is 6.64% and ranges from 1.15% to 12.84%. With the addition of CPLEX cuts, the LP gap is reduced to an average of 5.36% and ranges from 0.48% to 10.58%. However, when adding the mixed-dicut inequalities (20) the average LP gap is further reduced to 2.82% with a range from 0.03% to 7.14%. When adding the generalized mixed-dicut inequalities (21), the average LP gap is 1.94% with a range from 0.00% to 5.86%. In fact, constraints (21) are able to close the optimality gap, and obtain an integer optimal solution in 3 out of the 6 instances with 10 nodes. However, given that the number of cuts added is much larger to the number

Table 3.2: Comparison between CPLEX cuts and mixed-dicut inequalities

Instance	$CPLEX$				*MDI*			*GMDI*		
$\|N\|$-P-α	% LP	$\%LP_{cuts}$	$Cuts$	CPU	% LP	$Cuts$	CPU	% LP	$Cuts$	CPU
10-4-0.2	3.37	1.96	84	0.13	0.92	182	0.32	0.60	245	0.12
10-4-0.5	5.34	2.18	75	0.13	0.68	178	0.42	0.07	232	0.09
10-4-0.8	6.93	3.05	74	0.10	0.64	225	0.42	0.00	241	0.08
10-6-0.2	5.62	2.64	69	0.08	0.91	195	0.33	0.00	265	0.08
10-6-0.5	8.58	4.50	88	0.12	1.54	289	0.46	0.00	292	0.11
10-6-0.8	10.62	7.21	71	0.11	2.78	294	0.47	0.09	467	0.24
20-4-0.2	1.70	0.48	198	1.06	0.03	393	0.69	0.10	432	1.21
20-4-0.5	4.33	3.32	288	1.66	1.48	1119	2.71	1.28	1449	9.28
20-4-0.8	5.11	3.41	215	1.72	1.47	843	1.86	0.83	1077	4.78
20-6-0.2	5.60	3.62	288	2.14	1.15	1218	5.19	0.72	2104	17.43
20-6-0.5	8.26	6.60	294	1.67	2.80	1643	4.98	1.69	2802	28.56
20-6-0.8	9.68	7.60	300	2.43	4.46	1388	4.60	2.87	2300	18.44
20-8-0.2	7.35	5.98	234	1.73	2.93	1517	4.45	1.98	2070	19.95
20-8-0.5	12.84	10.58	313	3.33	6.48	1868	6.52	4.46	3370	36.71
20-8-0.8	12.66	9.93	329	5.20	6.18	1671	6.39	3.87	3059	24.50
25-4-0.2	1.79	1.35	262	2.45	0.13	1198	2.71	0.26	1052	5.25
25-4-0.5	3.15	2.75	214	2.10	0.42	1327	4.52	0.33	1290	10.34
25-4-0.8	4.50	4.03	249	3.13	1.63	1211	4.83	0.76	1289	13.77
25-6-0.2	3.46	2.44	275	2.37	0.22	1528	6.00	0.15	1721	12.98
25-6-0.5	6.35	5.57	273	2.24	1.80	2059	6.40	1.04	2635	43.74
25-6-0.8	8.87	6.99	392	6.78	4.29	1476	7.40	2.72	2488	38.57
25-8-0.2	7.51	6.31	386	4.58	3.43	2527	14.22	2.82	4001	94.13
25-8-0.5	10.12	8.26	390	5.99	4.71	2279	11.14	3.15	3472	80.77
25-8-0.8	11.09	8.98	378	7.56	5.84	1851	9.69	3.66	3255	57.95
40-4-0.2	1.65	1.55	411	7.43	0.21	2613	25.10	0.44	2179	82.53
40-4-0.5	3.40	3.10	446	17.43	0.94	2922	66.25	0.80	3597	378.98
40-4-0.8	5.29	5.23	398	20.77	2.44	3164	75.53	1.60	3786	484.93
40-6-0.2	4.10	3.44	525	17.94	1.39	4958	87.85	1.47	6180	958.25
40-6-0.5	7.45	7.28	619	22.33	3.88	5619	124.27	3.10	8441	2389.26
40-6-0.8	8.34	8.01	511	27.09	4.80	3989	92.17	3.06	7543	2236.03
40-8-0.2	6.54	5.84	660	17.03	3.38	6369	95.67	2.81	11774	2867.92
40-8-0.5	10.80	9.42	774	46.06	6.44	5039	125.03	4.82	10104	4731.06
40-8-0.8	10.21	8.86	779	48.03	5.96	4211	106.10	3.83	9105	3427.49
50-4-0.2	1.15	1.02	392	23.48	0.08	2250	40.66	0.23	2304	144.24
50-4-0.5	2.57	2.58	610	51.44	0.52	3677	216.2	0.62	4091	1326.32
50-4-0.8	5.02	4.68	504	70.46	2.45	3838	305.00	1.66	4768	2184.42
50-6-0.2	3.56	2.96	631	48.92	1.06	5486	239.69	1.21	7448	3604.78
50-6-0.5	7.92	7.54	850	65.39	4.78	6132	386.22	4.13	11016	11899.96
50-6-0.8	8.20	8.08	756	73.97	5.18	5219	337.82	3.57	12809	18687.13
50-8-0.2	6.41	5.87	867	52.23	3.72	7381	325.78	3.51	12827	11964.63
50-8-0.5	10.76	9.82	1047	94.08	7.04	6984	392.82	5.86	14684	27641.40
50-8-0.8	10.60	10.03	866	96.30	7.14	5640	348.38	5.25	1278	18705.52
Average	6.64	5.36	413.93	20.46	2.82	2713.57	83.27	1.94	4179.57	2719.86

of cuts added by CPLEX, the CPU time to solve the associated LPs substantially increases. We also note that the quality of the obtained LP bounds seem to depend on the size, number of hub facilities and discount factor. For instance, the LP gap is worse as N and p increase. Also, the LP gaps tend to deteriorate as the value of the discount factor α increases. It is interesting to observe that the number of generated cuts also depend on these parameters.

We next compare the impact of separating both families of inequalities (20) and (21) and adding them to formulation IP at the same time. In particular, we compare the results of the following experiments:

5. We solve the LP relaxation of formulation IP and we first dynamically add constraints (20) using the separation heuristic. When no more inequalities of this type can be found, constraints (21) are then added.

6. We solve the LP relaxation of formulation IP and we first dynamically add constraints (21) using the separation heuristic. When no more inequalities of this type can be found, constraints (20) are then added.

The results of these experiments are given in Table 2. For every instance, we report the percent LP gap obtained after adding cuts from both families (*%LP*), the number of added cuts from both families *(Cut1)* and *(Cut2)*, respectively, and the CPU time in seconds (CPU).

Table 2 shows that in the case of $MDI+GMDI$, the average percent LP gap is 1.75% and ranges from 0.00% to 5.56%, whereas in the case of $GMDI + MDI$, the average LP gap is slightly reduced to 1.68% and ranges from 0.00% to 5.43%. In 34 out of 42 instances, $GMDI + MDI$ results in lower LP gap as compared to $MDI + GMDI$. Moreover, by adding the two families of inequalities to formulation IP, we are able to obtain an integer solution from the LP relaxation in 5 instances. In general, combining both families of valid inequalities provides remarkable results in terms of the quality of the LP bounds. Although the $GMDI + MDI$ scheme provides on average the best LP bounds, the required CPU time is considerably larger than the other experiments performed. However, experiments 5 provide the best overall results in terms of a tradeoff between the quality of the LP bounds and the CPU time. Therefore, in the remainder of the experiments we are only considering this scheme.

In the second part of the computational experiments, we analyze the performance of our proposed exact and heuristic solution algorithms. In particular, we compare the quality of the obtained solutions using both constraints (20) and (21) within a branch-and-cut framework and the GRASP algorithm introduced in Section 4. These experiments are performed

Table 3.3: Results when combining mixed-dicut inequalities

Instance	*MDI+GMDI*				*GMDI+MDI*			
$\|N\|$-P-α	% LP	$\#Cut1$	$\#Cut2$	CPU	% LP	$\#Cut1$	$\#Cut2$	CPU
10-4-0.2	0.59	197	77	0.46	0.48	60	253	0.16
10-4-0.5	0.00	178	78	0.18	0.00	6	232	0.08
10-4-0.8	0.00	225	60	0.19	0.00	0	241	0.09
10-6-0.2	0.00	195	54	0.14	0.00	0	265	0.08
10-6-0.5	0.00	289	91	0.24	0.00	0	292	0.10
10-6-0.8	0.27	302	203	0.65	0.01	35	480	0.27
20-4-0.2	0.00	398	14	0.70	0.00	48	433	0.98
20-4-0.5	0.99	1210	484	7.38	0.97	384	1449	11.14
20-4-0.8	0.67	934	481	4.99	0.62	225	1077	5.53
20-6-0.2	0.49	1383	624	16.11	0.45	446	2104	18.27
20-6-0.5	1.54	1821	763	14.85	1.38	472	2802	27.76
20-6-0.8	2.77	1583	1032	13.61	2.67	368	2300	16.88
20-8-0.2	1.99	1582	816	28.15	1.75	443	2070	19.44
20-8-0.5	4.43	2061	1356	26.86	4.27	418	3370	40.07
20-8-0.8	3.86	1834	1598	19.88	3.73	308	3059	29.28
25-4-0.2	0.04	1216	130	3.90	0.02	339	1052	6.63
25-4-0.5	0.05	1393	325	20.38	0.05	296	1290	12.53
25-4-0.8	0.54	1376	576	15.80	0.55	298	1289	16.56
25-6-0.2	0.04	1539	136	7.51	0.01	156	1823	25.18
25-6-0.5	0.66	2209	1042	26.42	0.69	529	2635	49.97
25-6-0.8	2.55	1680	1212	32.84	2.45	460	2488	40.97
25-8-0.2	2.69	2804	1324	37.66	2.50	938	4001	104.47
25-8-0.5	2.98	2490	1846	55.97	2.88	540	3472	74.96
25-8-0.8	3.66	2092	1579	45.96	3.51	382	3255	69.03
40-4-0.2	0.17	2661	120	31.20	0.04	985	2179	101.96
40-4-0.5	0.55	3131	778	155.61	0.46	1025	3597	429.58
40-4-0.8	1.26	3609	1728	370.22	1.27	1061	3786	519.81
40-6-0.2	1.33	5104	194	112.62	1.01	1815	6180	1039.84
40-6-0.5	2.97	6269	2228	692.53	2.69	1719	8441	2576.85
40-6-0.8	2.77	4478	4310	1365.29	2.78	1190	7543	2247.17
40-8-0.2	2.57	6868	2912	704.81	2.52	1654	11774	3087.52
40-8-0.5	4.75	5516	4258	1590.11	4.53	1464	10104	4705.78
40-8-0.8	3.52	4675	5210	1945.96	3.55	1122	9105	3438.78
50-4-0.2	0.03	2315	179	61.12	0.04	533	2304	182.26
50-4-0.5	0.19	3864	889	505.81	0.22	1071	4091	1443.04
50-4-0.8	1.39	4436	1874	1391.38	1.36	1142	4768	2396.32
50-6-0.2	0.80	5878	1403	822.11	0.73	2110	7448	4124.67
50-6-0.5	3.75	6649	4312	4571.24	3.72	2002	11016	12997.43
50-6-0.8	3.31	5783	6134	7365.06	3.30	1481	12809	19314.28
50-8-0.2	3.05	8113	3867	3225.68	3.05	2635	12827	12591.72
50-8-0.5	5.56	7730	6523	8127.63	5.43	2134	14684	29865.80
50-8-0.8	4.88	6182	8712	12347.64	5.00	1331	14882	21670.62
Average	1.75	2958.38	1703.14	1089.69	1.68	800.60	4506.43	2935.81

on the same set of instances as before (ranging from 10 to 50 nodes). Throughout this experiment, we set a time limit to 86,400 seconds of CPU time. Instances that could not be solved to optimality within this time limit are marked with the label "$time$". Moreover, we set $\epsilon = 0.05$ for the minimum violation required for both families of cuts to be added. We also stop adding inequalities at a given node when the improvement of the LP bounds between the previous iteration and the current one is less that 0.08%.

The detailed results are reported in Table 3. The second set of columns reports the LP gap ($\%LP$), the percent deviation between final upper and lower bound ($\%Gap$), the CPU time (CPU) in seconds, and the number of explored nodes in the branching tree $(Nodes)$. Note that, the final gap ($\%Gap$) is computed as $(UB - LB)/(UB) \times 100\%$, where UB and LB denote the best upper and lower bounds obtained at termination, respectively. The third set of columns under heading *Branch-and-Cut* reports: $\%LP_{cuts}$ the LP bound at the root node after adding valid inequalities (20) and (21), (*%Gap*) the final percent deviation at termination, the CPU time (CPU) in seconds, and the number of explored nodes in the branching tree $(Nodes)$.

The fourth set of columns reports the results of the GRASP. In order to assess the quality and robustness of the solution obtained from GRASP, the algorithm was run 30 times for each instance. The best objective value obtained across all the 30 runs is used to compute the best percentage deviation $(\%Dev)$ with respect to the optimal solution value or the best LB bound obtained $(i.e., \%Dev = (best\, solution\, GRASP - LB)/(best\, solution\, GRASP) \times 100\%)$. The robustness is measured by using the average percent deviation $(\%Avg\ Dev)$ using the best solutions obtained in each of the 30 runs. The average CPU time in seconds across all the runs of the GRASP is also reported.

The results in Table 3 show that by using formulation IP and a commercial solver (CPLEX), we were able to solve 31 instances to optimality and the final gaps on the remaining instances range from 0.60% to 10.10%. On the other hand, the branch-and-cut algorithm succeeds in solving 35 out of the 42 instances to optimality within the time limit. For the remaining 7 instances, the final gaps range from 1.50% to 5.50% . The branch-and-cut algorithm is faster than CPLEX on 30 out of 31 instances that were solved to optimality using both the algorithms. Moreover, our branch-and-cut algorithm is able to solve 4 instances that CPLEX is unable to solve within the time limit. For the instances that could not be solved to optimality, the branch-and-cut always provides smaller final percent gaps than CPLEX.

Table 3 also shows that the GRASP algorithm is very effective in finding high quality

Table 3.4: Computational results for the branch-and-cut and GRASP algorithms for small/medium size instances

Instance	*CPLEX*				*Branch-and-Cut*				*GRASP*		
$\|N\|$-P-α	% LP	% Gap	CPU	Nodes	$\%LP_{cut}$	%Gap	CPU	Nodes	% Dev	% Avg	CPU
10-4-0.2	3.37	0.00	2.08	25	0.00	0.00	0.03	0	0.00	0.00	0.03
10-4-0.5	5.34	0.00	0.67	43	0.80	0.00	0.37	8	0.00	0.00	0.03
10-4-0.8	6.93	0.00	0.97	16	1.24	0.00	0.47	15	0.00	0.00	0.03
10-6-0.2	5.62	0.00	1.14	73	3.93	0.00	0.40	64	0.00	0.00	0.04
10-6-0.5	8.58	0.00	1.26	261	4.10	0.00	1.08	204	0.00	0.00	0.04
10-6-0.8	10.62	0.00	1.54	877	5.66	0.00	2.01	668	0.00	0.00	0.04
20-4-0.2	1.70	0.00	3.78	36	0.04	0.00	0.68	3	0.00	0.00	0.21
20-4-0.5	4.33	0.00	17.69	485	1.36	0.00	5.16	54	0.00	0.00	0.21
20-4-0.8	5.11	0.00	29.16	1024	0.91	0.00	3.46	43	0.00	0.00	0.25
20-6-0.2	5.60	0.00	30.48	753	0.83	0.00	8.30	85	0.00	0.00	0.32
20-6-0.5	8.26	0.00	350.89	9693	2.08	0.00	24.15	706	0.00	0.00	0.35
20-6-0.8	9.68	0.00	1127.92	32563	3.09	0.00	159.39	5422	0.00	0.05	0.39
20-8-0.2	7.35	0.00	180.73	4837	2.18	0.00	38.65	1177	0.00	0.00	0.45
20-8-0.5	12.84	0.00	2736.87	58631	4.93	0.00	1569.60	20306	0.00	0.00	0.52
20-8-0.8	12.66	0.00	8516.44	245771	4.36	0.00	2119.75	24271	0.00	0.03	0.51
25-4-0.2	1.79	0.00	13.65	66	0.26	0.00	3.00	19	0.00	0.00	0.42
25-4-0.5	3.15	0.00	46.69	248	0.28	0.00	5.67	16	0.00	0.00	0.49
25-4-0.8	4.50	0.00	118.65	928	0.81	0.00	12.91	50	0.00	0.00	0.50
25-6-0.2	3.46	0.00	32.26	312	0.20	0.00	5.86	22	0.00	0.00	0.66
25-6-0.5	6.35	0.00	327.76	3435	1.20	0.00	20.35	99	0.00	0.00	0.79
25-6-0.8	8.87	0.00	4065.04	37955	2.82	0.00	239.53	2229	0.00	0.00	0.88
25-8-0.2	7.51	0.00	3169.96	22421	3.01	0.00	340.35	2533	0.00	0.00	0.97
25-8-0.5	10.12	0.00	7711.25	68475	3.37	0.00	1127.22	6327	0.00	0.00	1.11
25-8-0.8	11.09	0.00	18229.57	152835	4.04	0.00	5854.66	14780	0.00	0.55	1.11
40-4-0.2	1.65	0.00	71.23	127	0.33	0.00	27.36	32	0.00	0.00	2.16
40-4-0.5	3.40	0.00	3445.22	1816	0.82	0.00	112.30	69.00	0.00	0.00	2.39
40-4-0.8	5.29	0.00	24354.44	16182	1.66	0.00	418.61	343	0.00	0.00	2.62
40-6-0.2	4.10	0.00	4119.21	3332	1.87	0.00	368.63	1200	0.00	0.00	3.67
40-6-0.5	7.45	2.10	86400	41396	3.68	0.00	12862.19	4225	0.00	0.00	3.87
40-6-0.8	8.34	1.40	86400	46600	3.56	0.00	48756.04	14547	0.00	0.00	3.68
40-8-0.2	6.54	0.70	86400	52491	3.77	0.00	17623.38	9436	0.00	0.00	5.45
40-8-0.5	10.80	8.70	time	42625	5.47	3.43	time	6708	3.43	3.43	5.32
40-8-0.8	10.21	8.00	time	43053	4.43	2.27	Time	8807	2.27	2.27	4.83
50-4-0.2	1.15	0.00	233.55	190	0.13	0.00	46.63	28	0.00	0.00	5.64
50-4-0.5	2.57	0.00	6729.47	4332	0.47	0.00	262.42	76	0.00	0.00	5.9
50-4-0.8	5.02	1.80	86400	12659	1.80	0.00	1569.83	407	0.00	0.00	5.41
50-6-0.2	3.56	0.00	25218.14	7791	1.41	0.00	919.15	638	0.00	0.00	8.75
50-6-0.5	7.92	6.60	time	13173	4.57	2.32	time	3554	2.39*	2.39	9.00
50-6-0.8	8.20	7.70	time	8609	4.35	2.64	time	3771	2.64	2.66	8.44
50-8-0.2	6.41	6.60	time	11128	3.92	1.75	time	3646	1.75	1.76	12.54
50-8-0.5	10.76	9.60	time	10109	6.70	5.35	time	2961	5.35	5.35	12.37
50-8-0.8	10.60	10.10	time	14139	6.37	5.48	time	2485	5.48	5.55	10.54

solutions for the problem. In particular, it succeeds in finding the optimal solution (or the best known solution) for 41 out of the 42 instances, while using only a fraction of CPU time compared to that of the branch-and-cut algorithms. In only one instance ($n = 50, p = 6,$ and $\alpha = 0.5$), the branch-and-cut algorithm was able to improve the best solution obtained with GRASP by 0.07%. The percent average deviations over 30 runs ranges from 0.00% to 5.55%, thereby depicting the robustness of the GRASP algorithm. For 37 instances, GRASP yields the same solution in each run whereas the average deviation for the other 5 instances range from 0.02% to 0.55%.

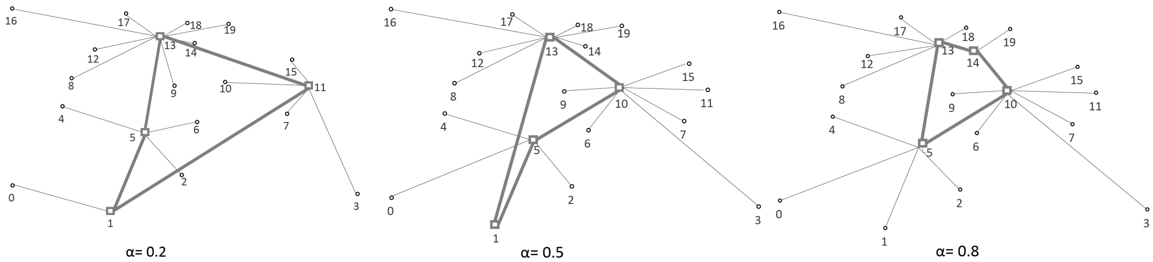


Figure 3.1: Optimal solutions for the CHLP for a 20 nodes instance.

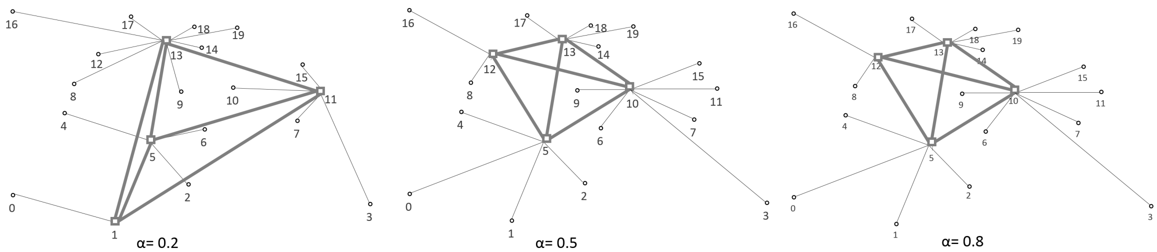


Figure 3.2: Optimal solutions for the p-hub median problem for a 20 nodes instance.

In order to further analyze the efficiency and robustness of proposed solution algorithms over large-scale instances, we have run a last set of computational experiments on instances ranging from 60 to 100 nodes. The results are summarized in Table 4.

It is worth mentioning that CPLEX fails to solve any of these instances within the time limit due to the size and complexity of problem. However, the exact branch-and-cut algorithm succeeds in solving 13 out the 24 instances to optimality and for the remaining instances, the final percent gap is within 6%. The GRASP is able to obtain the optimal solution for 12 out of the 13 instances that were solved to optimality using the exact algorithm. For the remaining one instance, the branch-and-cut was able to improve the best GRASP solution by 0.10%.

Table 3.5: Computational results for branch-and-cut and GRASP for large-scale instances

Instance	*CPLEX*	*Branch-and-Cut*				*Heuristic*		
$\lvert N \rvert$-P-α	% LP	$\%LP_{cut}$	% Gap	Sec	Nodes	% Dev	% Dev	Sec
60-4-0.2	1.69	0.96	0.00	242.93	224	0.01	0.01	8.52
60-4-0.5	2.99	1.09	0.00	760.68	188	0.01	0.01	10.06
60-4-0.8	5.41	2.10	0.00	8850.31	939	0.01	0.01	11.45
60-6-0.2	3.84	2.42	0.00	16838.45	3927	0.01	0.02	14.06
60-6-0.5	7.54	4.86	3.45	time	1801	3.45	3.58	19.45
70-4-0.2	1.57	0.83	0.00	729.41	379	0.00	0.00	13.54
70-4-0.5	3.33	1.43	0.00	4315.38	373	0.00	0.00	15.22
70-4-0.8	5.59	2.58	0.00	45543.09	1871	0.11*	0.22	18.37
70-6-0.2	3.88	2.05	0.00	17063.47	2221	0.00	0.00	24.58
70-6-0.5	7.83	5.43	4.26	time	802	4.26	4.26	27.86
75-4-0.2	1.52	1.06	0.00	1320.45	678	0.00	0.00	15.83
75-4-0.5	3.40	1.56	0.00	8595.06	501	0.00	0.00	20.97
75-4-0.8	5.64	2.54	0.22	time	1727	0.32*	0.32	22.83
75-6-0.2	3.94	2.58	0.25	time	2666	0.25	0.25	28.93
75-6-0.5	7.61	5.42	4.62	time	766	4.62	4.62	34.91
90-4-0.2	1.45	1.35	0.00	9477.66	2529	0.00	0.00	30.41
90-4-0.5	3.14	1.93	0.00	65523.91	1369	0.00	0.00	34.74
90-4-0.8	5.40	3.28	3.00	time	165	3.00	3.12	38.60
90-6-0.2	3.91	3.34	2.67	time	641	2.67	2.69	47.74
90-6-0.5	7.63	5.98	5.96	time	267	5.96	5.96	59.61
100-4-0.2	1.50	1.39	0.00	22352.74	2810	0.00	0.00	39.55
100-4-0.5	3.24	2.26	0.74	time	706	0.74	0.74	48.25
100-4-0.8	5.52	3.41	3.39	time	29	3.39	3.39	49.59
100-6-0.2	4.12	3.73	3.19	time	605	3.19	3.19	64.33

Chapter 4

An Exact Algorithm for the Modular Hub Location Problem

In this chapter, we study a *modular hub location problem* (MHLP) which considers explicitly the flow dependence of transportation costs based on modular arc costs. Thus, the total transportation cost is estimated not in terms of the per unit flow cost but in terms of the number of facility links used on each arc, eliminating the use of nonlinear functions and their linearizations to compute the discount factor for each hub arc. The cost is modeled using a stepwise function that determines, for each arc on the network, the total transportation cost as a function of the amount of flow routed through the arc. Our approach can be interpreted in terms of its ability to incorporate multiple capacity levels on the arcs. Another advantage is that it neither assumes a fully interconnected hub network nor a particular topological structure, instead it considers the design of the hub network as part of the decision process. Other variants of MHLP involving multiple assignments and direct connections were initially introduced in Mirzaghafour (2013). The material presented in this chapter is published in Tanash, Contreras, and Vidyarthi (2017).

The MHLP is related to other hub location models where capacities are considered at the arcs of the network. Sasaki and Fukushima (2003) study a capacitated multiple allocation HLP where capacity constraints are considered both on hub nodes and hub arcs. However, in their model, the flow between each O/D pair can go through at most one hub facility and hence there is no discount between hubs. Yaman and Carello (2005) introduce the capacitated single assignment hub location problem with modular link capacities (CHLP-ML) in which additional capacity constraints are considered on the incoming and outgoing flow at hubs. The CHLP-ML assumes hubs to be fully interconnected and does not consider O/D paths containing more than two hub nodes and one hub arc. Even though it is not explicitly mentioned in the paper, this model can be seen as one in which flow-dependent costs are considered. The authors present a quadratic mixed-inter programming formulation and compare different linearizations schemes based on the properties of the optimal solution. They also present a branch-and-cut algorithm and a tabu search heuristic for solving it. Corberán, Peiró, Campos, Glover, and Martí (2016) propose a metaheuristic algorithm based on strategic oscillation for the CHLP-ML that improves on the results obtained in Yaman and Carello (2005). However, to the best of our knowledge, the best heuristic algorithm for the CHLP-ML is given in Hoff, Peiró, Corberán, and Martí (2016), where a heuristic based on adaptive memory programing is developed to solve it. Yaman (2008) present a hub location model for a star-star network with modular link capacities in which hub nodes are directly connected to a central node. Rastani, Setak, and Karimi (2015) study a capacitated single-allocation HLP in which the capacities of hubs and hub links are parts of the decision process. The proposed model considers a flow-independent discount factor and assume hubs to be fully interconnected.

In this chapter, we present two mixed integer programming (MIP) formulations for the MHLP. The first formulation uses flow variables to compute the flow through hub arcs, whereas the second formulation uses path variables to determine whether a specified hub arc lies on the path between a pair of nodes. We propose a Lagrangean relaxation for the path-based formulation (PF) of MHLP by relaxing the linking constraints of the location/allocation and routing variables. This makes it possible to decompose the Lagrangean function into two independent subproblems which can be solved efficiently. We also propose a heuristic algorithm to obtain feasible solutions. To prove optimality, we develop a branch-and-bound algorithm that uses the Lagrangean relaxation and a heuristic to obtain lower and upper bounds at the nodes of the enumeration tree.

The remainder of this chapter is organized as follows. Section 1 formally defines the problem and presents the proposed formulations. In Section 2, we describe the proposed Lagrangean relaxation and study the structure of the subproblems and their solutions. Section 3 describes the primal heuristic algorithm. While in Section 4, we present a branch-and-bound algorithm. The computational results and analysis are presented in Section 5.

4.1 Problem Definition and Formulations

Let $G = (N, A)$ be a complete digraph without loops, where $N = \{1, \ldots, n\}$ is the set of nodes and A is the set of arcs. Let N also represent the set of potential locations, and let W_{ij} denote the amount of flow between nodes $i \in N$ and $j \in N$. Thus, $O_i = \sum_{j \in N} W_{ij}$ is the total flow originating at node $i \in N$, and $D_i = \sum_{j \in N} W_{ji}$ is the total flow destined to node $i \in N$. For each $i \in N$, f_i is the set-up cost for locating a hub facility. The distances between nodes i and j, $d_{ij} \geq 0$, are assumed to be neither symmetric nor satisfy the triangular inequality.

To estimate the transportation cost on both access and hub arcs, our model determines the number of facility links with a given capacity that will be needed to route the flow on these arcs. There is a fixed cost associated with these capacitated facility links as well as variable cost that depends on the amount of flow. Therefore, the transportation costs on arcs are modeled using a step-wise function. A hub arc $(k, m) \in A$ connects two different hub nodes k and m and has an associated transportation cost $c_{km} = l_c + b \times d_{km}$ for each facility link with capacity B used to route flow from k to m, where l_c and b represent the fixed and variable costs, respectively. Note that transportation costs of hub arcs model costs incurred when routing flow between two different hub facilities but not within the same facility. An access arc $(i, k) \in N \times N$ connects two nodes i and k, not necessarily different, and has an associated transportation cost $q_{ik} = l_q + p \times d_{ik}$ for each facility link with capacity R used to route flow from i to k, where l_q and p represent the fixed and variable costs, respectively. Note that access arcs model the transportation cost to collect and distribute flow between O/D nodes and hub facilities, even when an O/D node is selected to be a hub, i.e. $i = k$.

For each $i \in N$, let $v_i^1 = \lceil O_i/R \rceil$ denote the number of facility links required to route the flow originating from i directly to a hub, and let $v_i^2 = \lceil D_i/R \rceil$ denote the number of facility links required to route the flow from a hub directly to destination i. This means that, because of the single assignment assumption, the number of facility links required to connect each non-hub node to any hub node can be determined a priori and thus, it is not part of the decision process. The transportation costs on access arcs are thus flow independent because the number and actual utilization of the access arcs depend only on the flows O_i and D_i and the capacity R, which are parameters, and not on the assignment decisions. In order to account for the flow-dependent economies of scale when consolidating flows at hub facilities and using more efficient paths between hubs, we assume the following inequalities on hub arcs and access arcs: $B > R$, $b > p$ and $l_c > l_q$. This ensures that if facility links were fully loaded (or highly utilized), the unit transportation cost on hub arcs would be less than the unit flow cost on access arcs. That is, $\frac{c_{km}}{B} < \frac{q_{km}}{R}$. However, when links are only partially utilized it may happen that the unit transportation costs on hub arcs could be higher than those of access arcs.

Under these assumptions, the MHLP consists of locating a set of hub facilities, activating a set of hub arc facility links, allocating each node to exactly one hub, and determining the route of flows through the network such that the total setup and transportation cost is minimized. The model assumes a single assignment pattern of O/D nodes to hubs. As it is the case in other well-known hub location models with single assignments (Contreras, Díaz, & Fernández, 2011; Ernst & Krishnamoorthy, 1996), this assumption is consistent with applications in which outgoing and incoming flows of each non-hub node have to be processed by a single hub facility due to managerial or contractual reasons. However, an interesting feature of the MHLP is that it does not make any assumption on a particular topological structure to connect hub facilities. Instead, it considers a fixed set-up cost for the activation of hub arcs, allowing the model to select the most cost effective hub-level network structure. These features make the MHLP a very challenging problem to solve. Even if the location of hubs and the assignment of non-hub nodes to hubs are given, the remaining subproblem of activating facility links on the hub-level network is still NP-hard as it is equivalent to the well-known *network loading problem* (Magnanti et al., 1995).

In what follows, we present two MIP formulations for the MHLP based on the widely used path-based and flow-based formulations for classical HLPs (see, Contreras, 2015).

4.1.1 Path-Based Formulation

For each $i, k \in N$, we define binary variables z_{ik} equal to one if non-hub i is assigned to hub k. Note that, when $z_{kk} = 1$, node k is selected as a hub and assigned to itself. For each $(k, m) \in A$ we define integer variables y_{km} equal to the number of hub arcs between hub nodes k and m. For each $i, j, k, m \in N$, we also introduce continuous routing variables x_{ijkm} equal to the fraction of the flow originating from i and destined to j that is routed via hub arc (k, m). Using these sets of variables, the MHLP can be formulated as follows:

$$
\begin{aligned}
2(PF)\ \text{minimize} & \sum_{k\in N} f_k z_{kk} + \sum_{i\in N}\sum_{k\in N}\left(q_{ik}v_i^1 + q_{ki}v_i^2\right) z_{ik} \quad + \sum_{(k,m)\in A} c_{km} y_{km} && \\
\text{subject to} & \sum_{k\in N} z_{ik} = 1 && i \in N \quad (22)\\
& z_{ik} \leq z_{kk} && i, k \in N \quad (23)\\
& z_{ik} + \sum_{m\in N} x_{ijmk} = z_{jk} + \sum_{m\in N} x_{ijkm} && i, j, k \in N, i \neq j \quad (24)\\
& \sum_{i\in N}\sum_{j\in N} W_{ij} x_{ijkm} \leq B y_{km} && (k, m) \in A \quad (25)\\
& y_{km} \leq Q z_{mm} && (k, m) \in A \quad (26)\\
& y_{km} \leq Q z_{kk} && (k, m) \in A \quad (27)\\
& z_{ik} \in \{0, 1\} && i, k \in N \quad (28)\\
& y_{km} \in Z^+ && (k, m) \in A \quad (29)\\
& 0 \leq x_{ijkm} \leq 1 && i, j, k, m \in N. \quad (30)
\end{aligned}
$$

The objective function minimizes the sum of setup costs for locating hub facilities and the transportation cost on access and hub arcs. Note that when node i is assigned to hub k, the transportation costs of both access arcs (i, k) and (k, i) are considered to represent the collection and distribution cost, even when $i = k$. However, in the case of the transportation costs of hub arcs, these are considered only between two different hubs nodes k and m. Constraints (22) ensure that each non-hub node is assigned to exactly one hub. Constraints (23) ensure that each node is assigned to an open hub. Constraints (24) are the well-known flow conservation constraints, that are used to model O/D paths. Constraints (25) are capacity constraints that limit the amount of flow on each hub arc (k, m). Constraint (26) and (27) ensure that hub arc (k, m) is established only if k and m are hub nodes. Q is a sufficiently large number representing an upper bound on the number of hub arcs between hub nodes k and m. In this case, Q is set to $\lceil \sum_{i \in N} \sum_{j \in N} W_{ij}/B \rceil$. Constraints (28)-(30) are usual integrality and non-negativity constraints.

4.1.2 Flow-Based Formulation

For each $i \in N$ and $(k,m) \in A$, we define X_{ikm} equal to the amount of flow with origin i that traverse hub arc (k,m). We also use the z_{ik} and y_{km} variables for the location/allocation and network design decisions. The MHLP can then be formulated as follows:

$$(FF) \text{ minimize } \sum_{k \in N} f_k z_{kk} + \sum_{i \in N} \sum_{k \in N} \left(q_{ik} v_i^1 + q_{ki} v_i^2 \right) z_{ik} \quad + \sum_{(k,m) \in A} c_{km} y_{km}$$

$$\text{subject to } (22) - (23), (26) - (29)$$

$$\sum_{j \in N} W_{ij} z_{jk} + \sum_{m \in N} X_{ikm}$$

$$- \sum_{m \in N} X_{imk} - O_i z_{ik} = 0 \qquad i,k \in N \tag{31}$$

$$\sum_{i \in N} X_{ikm} \leq B y_{km} \qquad (k,m) \in A \tag{32}$$

$$X_{ikm} \geq 0 \qquad i,k,m \in N. \tag{33}$$

Constraints (31) are the flow conservation constraints whereas (32) are the capacity constraints.

4.2 Lagrangean Relaxation

Lagrangean relaxation (LR) is a well-known decomposition technique that exploits the inherent structure of the problem to obtain dual bounds on the optimal solution value (see, Guignard, 2003). LR has been successfully applied to solve different variants of HLPs

(An, Zhang, & Zeng, 2015; Contreras, Cordeau, & Laporte, 2011b; Contreras, Díaz, & Fernández, 2009). We now present a LR that is based on formulation PF for the MHLP. In the next section, we embed this relaxation into a branch-and-bound algorithm to obtain optimal solutions.

In the case of PF, if we relax constraints (24),(26), and (27), in a Lagrangean fashion, weighting their violations with multiplier vectors $\lambda^1, \lambda^2 \geq 0, \lambda^3 \geq 0$ of appropriate dimension, we obtain the following Lagrangean function:

$$
\begin{aligned}
2\ L(\lambda^1, \lambda^2, \lambda^3) = \min\ & \sum_{k \in N} f_k z_{kk} + \sum_{i \in N} \sum_{k \in N} q_{ik}(v_i^1 + v_i^2) z_{ik} + \sum_{(k,m) \in A} c_{km} y_{km} \\
& + \sum_{i \in N} \sum_{j \in N} \sum_{k \in N} \lambda^1_{ijk} (z_{ik} + \sum_{m \in N} x_{ijmk} - z_{jk} - \sum_{m \in N} x_{ijkm}) \\
& + \sum_{(k,m) \in A} \lambda^2_{km} (y_{km} - Q z_{mm}) + \sum_{(k,m) \in A} \lambda^3_{km} (y_{km} - Q z_{kk}) \\
\text{s.t.}\ & (22) - (23), (25), \text{ and } (28) - (30).
\end{aligned}
$$

For a given value of the Lagrangean multipliers $(\lambda^1, \lambda^2, \lambda^3)$, the Lagrangean function $L(\lambda^1, \lambda^2, \lambda^3)$ can actually be decomposed into two independent subproblems: one in the space of z variables and the other in the space of (x, y) variables. The subproblem in the space of z variables is:

$$
\begin{aligned}
2\ L_z(\lambda^1, \lambda^2, \lambda^3) = \min\ & \sum_{k \in N} \bar{F}_k z_{kk} + \sum_{i \in N} \sum_{k \in N} \bar{A}_{ik} z_{ik} \\
\text{s.t.}\ & (22), (23), (28),
\end{aligned}
$$

where the coefficients of the objective function are:

- $\bar{F}_k = f_k - \sum_{m \in N} Q \lambda^2_{mk} - \sum_{m \in N} Q \lambda^3_{km}$,
- $\bar{A}_{ik} = q_{ik}(v_i^1 + v_i^2) + \sum_{j \in N} (\lambda^1_{ijk} - \lambda^1_{jik})$.

Observe that the $L_z(\lambda^1, \lambda^2, \lambda^3)$ can be evaluated by solving a classical *uncapacitated facility location problem* (UFLP) (Cornuéjols, Nemhauser, & Wolsey, 1983). Even though this problem is known to be NP-hard, it can be solved in reasonable CPU times using ad-hoc solution algorithms.

The subproblem in the space of the (x, y) variables can be expressed as:

$$
\begin{aligned}
2\ L_{x,y}(\lambda^1, \lambda^2, \lambda^3) = \min \quad & \sum_{(k,m)\in A} \bar{R}_{km} y_{km} + \sum_{i\in N}\sum_{j\in N}\sum_{k\in N}\sum_{m\in N} \bar{M}_{ijkm} x_{ijkm} \\
\text{s.t.} \quad & (25), (29), (30),
\end{aligned}
$$

where the coefficients of the objective function are:

- $\bar{R}_{km} = \lambda^2_{km} + \lambda^3_{km} + c_{km}$,
- $\bar{M}_{ijkm} = \lambda^1_{ijm} - \lambda^1_{ijk}$.

Given that each of the y_{km} variables appear in exactly one constraint, we can further decompose $L_{x,y}(\lambda^1, \lambda^2, \lambda^3)$ into several independent subproblems, one for each (k, m) pair, of the form:

$$
\begin{aligned}
2\ L^{k,m}_{x,y}(\lambda^1, \lambda^2, \lambda^3) = \min \quad & \bar{R}_{km} y_{km} + \sum_{i\in N}\sum_{j\in N} \bar{M}_{ijkm} x_{ijkm} \\
\text{s.t.} \quad & (25), (29), (30).
\end{aligned}
$$

For a given candidate hub arc (k, m), the subproblem computes the optimal number of facility links to open and the commodities to be routed on this hub arc. These subproblems can be efficiently solved by iteratively setting y_{km} to a non-negative integer value and finding the optimal value for the associated x_{ijkm} variables. That is, upon fixing y_{km} the problem reduces to a continuous knapsack problem, which can be optimally solved with a greedy knapsack algorithm (Lawler, 1979). This algorithm first orders the x_{ijkm} variables so that

$$\frac{\bar{M}_{(s)km}}{W_{(s)}} \leq \frac{\bar{M}_{(s+1)km}}{W_{(s+1)}},$$

for $s = 1, \cdots, n^2 - n$, where $W_{(s)}$ denotes the demand flow of the s^{th} ordered node pair (i, j). Starting from $s = 1$, the algorithm adds the ordered items, i.e., $x_{(s)km} = 1$, one at a time to the knapsack and continues until the residual capacity is equal to zero or $M_{(s)km} > 0$. Note that only a fraction of the last considered item (denoted as r) may have been added, i.e., $x_{(r)km} = \left(By_{km} - \sum_{s=1}^{r-1} W(s)\right) / W(r) < 1$.

To determine the optimal value of y_{km}, the algorithms starts from $y_{km} = 1$ and evaluates the objective value by solving the corresponding continuous knapsack problem. If the objective value is strictly negative, y_{km} is increased by one to add B extra units of capacity to the knapsack so as to allow more $x_{(s)km}$ variables to take a positive value. The value of y_{km} is increased until the capacity increases to a point that all $x_{(s)km}$ can be set to one or whenever the next element to be added deteriorates the objective (i.e., $\bar{M}_{(s)km} \geq 0$). A value of $y_{km} = 0$ is selected as optimal whenever setting $y_{km} \geq 1$ yields strictly positive objective values.

4.2.1 Solving the Lagrangean Dual Problem

In order to obtain the best possible lower bound, we solve the *Lagrangean Dual* problem, which is given by:

$$(LD) \qquad L_D = \max_{\substack{\lambda^1 \\ \lambda^2, \lambda^3 \geq 0}} L(\lambda^1, \lambda^2, \lambda^3).$$

We apply the subgradient optimization method to solve problem LD. It is well known that the classical subgradient algorithm tends to suffer from slow convergence. To overcome this difficulty, we use a deflected subgradient algorithm. This algorithm uses a linear combination of the current subgradient direction s^t and the direction used in the previous iteration d^{t-1} to obtain the next direction of movement. That is, at every iteration t, $d^t = s^t + \theta^t d^{t-1}$. The efficiency of this method depends on selecting the deflected subgradient parameter θ^t (see for instance, Brännlund, 1995; Camerini, Fratta, & Maffioli, 1975). To this end, we use the following rule based on geometrical arguments (see, Guta, 2003):

$$\theta^t = \begin{cases} -\pi \frac{s^t d^{t-1}}{\|d^{t-1}\|^2} & if\ s^t d^{t-1} < 0, \\ 0 & \text{otherwise,} \end{cases}$$

where $0 \leq \pi \leq 2$. For a given vector $(\lambda^1, \lambda^2, \lambda^3)$, let $z(\lambda)$, $y(\lambda)$, and $x(\lambda)$ be the optimal solution to $L(\lambda^1, \lambda^2, \lambda^3)$. Thus, a subgradient of $L(\lambda^1, \lambda^2, \lambda^3)$ is given by

$$\begin{aligned} s(\lambda^1, \lambda^2, \lambda^3) \;\; = \;\; & \Bigg(\left(z_{ik}(\lambda) + \sum_{m \in N} x_{ijmk}(\lambda) - z_{jk}(\lambda) - \sum_{m \in N} x_{ijkm}(\lambda) \right)_{(i,j,k)}, \\ & \left(y_{km}(\lambda) - Q z_{kk}(\lambda) \right)_{(k,m)}, \\ & \left(y_{km}(\lambda) - Q z_{mm}(\lambda) \right)_{(k,m)} \Bigg). \end{aligned}$$

At each iteration t of the subgradient algorithm, the dual multipliers are updated as:

$$(\lambda^1, \lambda^2, \lambda^3)^{(t+1)} = (\lambda^1, \lambda^2, \lambda^3)^{(t)} + \delta^t \frac{\bar{\phi} - L((\lambda^1, \lambda^2, \lambda^3)^t)}{||\Gamma(\lambda^1, \lambda^2, \lambda^3)^t||^2} d^t,$$

where $\bar{\phi}$ denotes an upper bound on the optimal solution value and δ is a constant between 0 and 2.

4.2.2 Primal Heuristic

We exploit the information generated at some iterations of the subgradient algorithm to construct feasible solutions. In what follows, solutions are represented by a set of hub nodes H, a set of hub arcs D, and an assignment mapping M. Solutions are designated in the form $s = (H, D, M)$, where H represents the set of selected sites at which hubs are located, i.e., $H(i) = 1$ if site $i \in N$ is chosen to be a hub, $D((i,j)) : A \to Z^+$ represents the number of facility links installed on hub arcs (i, j) and $M : N \to H$ is the assignment mapping, i.e., $M(j) = k$ if node $j \in N$ is assigned to hub $k \in H$.

The proposed heuristic constructs feasible solutions as follows. Let $\hat{z}^t$, $\hat{y}^t$ and $\hat{x}^t$ be the optimal solution to the Lagrangean subproblems $L_z(\lambda)$ and $L_{x,y}(\lambda)$ at a given iteration t of the subgradient algorithm. The optimal solution of the subproblem $L_z(\lambda^t)$ provides a set of hubs and an assignment mapping of non-hub nodes to hubs, that is $H = \{k : \hat{z}_{kk} = 1, k \in N\}$, and $M(i) = \bar{k}$ where $\hat{z}_{i\bar{k}} = 1$. Since $L_z(\lambda^t)$ and $L_{x,y}(\lambda^t)$ are solved independently, the solution obtained from the subproblem $L_{x,y}(\lambda^t)$ might not be feasible for the set H of hubs obtained in solving L_z. Therefore, once the location/allocation variables are fixed, the next step is to determine the number of facility links to be activated on each hub arc in order to route the flows at minimum cost. This subproblem is actually equivalent to solving a *network loading problem* (NLP) on an auxiliary network.

Let $\hat{G} = (\hat{H}, \hat{A})$ be a directed graph where $\hat{H} = \{k \in N : \hat{z}_{kk} = 1\}$ is the set of open hubs at iteration t and $\hat{A} = \{(k, m) \in A : k, m \in \hat{H}\}$ is the set of candidate hub arcs. For each pair $(k, m) \in \hat{H} \times \hat{H}$, let $w_{km} = \sum_{i \in O(k)} \sum_{j \in O(m)}$ denote the amount of flow that needs to be routed from k to m, where $O(k) = \{i \in N : \hat{z}_{ik} = 1\}$. Recall that c_{km} represents the (transportation) cost for using one facility link with capacity B on hub arc (k, m). Using the y_{km} and x_{ijkm} variables defined in Section 2, the NLP can be formulated as:

$$\text{minimize} \quad \sum_{(k,m)\in\hat{A}} c_{km} y_{km}$$

$$\text{subject to} \quad \sum_{i\in\hat{H}} \sum_{j\in\hat{H}} w_{ij} x_{ijkm} \leq B y_{km} \qquad (k,m) \in \hat{A}$$

$$\sum_{m\in\hat{H}} x_{ijkm} - \sum_{m\in\hat{H}} x_{jikm} = \begin{cases} 1 & \text{if } k = i, \\ -1 & \text{if } k = j. \\ 0 & \text{if } k \neq i, j. \end{cases} \qquad i, j, k \in \hat{H}$$

$$0 \leq x_{ijkm} \leq 1 \qquad i, j, k, m \in \hat{H}$$

$$y_{ij} \in Z^+ \qquad (i, j) \in \hat{A}.$$

Even though the NLP is known to be a NP-hard, for instances of reasonable size it can be solved efficiently using a general purpose solver. The output of the NLP is a set of hub arcs to open and the associated routing decisions for all demand flow of the MHLP. Thus, the optimal solution of the NLP provides a feasible solution to the MHLP. The overall Lagrangean relaxation algorithm is depicted in Algorithm 1.

This constructive phase of the heuristic is executed every time the subgradient algorithm improves the best known lower bound. Once the subgradient algorithm terminates, we apply a local search procedure on the best known solution obtained so far. This procedure iteratively explores two neighborhoods namely classical shift and swap neighborhoods.

In the shift neighborhood, a non-hub node i currently assigned to hub m, is reassigned to a different open hub k. In the swap neighborhood, two non-hub nodes i and j currently assigned to hubs m and k with $k \neq m$, respectively, are reallocated to k and m, respectively.

Algorithm 1: Lagrangean relaxation heuristic

Initialize $z_D \leftarrow -\infty$; $(\lambda^1, \lambda^2, \lambda^3)^0 \leftarrow 0$; $\delta^0 \leftarrow 2$; $\bar{\phi} \leftarrow \infty$; $t \leftarrow 0$; $\pi \leftarrow 1.5$
while (*Stopping criteria not satisfied*) **do**
  Solve the Lagrangean function $L((\lambda^1, \lambda^2, \lambda^3)^t)$
  if $(L((\lambda^1, \lambda^2, \lambda^3)^t)) > z_D)$ **then**
    $z_D \leftarrow L((\lambda^1, \lambda^2, \lambda^3)^t)$
    Apply constructive heuristic to obtain upper bound UB^t
    if $(UB^t < \bar{\phi})$ **then**
      $\bar{\phi} \leftarrow UB^t$
    end if
  end if
  Evaluate the subgradient $\gamma(\lambda^1, \lambda^2, \lambda^3)^t$
  if($\gamma(\lambda^1, \lambda^2, \lambda^3)^t d^{t-1} < 0)$ **then**
    $\theta^t = -\pi\gamma(\lambda^1, \lambda^2, \lambda^3) d^{t-1} / \parallel d^{t-1} \parallel^2$
  else
    $\theta^t = 0$
  end if
  Obtain the direction $d^t = \gamma(\lambda^1, \lambda^2, \lambda^3)^t + \theta^t d^{t-1}$
  Calculate the step length $s^t \leftarrow \delta^t \frac{\bar{\phi} - L((\lambda^1, \lambda^2, \lambda^3)^t)}{||\gamma(\lambda^1, \lambda^2, \lambda^3)^t||^2}$,
  Set $(\lambda^1, \lambda^2, \lambda^3)^{(t+1)} \leftarrow (\lambda^1, \lambda^2, \lambda^3)^{(t)} + s^t d^t$
  Set $t \leftarrow t + 1$
end while

Let $s = (H, A, M)$ be the current solution, then

$$\mathcal{N}_{shift}(s) = \{s^{'} = (H, A, M^{'}) : \exists! j \in N,\ M^{'}(j) \neq M(j)\},$$

and

$$\mathcal{N}_{swap}(s) = \{s^{'} = (H, A, M^{'}) : \exists!(j_1, j_2),\ j_1^{'} = M(j_2),\ j_2^{'} = M(j_1),\ \forall j \neq j_1, j_2\}.$$

The local search procedure explores $\mathcal{N}_{shift}$ first until a local optimal solution is found. The algorithm then tries to improve the solution by exploring $\mathcal{N}_{swap}$. Each time the search improves the best known solution, the procedure starts with $\mathcal{N}_{shift}$. In both neighborhoods, a best improvement strategy is used. Note that in order to reoptimize the arc selection and routing decisions in each neighbor, we solve one NLP to optimality. An outline of the local search scheme is depicted in Algorithm 2.

Algorithm 2: Local search procedure

$stoppingcriteria \leftarrow false$
while($stoppingcriteria = false$)**do**
    $explore\ \mathcal{N}_{shift}$
    if(solution not improved in $\mathcal{N}_{shift}$) **then**
        $explore\ \mathcal{N}_{swap}$
        if(solution has not been updated) **then**
            $stoppingcriteria \leftarrow true$
        end-if
    end-if
end-while

4.3 Branch-and-Bound Algorithm

We describe a branch-and-bound algorithm for solving the MHLP to optimality. It uses the Lagrangean relaxation to obtain lower and upper bounds at every node of the enumeration tree. It is composed of three phases. In the first phase, the enumeration tree is created by branching on the location variables z_{kk}, producing terminal nodes in which all location variables have been fixed. The second phase proceeds from each unfathomed node, creating an enumeration tree by branching on the assignment variables z_{ik}. When all the location and allocation variables are fixed, the third phase finds the optimal link activation and routing decisions for each unfathomed node by solving an associated NLP.

Let $(\bar{z}, \bar{y}, \bar{x})$ be the best solution found at the end of the Lagrangean relaxation algorithm at any node of the tree. The branching strategy used in the first phase of the enumeration tree is as follows. If there are any unfixed location variables such that $\bar{z}_{kk} = 1$, we select among these the one with the largest reduced cost $\bar{F}_k$ and explore the branch with $\bar{z}_{kk} = 1$. We store the associated branch with $\bar{z}_{kk} = 0$ on a list of unexplored nodes for later. When there are no more location variables which have not been fixed in the tree such that $\bar{z}_{kk} = 1$, we branch on the remaining unfixed variables by selecting the one with the largest reduced cost and explore the branch with $\bar{z}_{kk} = 1$. The first phase is completed once all locational decisions have been fixed.

When some of the nodes of the first phase have not been fathomed, we continue with the second phase. In this phase, we select each of these unfathomed nodes from the previous phase, one at a time, in non-decreasing way with respect to their lower bounds obtained and branch on the assignment variables. During this phase, the tree is not binary. That is, the number of branches generated at a node of the tree when selecting a non-hub node i for branching is equal to the number of open hubs on its path. The non-hub nodes are selected to be explored in the order of decreasing values of the highest reduced cost associated with the $\bar{z}_{ik}$ variables.

When all nodes of the second phase have been explored, but a subset of terminal nodes (i.e., nodes of depth n) have not been eliminated we move to the third (and last) phase of the algorithm. Note that at this point all locational and assignment decisions have been fixed and thus, the resulting subproblems reduces to a NLP. For each of these unfathomed nodes, we solve an associated NLP to optimality. We explore the entire enumeration tree in a depth first search fashion. At each node of the tree, the dual multipliers are initialized using the dual solutions from its parent node.

4.4 Computational Experiments

We run computational experiments to compare and analyze the performance of the formulations, the Lagrangean relaxation and the branch-and-bound algorithm. All formulations and algorithms have been coded in C++ and run on an HP station with an Intel Xeon CPU E3-1240V2 processor at 3.40GHz and 24 GB of RAM under windows 7 environment. All MIP problems have been solved using Concert technology of CPLEX 12.5.1.

We generate a set of benchmark instances for the MHLP using the well known Australian post (AP) instances which can be downloaded from the OR library (see, Beasley, 1990). The AP data set consists of postal flow and Euclidean distances between 200 districts in an Australian city. In our experiments, we have selected problems with $|N|$ =10, 20, 25, 40, 50, 60, and 75 nodes, and disregarded the flows W_{ii} for each $i \in N$, i.e., $W_{ii} = 0$. For each problem size, we generated 9 instances for the MHLP. Each instance comprises a hub facility link capacity chosen from $B \in \{200, 300, 400, 500, 600, 650, 750\}$ with an associated variable cost in $b \in \{450, 500, 600, 800\}$, and a facility link capacity on access arcs chosen from $R \in \{100, 150, 200\}$ with an associated variable cost in $p \in \{300, 345, 400, 500\}$. The choice of the parameters in each generated instance is such that $\frac{c_{km}}{B} < \frac{q_{km}}{R}$ where $B > R$, $b > p$ and l_c=l_q=0, to guarantee that the unit transportation cost on hub arcs, when fully utilized, is smaller than the unit transportation cost on access arcs. That is, there is a potential discount factor associated with hub arcs due to consolidation and use of more efficient modes of transportation between hub nodes. We note that $\frac{b/B}{p/R}$ corresponds to the smallest discount factor that can be achieved on hub arcs when compared to the transportation cost of access arcs when fully utilized. In practice, some arcs might be underutilized and thus, the actual discount factor may be higher which in turn, may lead to a lower unit cost on access arcs than on some hub arcs.

In order to generate a variety of instances, we selected the values for the capacities and costs in such a way that we obtain different potential discount factors on hub arcs. In particular, we consider the following configurations:

i) for $\frac{b/B}{p/R} = 0.2$:

- $L1$:$(B = 750,\ R = 100,\ b = 600,\ p = 400)$,
- $L2$:$(B = 750,\ R = 100,\ b = 450,\ p = 300)$,
- $L3$:$(B = 600,\ R = 100,\ b = 600,\ p = 500)$,

ii) for $\frac{b/B}{p/R} = 0.4$:

- $L4 : (B = 400,\ R = 100,\ b = 800,\ p = 500)$,
- $L5 : (B = 650,\ R = 150,\ b = 600,\ p = 345)$,
- $L6 : (B = 500,\ R = 100,\ b = 600,\ p = 300)$,

iii) for $\frac{b/B}{p/R} = 0.63$:

- $L7 : (B = 200,\ R = 100,\ b = 500,\ p = 400)$,
- $L8 : (B = 300,\ R = 150,\ b = 500, p = 400)$,
- $L9 : (B = 400,\ R = 200,\ b = 500,\ p = 400)$.

For each one of these nine configurations, we generate seven different instances, one for each size of network. Therefore, we generated a total of 63 instances.

In all the experiments, the subgradient algorithm terminates when one of the following criteria has been met: *i)* the difference between the upper and lower bound is below a given threshold value, i.e.$|\bar{\phi} - z_D^t| < \epsilon$, *ii)* the improvement on the lower bound after t_{max} consecutive iterations is below a threshold value ψ, *iii)* the maximum number of iterations $iter_{max}$ has been reached.

After some tuning, we set the following parameters to: $\epsilon = 10^{-6}$, $\psi = 0.05$, and $t_{max} = 150$. For the first stage in the branch-and-bound algorithm, we set the maximum number of subgradient iterations at the root node to $iter_{max} = 4,000$ and to $iter_{max} = 25$ for the rest of the nodes. The parameter δ has been reduced by 0.25 after 100 consecutive iterations without improvement in the lower bound. In the second stage, the maximum number of subgradient iterations has been fixed to $iter_{max} = 300$ at the root node and $iter_{max} = 25$ for the rest of the nodes.

4.4.1 Comparison of Formulations and Algorithm

The first set of computational experiments has been performed to compare the path-based formulation (PF) with the flow-based formulation (FF) when solved using CPLEX. Throughout experiments, we used the default settings of CPLEX. The detailed results of this comparison on a set of instances ranging from 10 to 40 nodes are reported in Table 1. The first column provides the number of nodes, n, and the instance name $(n - name)$. The next set of columns reports the linear programming relaxation gap $(\%LP)$, the linear programming relaxation gap after adding CPLEX cuts $(\%LP_{cut})$, the percent deviation between the final upper and lower bounds (%GAP), the CPU time in seconds (CPU), and the number of explored node in the enumeration tree (Nodes), for both formulations. The $\%LP$ gap has been computed as $(UB - LP)/UB \times 100$, where UB is the best upper bound (or the optimal solution value), and LP is the optimal value of the LP relaxation. The final percent gap %GAP has been evaluated as $(UB - LB)/UB \times 100$, where UB and LB denote the best upper and lower bounds obtained at termination, respectively. Throughout experiments, the maximum time limit is set to one day of CPU time. Instances that could not be solved to optimality within this time limit have been marked with the label “time”.

As can be seen in Table 1, PF is able to optimally solve 17 out of the 36 instances within the time limit. The percent LP gap of PF ranges from 2.21 to 10.15. The column $\%LP_{cut}$ shows that the addition of CPLEX cuts has a significant impact on the improvement of the lower bound at the root node of the tree. Nevertheless, CPLEX is unable to solve the LP relaxation for all 40-node instances in one day of CPU time. In the case of the FF, CPLEX is able to solve 26 out of the 36 instances within the time limit. The $\%LP$ gaps for the instances that have been solved using PF are slightly better than that obtained in the FF. The percent LP gap of FF ranges from 3.36 to 11.48. However, given that there is a considerably smaller number of variables and constraints in FF, CPLEX is able to optimally solve all 25-node instances and one of the 40-node instances that the PF cannot

Table 4.1: Comparison between path-based and flow-based formulations

Instance	Path-based formulation (PF)					Flow-based Formulation (FF)				
	% LP	% LP_{cut}	%GAP	CPU	Nodes	% LP	% LP_{cut}	%GAP	CPU	Nodes
10-L1	7.75	2.03	0.00	39	374	8.61	3.84	0.00	< 5	935
10-L2	4.30	1.87	0.00	< 5	39	4.80	2.05	0.00	< 5	48
10-L3	9.06	2.58	0.00	142	2,521	9.38	5.47	0.00	33	22,449
10-L4	10.15	4.35	0.00	1,187	23,408	11.48	6.96	0.00	129	43,548
10-L5	4.01	2.65	0.00	< 5	35	4.59	3.03	0.00	< 5	25
10-L6	5.85	3.10	0.00	23	452	6.44	3.28	0.00	< 5	317
10-L7	4.47	3.05	0.00	28	715	6.31	3.39	0.00	< 5	676
10-L8	3.61	1.54	0.00	< 5	52	4.50	2.13	0.00	< 5	69
10-L9	5.02	3.99	0.00	11	199	6.24	4.52	0.00	< 5	467
20-L1	5.72	2.16	1.07	86,400	3,529	6.14	4.01	0.00	1,027	20,977
20-L2	2.96	1.44	0.00	6,543	603	3.36	2.10	0.00	67	1,451
20-L3	8.22	2.85	2.74	86,400	3,098	8.31	7.68	0.00	10492	679,516
20-L4	5.93	2.17	0.00	78,393	6,547	6.68	4.04	0.00	1,707	34,623
20-L5	3.48	1.69	0.00	1,286	472	3.92	3.70	0.00	72	1,485
20-L6	5.05	3.74	0.00	24,329	3,276	5.24	5.18	0.00	133	5,224
20-L7	3.35	2.80	0.00	24,049	3,960	4.15	3.53	0.00	166	8,135
20-L8	2.98	2.32	0.00	6,055	575	3.80	3.05	0.00	101	2,633
20-L9	2.97	2.26	0.00	3,116	357	3.67	2.78	0.00	56	1,091
25-L1	8.82	4.07	4.07	time	0	9.02	6.29	0.00	74,023	312,521
25-L2	3.84	1.84	1.79	time	3	4.36	3.57	0.00	3,300	14,906
25-L3	9.15	3.97	3.97	time	0	9.34	8.57	2.15	time	242,779
25-L4	8.39	3.17	3.14	time	200	8.82	8.69	0.29	time	378,612
25-L5	3.94	2.30	2.14	time	1,330	4.68	4.34	0.00	1,503	14,439
25-L6	4.11	2.89	2.89	time	539	4.96	4.72	0.00	2,180	24,962
25-L7	2.21	1.74	1.16	time	1,329	3.97	3.69	0.00	1,928	17,138
25-L8	2.45	1.77	0.00	time	873	3.39	3.05	0.00	560	4,023
25-L9	3.36	2.15	2.00	time	1,038	4.61	3.94	0.00	646	4,987
40-L1	n.a	n.a	n.a	time	n.a	6.91	6.89	4.08	time	167,604
40-L2	n.a	n.a	n.a	time	n.a	3.52	3.23	0.00	51,740	37,187
40-L3	n.a	n.a	n.a	time	n.a	7.43	7.35	5.14	time	289,952
40-L4	n.a	n.a	n.a	time	n.a	7.41	6.79	4.84	time	148,985
40-L5	n.a	n.a	n.a	time	n.a	5.12	4.88	2.63	time	69,828
40-L6	n.a	n.a	n.a	time	n.a	4.67	4.64	0.75	time	52,983
40-L7	n.a	n.a	n.a	time	n.a	4.21	4.12	0.86	time	107,754
40-L8	n.a	n.a	n.a	time	n.a	5.41	4.99	3.08	time	82,524
40-L9	n.a	n.a	n.a	time	n.a	5.73	4.71	3.55	time	38,074

solve. Moreover, FF was able to provide optimality gaps for the remaining unsolved 40-node instances.

In order to analyze the performance of our proposed exact algorithm, we conduct a second series of computational experiments using a set of instances ranging from 10 to 50 nodes. The results are summarized in Table 2. The first five columns have the same meaning as in Table 1. The next two columns under heading LR provide duality gap of the best lower bound obtained with Lagrangean relaxation with respect to the best known solution ($\%LR$) and the CPU time in seconds needed to obtain both lower and upper bounds using Lagrangean relaxation (CPU). The results of the columns under heading

$Branch\ and\ Bound$ report: the final percent deviation at termination ($\%Gap$), the CPU time in second (CPU), and the number of the explored nodes in the enumeration tree (Nodes).

The results in Table 2 show that by using FF, we were able to solve 26 out of the 45 problem instances to optimality using CPLEX (final percent gaps on the remaining instances range from 0.29 to 8.87). The exact algorithm, on the other hand, was able to confirm the optimality of the solutions obtained in 35 out of the 45 instances within the CPU time limit. For the remaining 10 unsolved instances, the final percent deviation is below 2.8. In all instances considered, the percent deviation of the LR is smaller than the one obtained with FF even after the addition of CPLEX cuts. As a result, the proposed algorithm produces significantly smaller enumeration trees and is much faster than CPLEX for all instances, except on the small size, 10-node instances. Moreover, our exact algorithm is able to optimally solve 9 instances that FF is unable to solve within the time limit. For the instances that have not been solved to optimality, our algorithm always provides much smaller percent gaps as compared to FF. We note that the percent of time taken by the algorithm for solving the UFLPs at every iteration and the NLPs during the local search and at the end of the enumeration tree never exceeds 5% of the total computational time for the larger instances with 40 and 50 nodes.

In order to further analyze the performance of our proposed algorithm, we have run a third series of computational experiments using 60-node and 75-node instances. The results are summarized in Table 3. The column $CPU_{LP_{cuts}}$ reports the computational time in seconds to solve the LP relaxation and to add CPLEX cuts whereas the other columns have the same meaning as in the previous tables.

As can be seen in Table 3, the lower bounds obtained from Lagrangean relaxation are significantly tighter than those obtained with FF. In particular, the LP gaps of FF range from 6% to 13%, whereas the LP gaps of the Lagrangean relaxation algorithm never exceed

Table 4.2: Results of branch-and-bound algorithm for small to medium-size instances

Instance	Flow-based Formulation				LR		Branch and Bound		
	% LP_{cut}	%GAP	CPU	Nodes	$\%LR$	CPU	%GAP	CPU	Nodes
10-L1	3.84	0.00	< 5	935	2.02	10	0.00	22	213
10-L2	2.05	0.00	< 5	48	1.86	6	0.00	9	42
10-L3	5.47	0.00	33	22,449	2.48	50	0.00	88	624
10-L4	6.96	0.00	129	43,548	4.52	22	0.00	337	4,537
10-L5	3.03	0.00	< 5	25	2.20	12	0.00	19	87
10-L6	3.28	0.00	< 5	317	3.18	7	0.00	15	127
10-L7	3.39	0.00	< 5	676	3.61	29	0.00	80	714
10-L8	2.13	0.00	< 5	69	1.58	13	0.00	22	137
10-L9	4.52	0.00	< 5	467	4.03	13	0.00	21	180
20-L1	4.01	0.00	1,027	20,977	1.79	35	0.00	102	1,268
20-L2	2.10	0.00	67	1,451	1.47	23	0.00	48	383
20-L3	7.68	0.00	10492	679,516	1.83	83	0.00	308	4,624
20-L4	4.04	0.00	1,707	34,623	2.05	47	0.00	241	4,173
20-L5	3.70	0.00	72	1,485	1.50	42	0.00	58	349
20-L6	5.18	0.00	133	5,224	3.75	37	0.00	149	3,102
20-L7	3.53	0.00	166	8,135	3.22	79	0.00	356	5,116
20-L8	3.05	0.00	101	2,633	2.38	39	0.00	78	818
20-L9	2.78	0.00	56	1,091	2.32	37	0.00	77	764
25-L1	6.29	0.00	74,023	312,521	1.57	129	0.00	697	9,155
25-L2	3.57	0.00	3,300	14,906	0.95	87	0.00	167	700
25-L3	8.57	2.15	time	242,779	2.01	145	0.00	887	11,632
25-L4	8.69	0.29	time	378,612	2.58	450	0.00	3,163	41,232
25-L5	4.34	0.00	1,503	14,439	2.03	67	0.00	276	1,957
25-L6	4.72	0.00	2,180	24,962	2.79	82	0.00	439	5,147
25-L7	3.69	0.00	1,928	17,138	2.27	168	0.00	1,331	13,750
25-L8	3.05	0.00	560	4,023	1.55	82	0.00	289	3,020
25-L9	3.94	0.00	646	4,987	1.77	101	0.00	249	2,185
40-L1	6.89	4.08	time	167,604	1.78	699	0.00	16,144	78,423
40-L2	3.23	0.00	51,740	37,187	0.64	437	0.00	865	1,177
40-L3	7.35	5.14	time	289,952	1.76	734	0.00	32,290	174,658
40-L4	6.79	4.84	time	148,985	2.68	731	1.91	time	290,062
40-L5	4.88	2.63	time	69,828	1.79	135	0.00	4,153	15,151
40-L6	4.64	0.75	time	52,983	2.48	497	0.00	24,200	101,864
40-L7	4.12	0.86	time	107,754	2.85	786	0.00	40,284	152,099
40-L8	4.99	3.08	time	82,524	2.35	681	0.00	59,693	323,662
40-L9	4.71	3.55	time	38,074	1.87	578	0.00	78,105	431,532
50-L1	8.33	7.69	time	62,875	2.79	1,502	1.84	time	239,520
50-L2	6.52	5.86	time	21,077	2.35	1,613	2.00	time	250,642
50-L3	9.76	9.53	time	125,427	2.41	1,903	2.16	time	166,954
50-L4	8.62	7.49	time	154,566	2.56	3,675	2.01	time	190,193
50-L5	9.21	8.56	time	14,841	4.45	1,224	2.61	time	210,039
50-L6	5.83	5.23	time	30,198	2.23	1,305	1.67	time	226,644
50-L7	5.03	3.56	time	105,850	3.34	2,169	1.79	time	173,901
50-L8	6.53	5.75	time	76,346	3.41	1,963	2.72	time	176,399
50-L9	5.40	5.30	time	9,144	1.87	1,319	0.12	time	283,396

5%. It is worth mentioning that both FF and our algorithm are unable to solve any of these 60 and 75 nodes instances within the time limit of one day of CPU due to the size and complexity of the problem. However, the final gap of our branch-and-bound algorithm is at most 3.05%.

4.4.2 Sensitivity Analysis

In the last part of the experiments, we perform a sensitivity analysis on some of the parameters of the problem to analyze the changes in optimal solution networks. In particular, Tables 4 to 7 show how optimal network configurations change depending on: (i) the

Table 4.3: Results of branch and bound algorithm for 60 and 75-node instances

Instance	Flow-based Formulation		LR		Branch and Bound		
	$\%LP_{cuts}$	$CPU_{LP_{cuts}}$	$\%LR$	CPU	%GAP	CPU	Nodes
60-L1	8.76	4,419	2.00	3,129	1.73	time	118,013
60-L2	7.82	3,338	1.75	2,523	1.53	time	131,604
60-L3	9.78	2,408	2.29	3,347	2.09	time	21,369
60-L4	8.57	1,877	2.95	4,191	2.13	time	15,657
60-L5	8.70	2,682	1.90	2,880	1.61	time	130,091
60-L6	8.03	3,392	2.69	3,128	2.13	time	98,955
60-L7	6.02	1,553	4.33	4,037	2.45	time	14,426
60-L8	7.60	1,920	3.69	3,346	2.56	time	15,460
60-L9	7.04	1,543	2.60	4,772	1.58	time	93,931
75-L1	11.05	11,339	2.30	9,678	2.21	time	5,721
75-L2	8.79	10,851	1.83	8,219	1.52	time	7,680
75-L3	10.09	8,390	2.00	15,620	0.86	time	4,940
75-L4	11.01	11,640	3.18	23,013	2.11	time	2,249
75-L5	12.64	11,727	2.80	9,069	0.28	time	4,391
75-L6	10.85	13,972	3.55	8,070	2.67	time	4,198
75-L7	8.25	11,235	4.96	13,183	3.05	time	2,488
75-L8	9.12	8,639	3.55	13,160	1.70	time	3,883
75-L9	8.61	9,888	2.39	10,451	0.44	time	4,959

capacity of hub arcs (B), (ii) the capacity of access arcs (R), (iii) the variable cost of access arcs (p), and (iv) the variable cost of hub arcs (b), respectively. The tables present the optimal network configuration - hub nodes, hub arcs, number of hub facility links (y_{km}), and the percent hub arc utilization (% Utilization). That is, the hub arc utilization measures how much of the available capacity is being used on each hub arc (k, m) and has been computed as $\sum_{i \in N} \sum_{j \in N} W_{ij} \bar{x}_{ijkm} / (B \bar{y}_{km}) 100$, where $(\bar{x}, \bar{y})$ denotes the optimal solution.

Table 4 illustrates the effect of changing the capacity of hub arcs (B) on optimal solution networks while the rest of the parameters remain fixed. We observe that at higher capacity levels of hub arcs, more hubs are opened. However, the MHLP tries to utilize hub arc facilities by activating fewer hub arcs and thus, resulting in a cost efficient hub-and-spoke network structure. For instance, with $B = 650$, MHLP opens six hubs at nodes 1, 3, 6, 8, 10, and 13 and connects them with only nine hub arcs. The average arc utilization is 75.7%. When B decreases to 325, the model closes hub 3 while the rest is open and increases the number of active hub arcs to 11. The average utilization increases to 86.86%. Upon decreasing B further to 206, only four hubs are opened at nodes $3, 5, 10$ and 13 with eight hub arcs. Note that the average utilization increases to 88.5%. In all cases, nodes 10 and 13 are chosen as hubs. In addition, we note that larger O/D paths are used when hub arc capacities are higher. For example, the flow routed from node 16 to node 0 visits five hub facilities when $B = 650$, four hubs when $B = 325$, and only two when $B = 206$.

Table 4.4: Effect of varying capacity of hub arcs on optimal solution networks with $n = 20$, $R = 110$, $p = 634$ and $b = 750$.

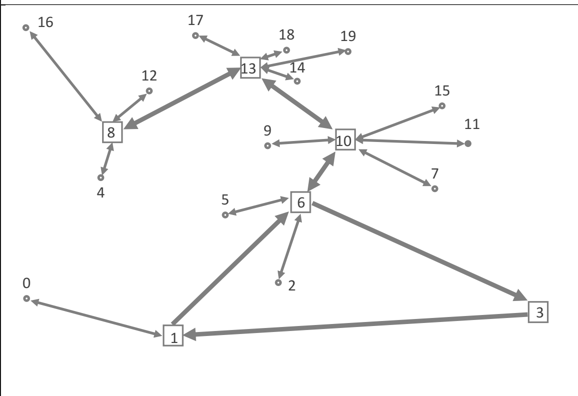
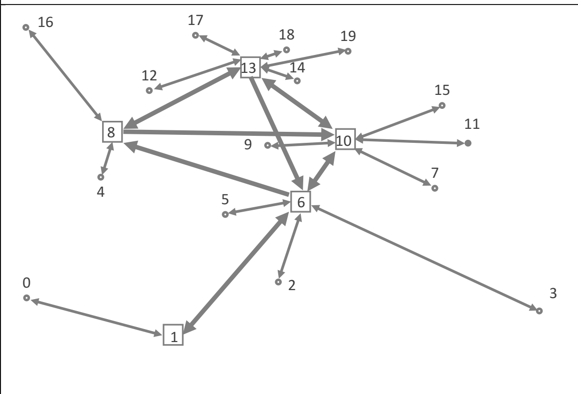
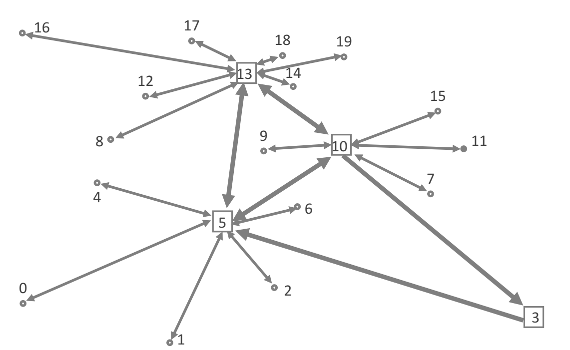
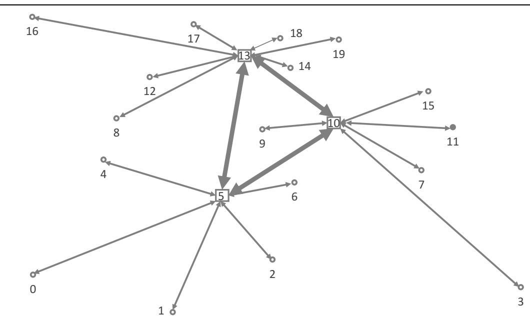
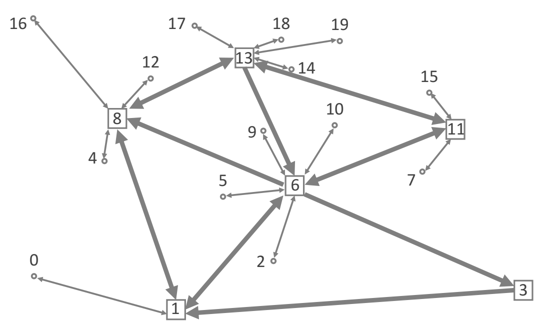
Capacity	Solution network	Hub arcs	y_{km}	% Utilization
$B = 650$		(1,6) (3,1) (6,3) (6,10) (8,13) (10,6) (10,13) (13,8) (13,10)	1 1 1 1 1 1 1 1 2	55.39 60.45 66.28 86.26 65.05 94.26 98.24 79.24 75.89
$B = 325$		(1,6) (6,1) (6,8) (6,10) (8,10) (8,13) (10,6) (10,13) (13,6) (13,8) (13,10)	1 1 1 1 1 1 1 1 1 1 2	79.25 89.37 100.00 84.00 44.09 100.00 100.00 99.96 100.00 77.32 81.42
$B = 206$		(3,5) (5,13) (10,3) (10,5) (5,10) (10,13) (13,5) (13,10)	1 1 1 1 2 2 2 3	54.65 98.12 73.06 85.48 100.00 98.88 97.82 100.00

Table 5 shows the changes in network configuration when varying the capacities of access arcs R, while the rest of the parameters remain fixed. From Table 5, we observe that as the capacity of links on access arcs decreases, the MHLP locates more hub facilities as well as more hub arcs to route the flow between all O/D nodes. For instance, when $R = 80$, the model opens two hubs at nodes 6 and 13 and activates only two hub arcs to route the flow between all O/D nodes. Moreover, the average hub arc utilization in the network is 86.45% which implies that higher flows through the hub arcs. Upon decreasing R to 50, the optimal solution recommends locating three fully interconnected hub facilities at nodes 5, 10, and 13. In this case, the average hub arc utilization is 89.35%. Further decreasing R to 25, the model opens six hubs at nodes 1, 3, 6, 8, 11, and 13 and activates 13 hub arcs to route the flow between all O/D nodes. Moreover, the average hub arc utilization in the network is 94.15% which implies that higher flows through the hub arcs. Furthermore, a lower capacity on access arcs leads to the selection of the most isolated node 3 as a hub given that it is the one with the highest amount of incoming/outgoing flow. It can also be seen that, as R decreases, the location of hub facilities tend to be closer to each other.

Table 6 illustrates the effects of varying the variable cost of hub arcs b on the optimal solution network while the capacities B and R and the variable cost p are fixed. As can be seen in Table 6, as the variable cost b decreases, the MHLP tends to locate more hub facilities and activate fewer hub arcs. This behavior can be attributed to the decrease in unit flow cost on hub arcs when b decreases. For example, when $b = 600$, the MHLP locates three fully interconnected hub facilities with an average hub arc utilization of 82.72%. Decreasing the unit flow cost by decreasing b to 500 results in locating four hub facilities at nodes 5, 8, 10 and 13, and activating 10 hub arcs. The average hub arc utilization is 91.81%. As expected, further decreasing the unit flow cost, i.e. b equal to 300, the optimal solution recommends locating six hub facilities at nodes 1,3,6,8,10, and 13 and using 13 hub arcs resulting in average utilization of 92.33%.

Finally, Table 7 illustrates the effects of varying the variable costs of access arcs p on the optimal solution networks while the rest of the parameters remain fixed. As can be

Table 4.5: Effect of varying capacity of access arcs in optimal solutions with $n = 20$, $B = 250$, $b = 200$, $p = 150$.

Capacity	Solution network	Hub arcs	y_{km}	% Utilization
$R = 80$		(13,6) (6,13)	4 3	92.14 80.83
$R = 50$		(13,10) (13,5) (10,13) (10,5) (5,13) (5,10)	3 2 1 1 2 1	78.13 87.00 100.00 100.00 71.90 99.08
$R = 25$		(13,11) (13,8) (13,6) (11,13) (11,6) (8,13) (8,1) (6,11) (6,8) (6,3) (3,1) (1,8) (1,6)	2 2 1 1 1 2 1 1 1 1 1 1 1	99.41 88.64 100.00 100.00 100.00 100.00 100.00 91.44 87.94 100.00 84.83 71.68 100.00

seen in this table, as the variable costs p decreases, the MHLP tends to locate fewer hubs and connect them with fewer hub arcs. For instance, when $p = 600$, MHLP selects five locations for hub facilities, i.e., 2, 6, 8, 10, and 13, and activates eight hub arcs to route the flow between all O/D pairs. The average hub arc utilization is 80.4%. Decreasing p to 317 results in locating only three hubs at nodes 5, 10, and 13 and connecting them using four hub arcs resulting in an average utilization of 90.06%. Further decreasing p to 212 leads to selecting only two hub facilities at nodes 6 and 13 with two hub arcs with an average utilization of 87.12%.

Table 4.6: Effect of varying the variable cost b of hub arcs on optimal solution networks with $n = 20$, $B = 250$, $R = 25$ and $p = 150$.

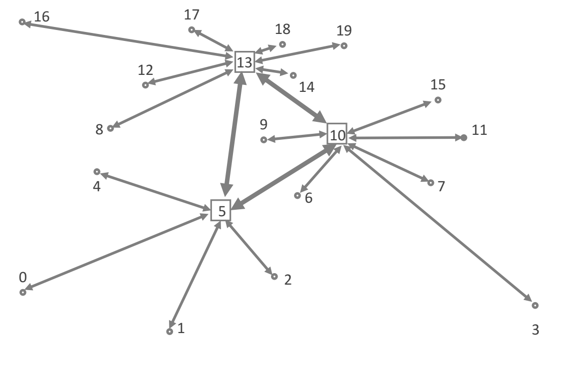
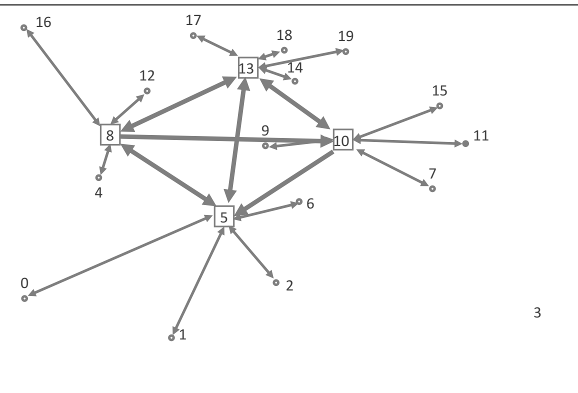
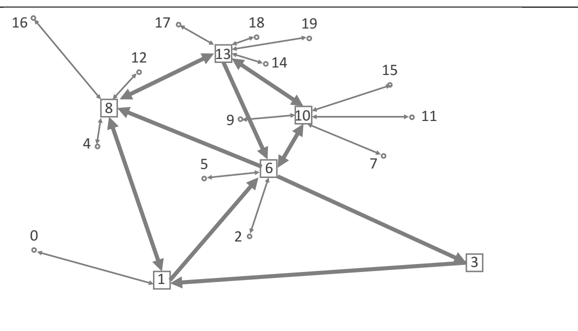
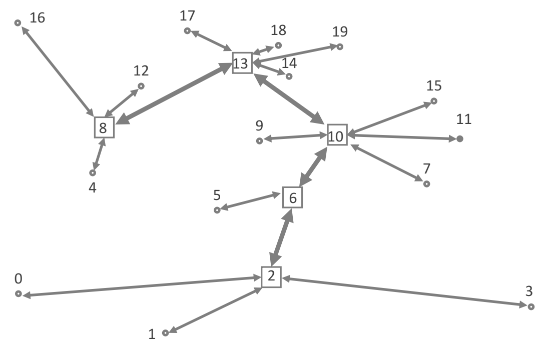
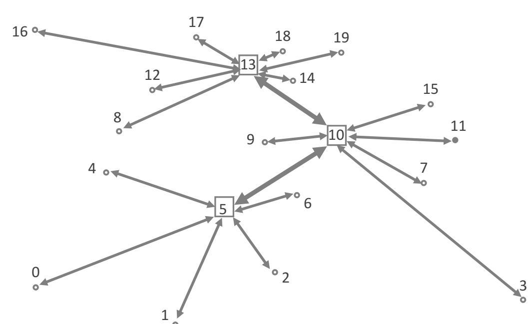
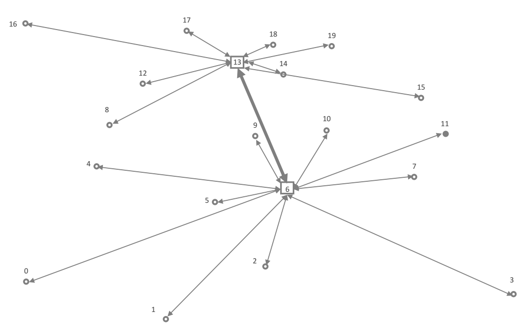
Cost	Solution network	Hub arcs	y_{km}	% Utilization
$b = 600$		(5,10)	1	100.00
		(5,13)	1	91.37
		(10,5)	1	65.82
		(10,13)	1	76.23
		(13,5)	2	80.31
		(13,10)	1	82.60
$b = 500$		(5,8)	1	84.00
		(5,10)	1	94.80
		(5,13)	1	93.59
		(8,5)	1	96.00
		(8,13)	1	100.00
		(10,5)	1	98.92
		(10,13)	1	77.74
		(13,5)	1	98.60
		(13,8)	2	74.42
		(13,10)	2	100.00
$b = 300$		(1,6)	1	97.50
		(1,8)	1	73.00
		(3,1)	1	83.60
		(6,3)	1	98.80
		(6,8)	1	87.78
		(6,10)	1	96.94
		(8,1)	1	100.00
		(8,13)	2	98.00
		(10,6)	1	78.53
		(10,13)	1	100.00
		(13,6)	1	100.00
		(13,8)	2	86.07
		(13,10)	2	100.00

Table 4.7: Effect of varying variable cost p of access arcs on optimal solution networks with $n = 20$, $B = 650$, $R = 110$ and $b = 750$.

Cost	Solution network	Hub arcs	y_{km}	% Utilization
$p = 600$		(2,6) (6,2) (6,10) (8,13) (10,6) (10,13) (13,8) (13,10)	1 1 1 1 1 1 1 2	61.73 82.59 86.26 65.05 94.25 98.24 79.24 75.89
$p = 317$		(5,10) (10,5) (10,13) (13,10)	1 1 1 2	88.00 100.00 93.69 78.54
$p = 212$		(6,13) (13,6)	1 2	100.00 74.24

Chapter 5

Dynamic Facility Location with Service Level Constraints

In this chapter, we study the *dynamic facility location problem with service level constraints* (DFLPSL). This problem seeks to locate a set of facilities with sufficient capacity through a planning horizon to satisfy customer demands at a minimum cost while a service level requirement for each customer is met. The service level constraints ensure that, for each customer in every time period t, the probability of meeting its demand in less than a predetermined amount of time (service level) is more than a threshold value θ. The proposed service level constraints capture two important sources of stochasticity in facility location by considering known probability distribution functions associated with processing and routing times. To the best of our knowledge, the service level constraints presented in this chapter are the first to consider the variability of travel time, service time and demands in a closed form expression.

The goal of this chapter is to present mathematical programming formulations and solution algorithms for the DFLPSL. It considers the stochastic aspects associated with modeling random demands, service times, and travel time. In order to obtain a closed form representation to the service level constraints, we analyze the stochastic behavior of our system by simplifying the probability density functions associated with the random quantities. The resulting nonlinear mixed-integer programming formulation is difficult to solve using commercial solvers, therefore, we develop two constructive heuristics and a local search heuristic to obtain feasible solutions. The first approach constructs feasible solutions using five deterministic sequential steps, whereas the second approach is based on a linear approximation of the service level constraints using subgradients. A series of computational experiments are performed to compare the efficiency of the two proposed solution algorithms.

The remainder of this chapter is organized as follows. Section 1 formally defines the problem and presents a nonlinear mixed integer programming formulation. In section 2, we present two heuristic algorithms for solving the problem. The computational results and analysis are presented in Section 3.

5.1 Problem Definition and Formulation

To formally define the problem, let $I \in \{1, 2, \ldots, i\}$ be the set of customers, $T = \{1, 2, \ldots, t\}$ be the set of time periods in the planning horizon, $J = \{1, 2, \ldots, j\}$ be the set of potential facility locations, and $K = \{0, 1, \ldots, k\}$ be the set of potential capacity levels for each facility. Let λ_i^t be the demand of customer i in time t and d_{ij} be the distance between customer i and facility j. Let c_{ijk}^t denote the unit flow cost of serving customer i from facility j with capacity level k at time period t. c_{ijk}^t is assumed proportional to the distance d_{ij}. The available capacity of the facility j of size k is given by μ_{jk}. Let $f_{jk_1k_2}$ represent the cost of changing the capacity level of a facility j from k_1 to k_2 at the beginning of period t and the cost to operate the facility at capacity level k_2 through the period t. Moreover, let k^j be the capacity level of an existing facility at location j. At the beginning of the planning horizon, k^j is 0 if no facility is opened at location j.

To develop our model, we extend the formulation of the dynamic facility location problem with generalized modular capacities proposed by Jena et al. (2015). We define the following sets of decision variables. Let x_{ik}^t be the fraction of the demand of customer $i \in I$ served by facility $j \in J$ at time period $t \in T$, and

$$y_{jk_1k_2}^t = \begin{cases} 1 & \text{if facility } j \text{ changes its capacity level from } k_1 \text{ to } k_2 \text{ at the beginning of period } t, \\ 0 & \text{otherwise.} \end{cases}$$

The DFLPSL can be stated as the following MNIP:

$$P1 \ \min \quad \sum_{j\in J}\sum_{k_1\in K}\sum_{k_2\in K}\sum_{t\in T} f_{jk_1k_2} y_{jk_1k_2}^t + \sum_{i\in I}\sum_{j\in J}\sum_{t\in T} \lambda_i^t c_{ij}^t x_{ij}^t$$

$$\text{s.t.} \quad \sum_{j\in J} x_{ij}^t = 1 \qquad \forall i \in I, t \in T \tag{34}$$

$$\sum_{i\in I} \lambda_i^t x_{ij}^t \leq \sum_{k1\in K}\sum_{k2\in K} \mu_{jk_2}^t y_{jk_1k_2}^t \qquad \forall j \in J, t \in T \tag{35}$$

$$\sum_{\substack{k1\in K\\ k1\neq 0}}\sum_{k2\in K} y_{jk_1k_2}^1 = 0 \qquad \forall j \in J \tag{36}$$

$$\sum_{k1\in K} y_{jk_1k_2}^{t-1} = \sum_{k1\in K} y_{jk_2k_1}^t \qquad \forall j \in J, t \in T \tag{37}$$

$$\sum_{k\in K} y_{j0k}^1 = 1 \qquad \forall j \in J \tag{38}$$

$$\mathcal{P}\{ \text{ waiting time at facility+travel time} \leq \tau\} \geq \theta \qquad \forall i \in I, j \in J, t \in T \tag{39}$$

$$x_{ij}^t \geq 0 \qquad \forall i \in I, j \in J, t \in T \tag{40}$$

$$y_{jk_1k_2}^t \in \{0,1\} \qquad \forall j \in J, k_1, k_2 \in K, t \in T \tag{41}$$

The first term of the objective function represents the setup cost for locating facilities as well as changing capacity levels between periods, whereas the second term represents the total transportation cost. Constraints (34) are the demand constraints for the customers. Constraints (35) are the capacity constraints at facilities. The set of constraints (36) guarantees that there is no facility at the beginning of the planning horizon. Constraints (37) link the capacity change variables in consecutive time periods. Constraint (38) ensure that only one capacity level must be chosen at the beginning of the planning horizon. Constraints (39) are service level constraints that ensure the probability of the response time of facility j to the demand node i within a τ units of time is at least θ. Constraints (40) and (41) are the non-negativity and integrality constraints.

We next describe the assumptions we consider for the service level constraints (39). We assume the demand of each customer i to be independent and to occur continuously over the time according to a Poisson process with a mean rate of λ_i^t. Since each facility serves a set of demand nodes, the aggregate demand at that facility is the sum of demand of the customers assigned to it, which can be described as another stochastic process that equals the sum of several Poisson processes Λ_j^t. To this end, we can write Λ_j^t in terms of the allocation decision variables x_{ij}^t as:

$$\Lambda_j^t(x) = \sum_{i \in I} \lambda_i^t x_{ij}^t.$$

Now, let us assume that service times at facilities are random variables that follow an exponential distribution with an expectation of $1/\mu_{jk}^t$, where μ_{jk}^t is the service rate of facility j that operates at capacity level k through period t. The service rate reflects the server capacity or the amount of demand a facility can process in a given time period. For each facility j in a given time t, the model selects only one capacity level from the discrete set of potential capacity levels, $\mu_{jk} = \{\mu_{j1}, \mu_{j2}, \ldots, \mu_{jK}\}$. Thus, each facility can be modeled as an $M/M/1$ queuing system, where the mean service rate of facility j if the capacity level k is selected, is given in terms of the locational decision variables y as:

$$\Omega_j^t(y) = \sum_{k_1} \sum_{k_2} \mu_{jk_2} y_{jk_1k_2}^t.$$

We consider the response time as the sum of the sojourn times at a facility (waiting in queue + service time) and the travel time on the link from facility j to demand node i. In order to write constraints (39) in terms of the decision variables, let $W_j^t(\Lambda_j^t(x), \Omega_j^t(y))$ denote the total time required to process demands at facility j in time t, and let the random variables T_{ij} represent the travel time on the link from facility j to demand node i. Then the service level constraints can be expressed as follows:

$$\mathcal{P}\{W_j^t(\Lambda_j^t(x), \Omega_j^t(y)) + T_{ij}x_{ij}^t \leq \tau\} \geq \theta \qquad \forall j \in J, i \in I, t \in T. \tag{42}$$

Constraints (42) state that the probability to fulfill a customer demand i in time t in less than τ unit of time is more than θ, where $\theta \in (0, 1)$. In order to obtain a closed form expression for service level constraints (42), we analyze the stochastic behavior of our system to simplify the probability density functions. Under a stationary process (i.e., $\Lambda_j^t(y) < \Omega_j^t(x)$) and first come first serve queuing discipline, the probability density function of the sojourn time w_j^t at facility j at time t is given by

$$f_X(w_j^t) = \begin{cases} [\Omega_j^t(y) - \Lambda_j^t(x)]e^{-[\Omega_j^t(y) - \Lambda_j^t(x)]w_j^t}, & w_j^t \geq 0;\ \Omega_j^t(y) > \Lambda_j^t(x) \quad \forall t, j; \\ 0, & \text{Otherwise.} \end{cases} \tag{43}$$

Polus (1979) observed that, travel time behavior on an arterial route is seen to closely follow a gamma distribution. Thus, for a given traveling path from facility j to demand node i in a road network, the travel time through it is modeled as $t_{ij}^t \sim \Gamma(\alpha, \beta_{ij}^t))$ with a probability density function of

$$f_Y(t_{ij}^t) = \begin{cases} \frac{(\beta_{ij}^t x_{ij}^t)^\alpha}{\Gamma(\alpha)} t_{ij}^{t(\alpha-1)} e^{-\beta_{ij}^t x_{ij}^t t_{ij}^t}, & t_{ij}^t \geq 0 \\ 0, & \text{otherwise}, \end{cases} \tag{44}$$

where $\beta_{ij}^t = \frac{E(t)}{Var(t)}$ and $\alpha = \frac{E^2(t)}{Var(t)}$ are the shape and scale parameters, respectively. Moreover, the coefficient of variation of the gamma distribution is given by

$$CV_t = \frac{SD(t)}{E(t)} = \frac{\sqrt{Var(t)}}{E(t)} = \frac{1}{\sqrt{\alpha}}.$$

The acumulative joint distribution function that combines sojourn and travel times can be stated as the sum of the two random variables (i.e., $Z_{ij}^t(x,y)$) (see Theorem 7.1 of Grinstead & Snell, 2012) as:

$$\mathcal{P}\{Z_{ij}^t(x,y) \leq \tau\} = \begin{cases} 1 - \frac{\Gamma(\alpha, \beta_{ij}^t x_{ij}^t \tau)}{\Gamma(\alpha)} - \frac{(\beta_{ij}^t x_{ij}^t)^\alpha}{(\omega_{ij}^t(x,y))^\alpha} e^{-(\Omega_j^t(y) - \Lambda_j^t(x))\tau} & [1 - \frac{\Gamma(\alpha, (\omega_{ij}^t(x,y))\tau)}{\Gamma(\alpha)}] \\ & if \omega_{ij}^t(x,y) > 0, \\ 1 - \frac{\Gamma(\alpha, \beta_{ij}^t x_{ij}^t \tau)}{\Gamma(\alpha)} & \text{Otherwise}, \end{cases} \tag{45}$$

where $\omega_{ij}^t(x,y) = \beta_{ij}^t x_{ij}^t + \Lambda_j^t(x) - \Omega_j^t(y)$ and $\Gamma(\alpha, \omega_{ij}^t(x,y)z)$ is the upper incomplete gamma function. We note that for integer values of α, the upper incomplete gamma function can be computed as:

$$\Gamma(\alpha, x) = (\alpha - 1)! e^{-x} \sum_{k=0}^{\alpha-1} \frac{x^k}{k!}.$$

5.2 Solution Algorithms

In this section, we present two different approaches to obtain feasible solutions for the DFLPSL. The first one is a constructive heuristic that uses a relaxation of the nonlinear integer model formulation to obtain an initial set of open facilities, and then sequentially extends the capacities of the opened facilities of the current solution until feasibility is achieved. The second approach is based on solving a linear integer programming formulation that approximates the service level constraints by a set of supporting hyperplanes that are tangent to constraints (45) at several points of the feasible region. Finally, a local search procedure is used to improve initial solutions obtained by the two techniques by exploring different types of neighborhoods.

5.2.1 Preprocessing Phase

One way of reducing the size of the formulation $(P1)$ is to perform some preprocessing capable of eliminating variables that are known not to take a strictly positive value in any optimal solution of the problem. The preprocessing works as follows. Assume that a facility $j \in J$ will only serve a demand node $m \in I$ in period t. Thus, $\Lambda_j^t(x) = \lambda_m^t x_{mj}^t$. Moreover, assume that the capacity of the facility j is set to the maximum possible level throughout the planning horizon t, i.e., $\mu_{jk_2} = \mu_{j|K|}$. We can say a node m can never be served from facility j if

$$\mathcal{P}\{Z_{mj}^t \leq \tau\} < \theta,$$

given that the solution would be infeasible. Therefore, for each $m \in I$, $j \in J$ and $t \in T$, we compute $\mathcal{P}\{Z_{mj}^t \leq \tau\}$ and set x_{mj} to zero if $\mathcal{P}\{Z_{mj}^t \leq \tau\} < \theta$.

5.2.2 Constructive Heuristic

A constructive heuristic is an iterative procedure that seeks to construct an initial feasible solution. The main idea in the constructive heuristic is to start with an empty or partial solution, then iteratively add new elements to the current solution until it finds a feasible solution. We propose a deterministic constructive procedure based on the solution of an MIP formulation. In particular, the procedure starts with the optimal solution obtained by solving the formulation $P1$ without including service level constraints (45) using a commercial solver. That is,

$$(P2)\ \min \sum_{j\in J}\sum_{k_1\in K}\sum_{k_2\in K}\sum_{t\in T} f_{jk_1k_2}y^t_{jk_1k_2} + \sum_{i\in I}\sum_{j\in J}\sum_{t\in T}\lambda^t_i c_{ij}x^t_{ij}$$

$$\text{s.t. } (34)-(38)(40)-(41).$$

Note that, the solutions of the relaxed formulation provide valid lower bounds which can be used later to evaluate the quality of solutions obtained in heuristic approaches. The optimal solution of $P2$ provides, for each period t, a set of facilities H^t with capacity levels k and allocation pattern M^t for demand nodes. In what follows, solutions are designated in the form $S = (H^t, M^t)$, where H^t represents the set of open facilities, i.e., $H^t(j,k) = 1$ if location $j \in J$ is selected to be a facility at time t with a capacity level k, and $M^t : I \to H^t$ represents the allocation scheme of customers at period t. That being said, If $j \in M^t(i)$ then customer i is assigned to facility j in time t.

The proposed heuristic algorithm constructs a feasible solution as follows. Let $(\bar{x}^t, \bar{y}^t)$ be the optimal solution of $P2$. $\hat{H}^t = \{(j,k_2) : \bar{y}^t_{jk_1k_2} = 1\}$ represents the current set of open facilities and $\bar{j} \in \hat{M}^t(i)$ where $\bar{x}^t_{i\bar{j}} > 0$ denotes the current allocation scheme. Note that, $(\bar{x}^t, \bar{y}^t)$ can be either a feasible solution for $P1$ that satisfies service level constraints (45), or still partial, i.e., infeasible solution. In the former case, the algorithm terminates with an optimal solution to the original problem $P1$. In the latter case, however, the proposed heuristic starts with the partial solution, by only considering the current set of open facilities and their capacity levels. To generate a feasible solution, we use the following four sequential steps:

(1) *Allocation scheme*: In the first step, the proposed algorithm tries to find a feasible solution using the current set of capacitated facilities $\hat{H}^t$ by finding a feasible allocation

scheme. Note that, the allocation scheme can be found independently for each time period t. To this end, we developed a greedy-based algorithm that allows demands to be served from the nearest available facility in a greedy fashion. Availability refers to facility ability to serve a demand node with its available capacity without violating service level constraints (45). To this end, for each i, the algorithm works by ordering x_{ij}^t variables such that

$$\lambda_i^t c_i(s) < \lambda_i^t c_i(s+1),$$

for $s = 1, \ldots, |\hat{H}|^t$, where $\lambda_i^t c_{i(s)}$ denotes the cost of serving the demand of node i from the $s^{(th)}$ ordered facility. For a given node i, starting from $s = 1$, the algorithm temporally fixes $x_{i(s)}^t$ to the remaining unserved demand $r^t(i)$, i.e., $r^t(i) = \lambda_i^t - \sum_{j \in \hat{H}^t \setminus \{\hat{H}^t(s)\}} \lambda_i^t x_{ij}^t$. The algorithm checks for the residual capacity $\Delta(s) = C_{max}^t(s) - \sum_{m \in I \setminus \{i\}} \lambda_m^t x_{m(s)}^t$, where $C_{max}^t(s)$ represents an upper limit on the total demand that can be served from a facility s in time t. To this end, if $\Delta(s)$ is more than $r^t(i)$, then $x_{i(s)} = \frac{r^t(i)}{\lambda_i^t}$. If $\Delta(s)$ is less than $r(i)^t$ and greater than zero, then $x_{i(s)}^t = \frac{\Delta(s)}{\lambda_i^t}$. Otherwise, $x_{i(s)}^t$ is equal to zero. Therefore,

$$x_{i(s)}^t = \begin{cases} min\{\frac{r^t(i)}{\lambda_i^t}, \frac{\Delta(s)}{\lambda_i^t}\} & \text{if } \Delta(s) > 0 \\ 0 & \text{Otherwise.} \end{cases}$$

The above procedure is repeated for the second ordered item (i.e., $s = 2$) until $\sum_{j \in \hat{H}^t} x_{ij}^t = 1$.

In order to compute $C_{max}^t(j)$, let $\hat{X}_j^t$ be the current set of demand nodes that are assigned to j in time t and satisfies service level constraints. Let $m \in I \setminus \hat{X}_j^t$ be the demand node that needs to be served from facility j, at time t. We temporally fix x_{mj}^t to $\frac{r^t(m)}{\lambda_m^t}$ and add it to the set $\hat{X}_j^t$. That is, $X_j'^t = \{\hat{X}_j^t \cup \{m\} : m \in I \setminus \hat{X}_j^t\}$. Then we check for violation in the service level constraints (45) for each demand node $i \in X_j'^t$. If the current allocation scheme satisfies constraints (45) then $C_{max}^t(j) = \mu_{jk_2}^t$.

However, if some demand nodes violate constraints (45), then we set $C^t_{max}(j) = \sum_{i \in X''^t_j} \lambda^t_i x^t_{ij}$ and iteratively decrement $C^t_{max}(j)$ by a constant δ until constraints (45) are satisfied. If the above procedure fails to find a feasible allocation scheme, then we go to step 2.

(2) *Increase capacities*: In this step of the algorithm, if the service level constraints in the current solution are violated for some customers i, capacities are increased sequentially at each time period according to the following procedures. A candidate list of facilities is sorted in decreasing order based on the overall number of customers assigned to each facility that violate constraints (45). Starting from the first ordered facility, we increase its capacity to the next level and check for any violation in constraints (45). If some demand nodes still violate constraints (45), we increment its capacity to the next level until constraints (45) are met or the maximum capacity level for this facility is reached. Note that, each time a capacity increases to the next level, step 1 is used to try to obtain a feasible allocation scheme, consequently the candidate list is updated. If the maximum capacity level of the current selected facility is reached without obtaining a feasible solution, we move to the next ordered facility. If this step fails to obtain a feasible solution, then we proceed to the next.

(3) *Facility relocation*: This step affects the current set of open facilities by closing an open facility at location $m \in H^t$ with a capacity level k_2 through the planning horizon t and locates a new facility at candidate location $j\prime \in J \backslash H^t$ with the same capacity level k_2 through the planning horizon t. The candidate facility location $j\prime$ is selected without a specific order. Note that, at each iteration, step 1 is used to find a feasible allocation scheme. If this step fails to provide a feasible solution, then we go to step 4.

(4) *Locate a new facility*: If the above steps fail to find a feasible solution for the original

problem, the algorithm locates a new facility at candidate location $j\prime \in J\backslash H^t$ with a capacity level $k_2 = 1$ through the planning horizon t. Once the new facility added to the current solution we go back to step 1.

5.2.3 Branch-and-Cut Heuristic

The second solution approach for the DFLPSL is based on approximating the nonlinear constraints (45) with a set of linear inequalities. These are valid only at local points of the feasible region, that is, not necessarily global valid inequalities. The basic idea is to solve a relaxed version of the nonlinear formulation by iteratively adding a set of linear constraints only when they are violated by the current integer solution.

For simplicity, let $F(x, y, \tau) = \mathcal{P}\{Z^t_{mj} \leq \tau\}$. To derive the set of linear constraints, let $(\bar{y}, \bar{x})$ be the current integer solution that violates constraints (45), that is $F(\bar{x}, \bar{y}, \tau) < \theta$ for some i in time t. Suppose that F is a concave function that satisfies

$$F(x, y, \tau) \leq \bigtriangledown^T(\bar{x})(x - \bar{x}) + \bigtriangledown^T(\bar{y})(y - \bar{y}) + F(\bar{x}, \bar{y}, \tau),$$

for all y and x, where $\bigtriangledown(\bar{y})$ and $\bigtriangledown(\bar{x})$ is any subgradient of $F(x, y, \tau)$ at $\bar{y}$ and $\bar{x}$, respectively. Now, we want $F(x, y, \tau) \geq \theta$, thus we should have

$$\theta \leq F(x, y, \tau) \leq \bigtriangledown^T(\bar{x})(x - \bar{x}) + \bigtriangledown^T(\bar{y})(y - \bar{y}) + F(\bar{x}, \bar{y}, \tau).$$

Therefore, we can write the set of linear constraints as

$$\bigtriangledown^T(\bar{x})(x - \bar{x}) + \bigtriangledown^T(\bar{y})(y - \bar{y}) + F(\bar{x}, \bar{y}, \tau) \geq \theta. \tag{46}$$

The addition of cuts (46) removes the current solution $(\bar{x}, \bar{y})$ from the set of feasible solutions without removing any solution that is feasible for the original problem $P1$ if and only if the concavity assumption of $F(x, y, \tau)$ holds. If $F(x, y, \tau)$, however, is not a concave function, then we may cut out feasible solutions of the $P1$, including the optimal one. Unfortunately, in our case such assumption does not hold and adding these inequalities may have the undesirable effect of excluding part of the feasible region, which may contain the optimal solution. As we will show in Section 3.2, this approach provides promising feasible solutions which can be further improved by using local search methods.

We use the finite difference method to compute the partial derivatives of $F(x, y, \tau)$. The partial derivatives via the finite difference method can be computed using forward, backward or central differences. We evaluate the partial derivative $\bigtriangledown(\bar{x})$ at $\bar{x}$ as follows. We evaluate the function F_Z at $(\bar{x}, \bar{y})$ and at $((\bar{x} - he_{ij}^t), \bar{y})$ for $i = 1 \ldots I$; $j = 1 \ldots, J$ and $t \ldots T$, where e_{ij}^t is a unit matrix with a one in position i, j and t and zero elsewhere and $d\Lambda \in \{0, 1\}$ represents the incremental rate of the arrival rate (i.e., step size). We compute the partial derivatives, using backward finite difference method, as:

$$\bigtriangledown(\bar{x}_{ij}^t) = \frac{F_z(\bar{x}, \bar{y}, \tau) - F_z((\bar{x} - d\Lambda e_{ij}^t), \bar{y}, \tau)}{d\Lambda}$$

if $\bar{x}_{ij}^t - d\Lambda e_{ij}^t > 0$. However, when $\bar{x}_{ij}^t - d\Lambda e_{ij}^t < 0$, then the corresponding partial derivatives are computed using the forward difference method as:

$$\bigtriangledown(\bar{x}_{ij}^t) = \frac{F_z((\bar{x} + d\Lambda e_{ij}^t), \bar{y}, \tau) - F_z(\bar{x}, \bar{y}, \tau)}{d\Lambda}$$

Finally, to compute $\bigtriangledown(\bar{y})$ at $\bar{y}$, we use forward finite difference as follows,

$$\bigtriangledown(\bar{y}_{jk_1k_2}^t) = \frac{F_z(\bar{x}, (\bar{y} + d\mu M_{jk_1k_2}^t), \tau) - F_Z(\bar{x}, \bar{y}, \tau)}{d\mu}$$

where $d\mu$ represents the step size or the incremental change of the service rate and $M_{jk_1k_2}^t$ is a unit matrix with a one in position $(j, k1, k_2, t)$ and zero elsewhere.

5.2.4 Local Search

The local search is an iterative procedure that moves from one solution to another according to some neighborhood structure. Our local search procedure explores two types of neighborhoods. The first type affects the current capacity level of facilities by considering two neighborhoods, namely, increase and decrease capacity levels. The increase capacity level neighborhood $\aleph_1$ improves the current solution by allowing some facilities to take higher capacity level so that a trade-off between location and transportation costs is achieved while ensuring a minimum service level requirement for each customer. To this end, starting from $t = 1$, for each facility $j \in H^t$ with capacity level k_2, we perform the following steps: (i) if $k_2 < |K|$, increases the capacity level of facility $j \in H^t$ from $k2$ to $k2 + 1$; (ii) find a new transportation scheme using step 1 explained in section 2.2; (iii) If the total cost of the new solution is better than the best known solution, set the capacity level of j to $k_2 + 1$ and go back to step (i); (iv) if the new solution is worse than the best known solution then increment t by one and go back to step (i). Let $S = (H^t, M^t)$ be the current solution then

$$\aleph_2 = \{S' = (H'^t, M'^t) : S' = H^t \backslash \{(j, k_2)\} \cup \{(j, (k_2 + 1))\}, k_2 \leq |K|\}$$

Decrease capacity neighborhood $\aleph_2$ allows some facilities, through the planning horizon t, to take lower capacity levels while service level constraints are met. Starting from a facility $j \in H^t$ with a capacity level k_2, k_2 is decreased sequentially at each time period t until the solution become infeasible or $k_2 = 0$. Let $S = (H^t, M^t)$ be the current solution, then

$$\aleph_2 = \{S' = (H'^t, M'^t) : S'^t = H^t \backslash \{(j, k_2)\} \cup \{(j, (k_2 - 1))\}, k_2 \geq 0\}$$

In this neighborhood structure, a best improvement strategy is considered. Note that, facilities are considered without a specific order.

The second type of neighborhoods affects the current set of open facilities by interchanging an open facility by a closed one. Let $S = (H^t, M^t)$ be the current solution and let $j_1 \in J \backslash H^t$ be the candidate location to replace the open facility located at site $j_2 \in H^t$ with capacity level k_2, then

$$\aleph_3 = \{S' = (H'^t, M'^t) : S'^t = H^t \backslash \{(j_2, k2)\} \cup \{(j_1, k_2)\}, j_2 \in S', j_1 \in J \backslash H^t\}.$$

In this neighborhood, a best improvement strategy is also considered. An outline of the local search scheme is depicted in Algorithm 5.2.

Algorithm 5.2: Local search procedure

```
stoppingcriteria ← false
while( stoppingcriteria = false)do
    explore ℵ1
    if(solution not improved in ℵ1) then
        explore ℵ2
        if(solution not improved in ℵ2) then
            explore ℵ3
            if(solution has not been updated) then
                stoppingcriteria ← true
            end-if
        end-if
    end-if
end-while
```

5.3 Computational Experiments

We performed computational experiments to compare and analyze the performance of the two heuristic algorithms. The formulation and solution algorithms were coded in C++ and ran on an HP station with an Intel Xeon CPU E3-1240V2 processor at 3.40GHz and 24 GB of RAM under windows 7 environment. MIP problems were solved using Concert technology of CPLEX 12.6.2. We have used the 2000 census data (refer to Daskin, 1995) to generate a set of benchmark instances for the DFLPSL. This data set consists of the resident population of 150 large cities in the continental United States. For generating instances, we identified seven relevant factors in the design of our experiments, namely the number of customers ($|I| \in \{30, 50, 70, 90, 110, 130\}$), the number of potential facility locations ($|J| = 20$), the number of planning horizons ($|T| = 4$), the number of capacity levels ($|K| = 5$), service level parameter ($\theta \in \{0.8, 0.85, 0.9, 0.95$) and the promised response time ($\tau \in \{1, 1.25, 1.5\}$). At the first time period of the planning horizon ($i.e., t = 1$), the mean customer demand rates λ_i^1 are obtained by dividing the population of those cities by 1000. For $t > 1$, we note that both downward and upward demand fluctuation are possible over the time horizon. In particular, changes are allowed up to ± 20 between two consecutive time periods. That is, $\lambda_i^t = (1+\pi_i^t)\lambda_i^{t-1}$, where $\pi_i^t = U[-0.2, 0.2]$. The service rate μ_{jk} of a facility at location j with capacity level k is set to $1.1 \times \zeta_k \frac{\sum_{i\in I}\sum_{t\in T}\lambda_i^t}{|T|}$, where $\zeta_k = 0.2, 0.5, 0.5, 0.75, 1$ with 5 capacity levels. The unit transportation costs c_{ij} have been computed based on the Euclidean distance between the points. That is, $c_{ij} = \rho d_{ij}$, where d_{ij} represents the distance between nodes i and j and $\rho \in \{0.01, 0.3, 1.25\}$ denotes the transportation cost structure (i.e., low, moderate, high). The aggregate costs $f_{jk_1k_2}^t$ capture economies of scale by taking into account four different costs as follows:

(1) Fixed opening cost FO_j of a new facility at candidate location j. The fixed opening cost is set to the 2015 median home value in the city at location j.

(2) The operating costs of a facility j at capacity level k throughout time t. Operating costs are set to $5\mu_{jk}$.

(3) The cost of increasing the capacity level of a facility j from level k_1 to level k_2. The capacity expansion cost is set to $10(\mu_{jk_2} - \mu_{jk_1})$.

(4) The cost of reducing the capacity level of a facility j from level k_1 to level k_2. The capacity reduction cost is set to $15(\mu_{jk_1} - \mu_{jk_2})$.

By combining the aforementioned costs, the aggregate cost $f_{jk_1k_2}^t$ can be expressed as:

$$f^t_{jk_1k_2} = \begin{cases} FO_j + 10(\mu_{jk_2} - \mu_{jk1}) + 5\mu_{jk_2} & \text{if } k_2 > k_1, k_2 \neq 0 \ , \\ FO_j + 15(\mu_{jk_1} - \mu_{jk_2}) + 5\mu_{jk_2} & \text{if } k_2 < k_1, k_1 \neq 0 \ , \\ FO_j + 5\mu_{jk_2} & \text{if } k_1 = k_2, k_1 \neq 0 \ . \end{cases}$$

5.3.1 Sensitivity Analysis

In the first part of the experiments, we performed a sensitivity analysis on some of the parameters of the problem to analyze the changes in solution structure and the objective values. Since optimal solutions are not known, we used best known solutions to complete this section. Through these experiments, we considered instances with $|I| = 30$, $|T| = 1$, $\tau = 1$ and $|K| = 5$. Note that, when $T = 1$ and $\theta = 0\%$, the problem reduces to a *facility location problem with multiple capacity levels* (FLPCL). Figure 5.1 and 5.2 show the percent total cost increases $\%Cost$ between DFLPSL and FLPCL when varying the desired service level θ and promised response time τ, respectively. The $\%Cost$ is computed as $\frac{DFLPSL\ total\ cost - FLPCL\ total\ cost}{DFLPSL\ total\ cost} \times 100$.

Figures 5.3-5.6 show how the network configurations change depending on: (i) the variable costs structure, (ii) the desired service level θ, (iii) the travel time coefficient of variation CV, (iv) the promised response time τ. In each figure, we report some information of the designed system such as set of open facilities (*Facility*), selected capacity level at each open facility (*Level*) and the percent facility utilization ($\%UTILIZ$). The percent utilization measures how much of the available capacity is being used on each facility and computed as $(\sum_{i\in I} \lambda^t_i \bar{x}^t_{ij}/\mu^t_{jk2}\bar{y}^t_{jk1k2}) \times 100$, where $(\bar{x}, \bar{y})$ denotes the best known solution.

In figure 5.1, we let θ vary from 40% to 99% and observe its impact on the total cost. At the same time, we select three values for the response time, i.e., $\tau = 1, 1.25$ and 1.5 days, fix CV to 0.5 and select moderate transportation cost structure (i.e., $\rho = 0.3$). As the service level parameter increases, the $\%Cost$ increases. The main reason for this is that as the θ increases, there is a need to open more facilities (possibly relocate facilities to more expansive locations) to react faster to demands. However, more facilities lead to dispersion concept which results in shorter response time and higher service level. Figure 5.1 also shows that the higher value of τ, the lower impact on the total costs. For instance, at $\theta = 92\%$, when $\tau = 1.5$ days, the percent increase in the total cost is 7.57%, in contrasts with $\tau = 1$ where the increase is 44%.

In figure 5.1, we let θ vary from 40% to 99% and observe its impact on the total cost. At the same time, we select three values for the response time, i.e., $\tau = 1, 1.25$ and 1.5 days, fix CV to 0.5 and select moderate transportation cost structure (i.e., $\rho = 0.3$). As the service level parameter increases, the percentage cost difference between FLPCL and FLPCL *with service level* solutions increases. The main reason for this is that as the θ increases, there is a need to open more facilities (possibly relocate facilities to more expansive locations) to react faster to demands. However, more facilities lead to dispersion concept which results in shorter response time and higher service level. Figure 5.1 also shows that the higher value of τ, the lower impact on the total costs. For instance, at $\theta = 92\%$, when $\tau = 1.5$ days, the percent increase in the total cost is 7.57%, in contrasts with $\tau = 1$ where the increase is 44%.

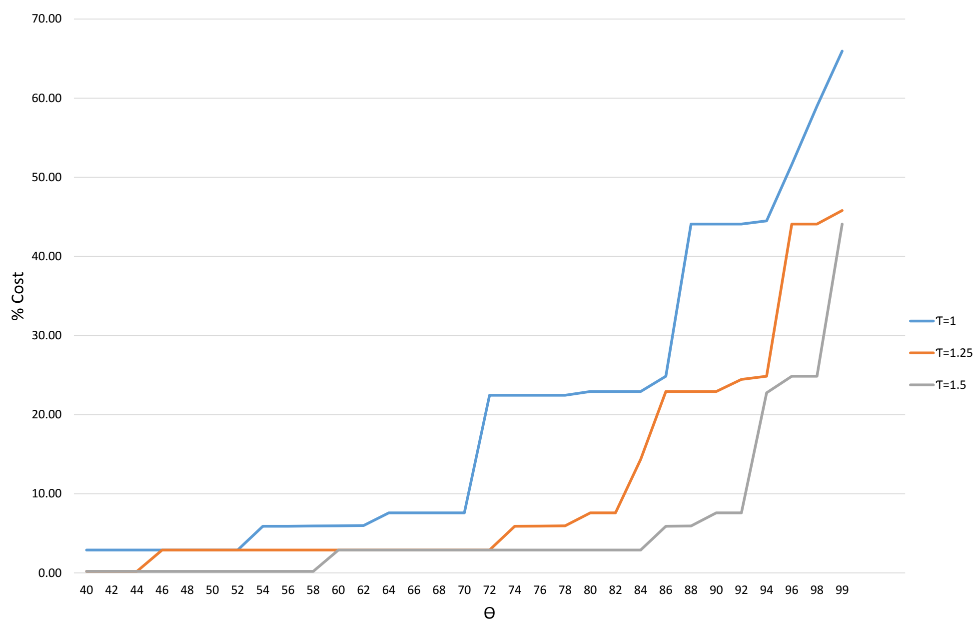


Figure 5.1: %Cost corresponding to different θ

Figure 5.2, illustrates the changes in the objective function values when varying the promised response time τ. In particular, we let τ changes from 0.9 to 2.75 days and observe the impact on the total costs. Moreover, for each τ, we select three values to service level parameter $\theta \in \{0.85, 0.90, 0.95\}$. Also, we fix CV to 0.5 and select moderate transportation cost structure($\rho = 0.3$). As can be seen in the figure, as τ increases, the overall cost decreases. That is, promising a short response time to customers leads to either locate more facilities, relocate some facilities to more expansive sites and\or increase capacity levels. Note that, when $\tau > 2.3$ and $\theta < 85\%$, the objective value slightly changed, thus there is no need to make longer response time decisions.

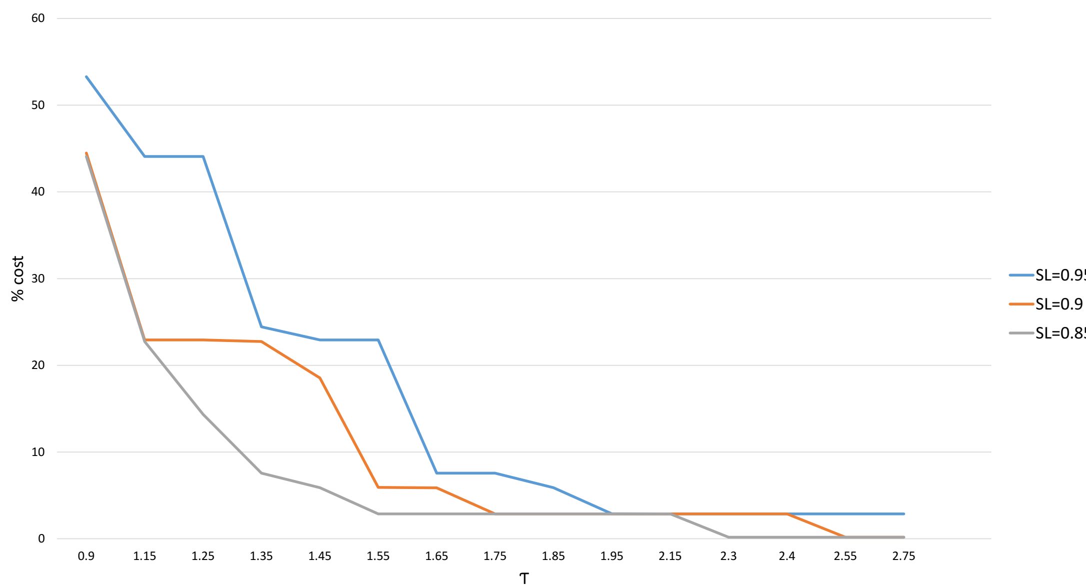
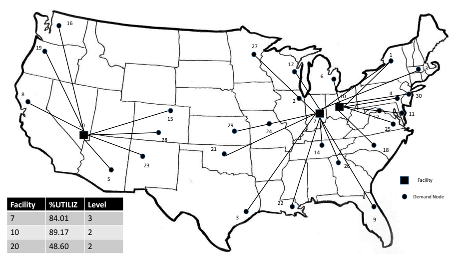


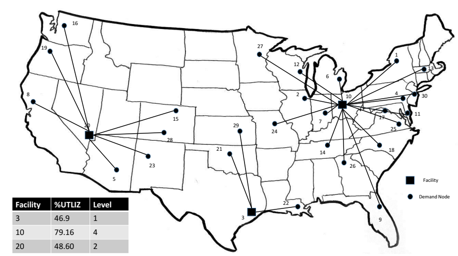
Figure 5.2: %Cost corresponding to different τ

Figure 5.3 shows the characteristics of the network when considering different transportation costs structures ρ (i.e., high, moderate and low). Moreover, we fix θ to 85%, τ to 1 day and CV to 0.5. As expected, the trade-off between transportation costs, facilities fixed costs and promised response time would determine the optimal number of facilities. In particular, a high transportation cost leads to more facilities in the network enabling facilities to occur closer to demand nodes. In the case of low transportation costs structure, however, it seems that the reduction in transportation costs by adding extra facilities is less than their fixed costs. For instance, when the transportation costs are low compared to the fixed costs, the model locates three facilities at locations $7, 10$ and 20. The transportation costs account for 13% of the total cost, and the fixed costs account for 87.0% of the total cost. Moreover, facility utilizations are 84.01%, 89.17% and 48.60%, respectively.

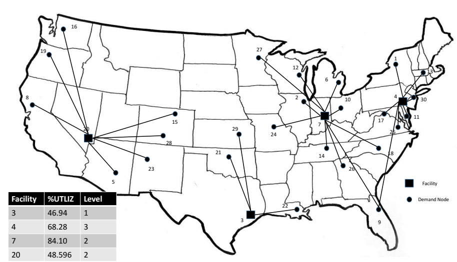
In the case of moderate transportation costs structure, Figure 5.3(b), the model relocates the facility at location 7 to 3 and selects the first capacity level to serve its demand with a 46.9% utilization. Moreover capacity level at location 10 increased to level 4, and facility utilization decreases to 79.16%. Furthermore, the transportation costs account for 30.9% of the total costs, whereas the fixed costs account for 69.1% of the total costs. Finally, Figure 5.3(c) shows that high transportation costs structure lead to locate four facilities and select capacity levels; 1 for location 3, 3 for location 4, 2 for location 7 and 2 for location 20 with utilizations of 46.94%, 68.28%, 84.10% and 48.56%, respectively. Note that, In the high transportation costs structure, the fixed costs account for 54.40% of the total costs, while the variable costs account for 45.60%.



(a) Low variable costs. 87.0%fixed costs



(b) Moderate variable costs. 69.1%fixed costs

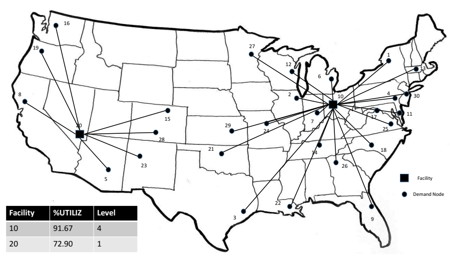


(c) high variable costs. 54.40% fixed cost

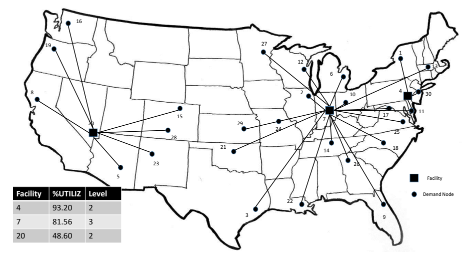
Figure 5.3: Network structures with different variable costs and service level of 85%.

Figure 5.4 presents the characteristics of the network when considering different service level parameter (θ's) while the rest of the parameters remain fixed (i.e., $\tau = 1, CV = 0.5$, and $\rho = 0.3$). As can be seen in the figure, Increasing θ leads to increase not only the number of open facilities, but also reduces facilities utilizations. For instance, Figure 5.4(a) shows the optimal solution structure to FLPCL, the model locates two facilities at locations 10 and 20 with utilization of 91.67% and 72.90%, respectively. The total cost is

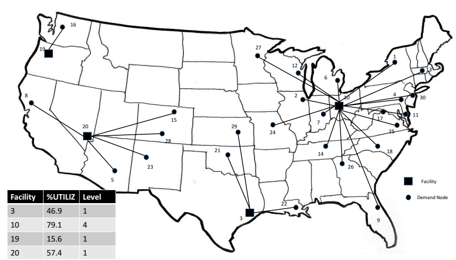
$1,307,624.183. Incurring 85% service level, increases the total cost to $1,693,391.234 ($22.78\%$ increase) by adding an additional facility at location 7 with 93.20% utilization and relocating facility from location 10 to 7 with 81.56% utilization. The utilization of facility 20 is reduced to 48.6%. When θ is selected to be 90%, the best known solution selects four facilities at locations $3, 10, 19$ and 20 with an increase in the total cost of 27.3%. Moreover, facilities utilization are 46.9%, 79.1%, 15.6% and 57.4%, respectively. Finally, when the desired service level is 95%, we see the facility at location 20 shifted to location 5 with 46.2% utilization and the total cost increases by 27.6% and facilities utilizations are 46.9% for 3, 79.2% for 10 and 26.7% for 19.



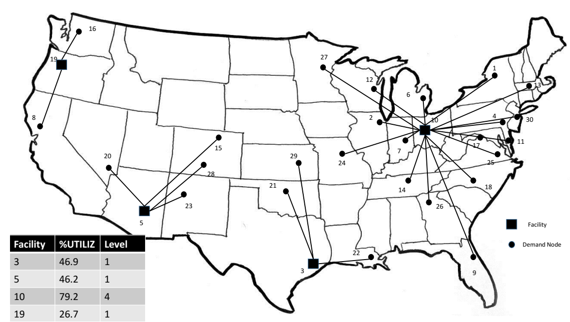
(a) $\theta = 0\%$; total cost is $1,307,624.183



(b) $= \theta = 85\%$; total cost is $1,693,391.234



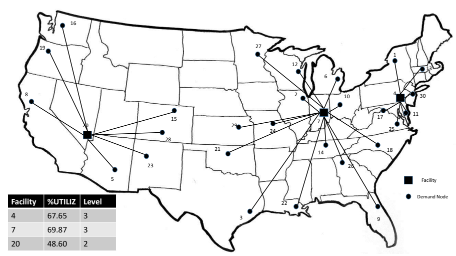
(c) $\theta = 90\%$; total cost is $1,804,333.139



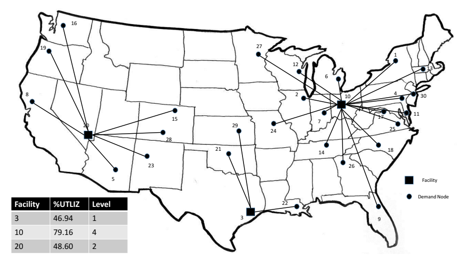
(d) $\theta = 95\%$; total cost is $1,806,182.53

Figure 5.4: Network structures with different service levels, $\tau = 1$ and $CV = 0.5$

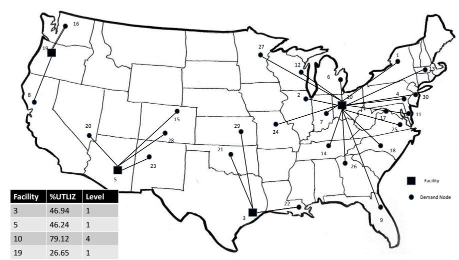
Figure 5.5 shows the network structure when considering different coefficients of variations ($CV's$) for the travel time. In particular, we consider $CV \in \{0.3, 0.4, 0.5\}$ to represent different variability levels in the travel time. Moreover, we fix the service level parameter to 90% and τ to 1 day. As can be seen in the figure, the more variability in the travel time, the more facilities will open. In the case of a lower variability level $CV = 0.3$, the best-known solution, a set of facilities are located at locations 4, 7 and 20 with capacity levels of 3, 2 and 2 respectively. However, for a moderate variability level, $CV = 0.4$ the set of facilities changes to become nodes 3, 10 and 20 with capacity levels of 1, 4 and 2, respectively. In the case of higher variability level $CV = 0.5$, the set of facilities becomes $\{3, 5, 10, 19\}$ with capacity levels of 1,1,4 and 1 respectively.



(a) $CV = 0.3$



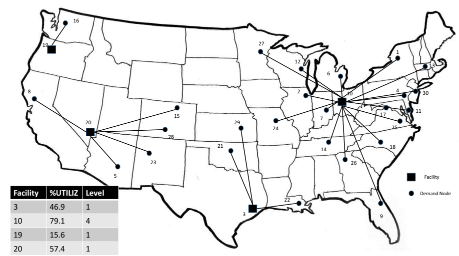
(b) $CV = 0.4$



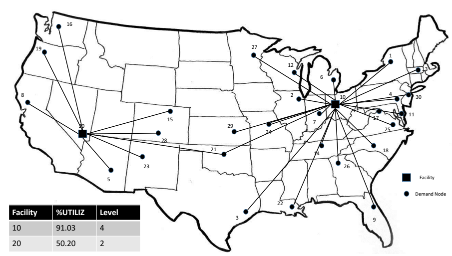
(c) $CV = 0.5$

Figure 5.5: Network structures with different CVs and service level= 90%.

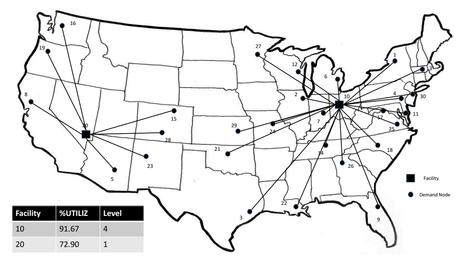
Finally, Figure 5.6 shows the changes in network configuration upon varying the response time τ, while the rest of the parameters remain fixed. From Figure 5.6, we observe that as the response time increases, the model locates fewer facilities to serve demands. For example, when $\tau = 1$, the model opens four facilities at locations $3, 10, 19$ and 20 and selects capacity levels $1, 4, 1, 1$, respectively. Moreover, the facilities utilizations are 46.9%, 79.1%, 15.6% and 57.4%, respectively. Upon increasing τ to 1.25, the model recommends closing facilities at locations 3 and 19 and increasing capacity level of the facility at location 2 to level 2. However, the utilization of facility 10 increases to 91.03%, while facility at location 20 utilization increases to 50.20%. Finally, when $\tau = 1.5$, the model recommends keeping facilities at locations 10 and 20, but decreasing capacity level at facility 20 to level 1. Furthermore, the utilization of facility 20 increases to 72.90%.



(a) τ=1



(b) τ=1.25



(c) τ=1.5

Figure 5.6: Network structures with different promised response times τ, $\theta = 90\%$ and $CV = 0.5$

5.3.2 Comparison of Solution Algorithms

The second set of computational experiment is performed to compare the performance of the two heuristic algorithms. The detailed results of this comparison on a set of 72 instances ranging from 30 to 130 demand nodes, 20 potential facility locations, 5 capacity levels and 4 time periods are reported in Table 5.1. The first column provides the number of demand nodes, $|I|$, desired service level, θ, and promised response time τ ($|I| - \theta - \tau$).

The second column $\%Fix$ reports the percentage of allocation variable fix with the pre-processing phase. That is, this column is computed as $\frac{FA}{I*J*T} \times 100$, where FA is the number of eliminated allocation variables fixed through the planning horizon. The next set of columns reports the CPU time in seconds (CPU), the percent deviation with respect to the best known solution obtained across the two heuristic algorithms, and the percent deviation between final upper and lower bounds ($\%GAP$). The percent deviation ($\%DEV$) is computed as $\frac{best\ known\ solution - huristic\ final\ solution}{huristic\ final\ solution} \times 100\%$. The optimality gap ($GAP\%$) is computed as $\frac{best\ known\ solution - LB}{best\ known\ solution} \times 100\%$, where LB denotes the lower bound obtained by solving the relaxed integer programming formulation, i.e., without service level constraints and with variables elimination test.

Table 5.1 summarizes the performance of the two heuristic algorithms. The first column provides the number of demand nodes, $|I|$. The next set of columns under heading $HEUR1$ reports, for each $|I|$, the average CPU time in seconds, the average percent deviation with respect to the best known solution $(\%DEV)$ and the number of best solutions found $\#Best$ on the 72 considered instances.

Table 5.1: Performance comparison between HEUR1 and HEUR2

Name	Var	$HEUR1$			$HEUR2$		
$\|I\| - \theta - \tau$	$\%Fix$	CPU	$\%DEV$	$\%GAP$	CPU	$\%DEV$	$\%GAP$
$30 - 0.80 - 1.00$	47.33	246	0.00	19.06	189	1.99	20.67
$30 - 0.85 - 1.00$	52.50	122	0.00	19.06	100	2.26	20.89
$30 - 0.90 - 1.00$	57.00	231	0.00	1.93	158	0.00	1.93
$30 - 0.95 - 1.00$	62.00	232	1.10	3.67	156	0.00	2.61
$30 - 0.80 - 1.25$	37.00	104	0.00	4.74	54	0.00	4.74
$30 - 0.85 - 1.25$	40.33	99	0.00	4.74	154	0.00	4.74
$30 - 0.90 - 1.25$	44.83	196	0.00	19.65	202	0.53	20.08
$30 - 0.95 - 1.25$	53.67	237	0.00	1.93	157	0.00	1.93
$30 - 0.80 - 1.50$	30.17	1	0.00	0.00	56	0.00	0.00
$30 - 0.85 - 1.50$	34.67	65	0.00	3.02	58	0.00	3.02
$30 - 0.90 - 1.50$	37.17	98	0.00	4.74	57	0.00	4.74
$30 - 0.95 - 1.50$	42.50	106	0.00	20.05	135	3.64	22.96
$50 - 0.80 - 1.00$	50.60	147	0.00	3.39	134	0.00	3.39
$50 - 0.85 - 1.00$	55.60	234	0.00	10.97	275	2.90	13.55
$50 - 0.90 - 1.00$	59.70	405	0.00	8.50	449	2.69	10.96
$50 - 0.95 - 1.00$	64.20	815	0.00	9.74	687	0.00	9.74
$50 - 0.80 - 1.25$	40.20	155	0.00	3.59	144	0.00	3.59
$50 - 0.85 - 1.25$	43.50	145	0.00	3.59	135	0.00	3.59
$50 - 0.90 - 1.25$	48.00	140	0.00	3.59	127	0.00	3.59
$50 - 0.95 - 1.25$	56.50	432	0.00	9.97	421	0.34	10.27
$50 - 0.80 - 1.50$	33.80	162	0.00	3.59	150	0.00	3.59
$50 - 0.85 - 1.50$	37.90	156	0.00	3.59	147	0.00	3.59
$50 - 0.90 - 1.50$	40.40	154	0.00	3.59	142	0.00	3.59
$50 - 0.95 - 1.50$	46.00	139	0.00	3.59	127	0.00	3.59
$70 - 0.80 - 1.00$	50.93	240	0.00	3.38	223	0.00	3.38
$70 - 0.85 - 1.00$	55.79	379	0.00	6.46	347	0.00	6.46
$70 - 0.90 - 1.00$	60.86	836	2.27	7.63	708	0.00	5.53
$70 - 0.95 - 1.00$	65.43	943	0.00	9.27	1032	0.00	9.27
$70 - 0.80 - 1.25$	40.43	246	0.00	3.69	230	0.00	3.69
$70 - 0.85 - 1.25$	43.71	240	0.75	3.62	204	0.00	2.89
$70 - 0.90 - 1.25$	48.29	234	0.00	3.38	213	0.00	3.38
$70 - 0.95 - 1.25$	56.93	598	2.02	4.93	524	0.00	3.00
$70 - 0.80 - 1.50$	33.86	252	0.00	3.69	245	0.00	3.69
$70 - 0.85 - 1.50$	38.07	248	0.00	3.69	247	0.00	3.69
$70 - 0.90 - 1.50$	40.57	241	0.00	3.69	224	0.00	3.69
$70 - 0.95 - 1.50$	46.36	231	0.00	3.38	213	0.00	3.38

Table 5.1 continued

Name	Var	$HEUR1$			$HEUR2$		
$\|I\| - \theta - \tau$	$\%Fix$	CPU	$\%DEV$	$\%GAP$	CPU	$\%DEV$	$\%GAP$
$90 - 0.80 - 1.00$	52.39	606	4.59	7.40	298	0.00	3.15
$90 - 0.85 - 1.00$	57.11	529	0.00	3.33	486	0.00	3.33
$90 - 0.90 - 1.00$	61.94	1112	1.87	6.57	1122	0.00	4.82
$90 - 0.95 - 1.00$	66.28	1151	1.33	8.18	1222	0.00	6.96
$90 - 0.80 - 1.25$	42.56	348	0.00	3.38	324	0.00	3.38
$90 - 0.85 - 1.25$	45.61	333	0.38	3.44	287	0.00	3.08
$90 - 0.90 - 1.25$	49.94	624	8.89	11.05	289	0.00	3.15
$90 - 0.95 - 1.25$	58.11	825	2.06	5.12	734	0.00	3.17
$90 - 0.80 - 1.50$	36.22	343	0.00	3.38	320	0.00	3.38
$90 - 0.85 - 1.50$	40.33	338	0.00	3.38	313	0.00	3.38
$90 - 0.90 - 1.50$	42.72	328	0.00	3.38	308	0.00	3.38
$90 - 0.95 - 1.50$	48.17	307	0.00	3.22	294	0.00	3.22
$110 - 0.80 - 1.00$	51.50	836	3.95	8.38	439	0.00	4.76
$110 - 0.85 - 1.00$	56.32	689	0.00	3.41	678	0.00	3.41
$110 - 0.90 - 1.00$	61.18	1458	0.00	7.11	1381	0.63	7.69
$110 - 0.95 - 1.00$	66.27	1802	1.15	8.71	1641	0.00	7.66
$110 - 0.80 - 1.25$	41.00	454	0.00	3.76	447	0.00	3.76
$110 - 0.85 - 1.25$	44.23	434	0.20	3.74	420	0.00	3.54
$110 - 0.90 - 1.25$	49.05	674	3.37	7.86	409	0.00	4.76
$110 - 0.95 - 1.25$	57.55	1230	2.15	5.33	1049	0.00	3.30
$110 - 0.80 - 1.50$	34.23	490	0.00	3.76	441	0.00	3.76
$110 - 0.85 - 1.50$	38.59	472	0.00	3.76	450	0.00	3.76
$110 - 0.90 - 1.50$	41.18	464	0.00	3.76	447	0.00	3.76
$110 - 0.95 - 1.50$	47.18	701	3.36	7.86	420	0.00	4.76
$130 - 0.80 - 1.00$	51.58	918	2.94	7.95	579	0.00	5.25
$130 - 0.85 - 1.00$	56.38	1282	0.00	11.66	1407	0.89	12.45
$130 - 0.90 - 1.00$	61.08	1803	1.95	9.80	1770	0.00	8.04
$130 - 0.95 - 1.00$	66.19	2231	1.43	9.05	2143	0.00	7.75
$130 - 0.80 - 1.25$	41.19	555	0.00	3.42	575	0.00	3.42
$130 - 0.85 - 1.25$	44.69	536	0.09	3.55	538	0.00	3.46
$130 - 0.90 - 1.25$	49.35	857	2.93	7.94	516	0.00	5.25
$130 - 0.95 - 1.25$	57.58	1300	0.00	2.81	1329	0.00	2.81
$130 - 0.80 - 1.50$	34.50	563	0.00	3.49	586	0.00	3.49
$130 - 0.85 - 1.50$	38.73	577	0.00	3.49	602	0.00	3.49
$130 - 0.90 - 1.50$	41.46	596	0.00	3.42	629	0.00	3.42
$130 - 0.95 - 1.50$	47.54	840	2.92	7.93	523	0.00	5.25

Tables 5.1 and 5.2 clearly show that HEUR2 (which incorporates a branch-and-cut

Table 5.2: Performance comparison between HEUR1 and HEUR2

	$HEUR1$				$HEUR2$			
$\|I\|$	CPU	$\%DEV$	$\%AGAP$	$\#Best$	CPU	$\%DEV$	$\%AGAP$	$\#Best$
30	144.82	0.09	8.55	4/12	123.08	0.70	9.03	2/12
50	256.90	0.00	5.64	3/12	244.80	0.49	6.09	0/12
70	390.80	0.42	4.73	0/12	367.38	0.00	4.34	4/12
90	570.33	1.59	5.15	0/12	499.68	0.00	3.70	6/12
11	808.70	1.18	5.62	1/12	685.28	0.05	4.58	6/12
130	1004.83	1.02	6.21	1/12	933.05	0.07	5.34	6/12

framework) provides the best performance by yielding the smallest average percent deviation $\%DEV$ compared to the constructive heuristic approach HEUR1, specially for the largest-size instances. For example, HEUR2 yields an overall average of 0.22% (ranging from 0.00% to 0.70%), whereas HEUR1 provides 0.72% (ranging from 0.00% to 1.59%). Moreover, HEUR2 was able to find better solutions in 24 out of the 72 considered instances, while HEUR1 was only able to find 9 better solutions out of the 72 instances.

Chapter 6

Conclusions

This dissertation considers three challenging combinatorial optimization problems related to location and network design problems: $i)$ the Cycle Hub Location problem, $ii)$ the Modular Hub Location problem, and $iii)$ the Dynamic Facility Location Problem with Service Level Constraints. For the CHLP, we have presented and compared the path based and flow based formulations for the problem. We presented two solution approaches: a branch-and-cut based exact approach and a heuristic approach. Two families of valid inequalities based on mixed-dicut inequalities were presented and extensive computational experiments were conducted to evaluate their impact on the quality of LP bounds. These valid inequalities were embedded into a branch-and-cut framework to improve the lower bound at some nodes of the enumeration tree. A GRASP meta-heuristic was also presented to efficiently obtain high quality solutions. Computational results on benchmark instances with up to 100 nodes confirm the efficiency and robustness of the proposed algorithms.

Another main contribution of this dissertation is the work on the MHLP with single assignments. The MHLP explicitly models the flow dependency of transportation cost using modular arc costs. Moreover, it does not assume a particular topological structure, instead it considers the design of the entire hub network as a part of the decision process. We presented two mixed integer programming formulations -a flow-based and a path-based formulations and compared their strengths using linear programming relaxation bounds. We proposed a Lagrangean relaxation of the path-based formulation by relaxing the linking constraints of the location/allocation and routing variables.

We presented a primal heuristic to construct a feasible solution and compute upper bounds. Further, we presented a branch-and-bound based exact algorithm that uses the Lagrangean relaxation as a bounding procedure at the nodes of an enumeration tree. Computational results on benchmark instances with up to 75 nodes confirm the efficiency and the robustness of the proposed algorithms. We also analyzed the effect of changes in hub and access arcs capacities as well as changes in variable costs of hub and access arcs on optimal solutions. The obtained results showed that solution networks tend to have more open hubs when increasing the capacities of hub arcs, whereas an opposite behavior is observed for the case of access arcs. For the case of the variable costs of hub and access arcs, more hub arcs and hub nodes are activated in optimal solution networks when the variable costs on hub arcs decrease and variable costs on access arcs increase.

In the last part of this dissertation, we studied DFLPSL. This problem seeks to locate a set of facilities with sufficient capacities to meet customers demands through a planning horizon at a minimum cost while service level requirement for each costumer is met. To ensure the service quality, we imposed a set of service level constraints that guarantees the probability of a customer will receive its demand in less than a predefined units of time is more than a threshold value. The service level constraints consider both travel and service times, and demands as random variables that are followed certain probability distributions. In particular, we assumed that each facility is modeled as an M/M/1 queuing system. Moreover, we assumed that the travel time between facilities and customers is followed a gamma distribution. We derived a closed form expression of the service level constraints by simplifying the probability density functions associated with random quantities. The resulted formulation is nonlinear therefore, commercial solvers cannot solve it directly in its present format. To this end, we proposed two heuristics algorithms followed by a local search procedure to obtain good quality solutions. The first one, HURE1, that uses the information obtained from solving a relaxed liner programming formulation to construct feasible solutions to the problem, whereas the second one, HURE2, is based on liner approximation to the service level constraints using subgradient methods. Computational experiments showed that HURE2 outperformed HURE1 in terms of CPU time, the average final gaps $\%AGAP$ and the total number of best obtained solutions in the all tested instances. We also analyzed the sensitivity of our model to the input parameters such as service level parameter θ, response time τ and coefficient of variation CV. Solutions with larger θ and CV and smaller τ have more located facilities in the network to react faster to demands.

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 directions in which we are interested. One possible direction is the development efficient formulations and sophisticated solution algorithms for different variants of the MHLP such as single allocation with direct connections, multiple allocation with or without direct connections. Another promising avenue that can be the incorporation of capacity constraints on hub nodes.

Another topic of research is to improve the proposed heuristic for the DFLPSL by considering more neighborhood structures. Moreover, different metaheuristic algorithms can be developed and compared to obtain high quality solutions to the problem. Exact solution algorithm, such as branch-and-bound, might also be considered to solve the problem. Clearly, solving the dynamic facility location problem without including service level constraints provides valid lower bounds for the original problem. These lower bounds can be utilized in solving the DFLPSL within a branch-and-bound framework.

References

Aboolian, R., Berman, O., & Drezner, Z. (2008). Location and allocation of service units on a congested network. *IIE Transactions*, *40*(4), 422–433.

Aboolian, R., Berman, O., & Drezner, Z. (2009). The multiple server center location problem. *Annals of Operations Research*, *167*(1), 337–352.

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

Alumur, S., & Kara, B. Y. (2009). A hub covering network design problem for cargo applications in turkey. *Journal of the Operational Research Society*, *60*(10), 1349–1359.

Alumur, S. A., Kara, B. Y., & Karasan, O. E. (2009). The design of single allocation incomplete hub networks. *Transportation Research Part B: Methodological*, *43*(10), 936–951.

Alumur, S. A., Nickel, S., Saldanha-da Gama, F., & Seçerdin, Y. (2016). Multi-period hub network design problems with modular capacities. *Annals of Operations Research*, *246*(1-2), 289–312.

Amiri, A. (1997). Solution procedures for the service system design problem. *Computers & Operations Research*, *24*(1), 49–60.

Amiri, A. (1998). The design of service systems with queueing time cost, workload capacities and backup service. *European Journal of Operational Research*, *104*(1), 201–217.

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

Antunes, A., & Peeters, D. (2001). On solving complex multi-period location models using simulated annealing. *European Journal of Operational Research*, *130*(1), 190–201.

Baldacci, R., Dell'Amico, M., & González, J. S. (2007). The capacitated m-ring-star problem. *Operations Research*, *55*(6), 1147–1162.

Ballou, R. H. (1968). Dynamic warehouse location analysis. *Journal of Marketing Research*, 271–276.

Baron, O., Berman, O., & Krass, D. (2008). Facility location with stochastic demand and constraints on waiting time. *Manufacturing & Service Operations Management*, *10*(3), 484–505.

Beasley, J. E. (1990). Or-library: distributing test problems by electronic mail. *Journal of the operational research society*, *41*(11), 1069–1072.

Behmardi, B., & Lee, S. (2008). Dynamic multi-commodity capacitated facility location problem in supply chain. In *Iie annual conference. proceedings* (p. 1914).

Berman, O., & Drezner, Z. (2006). Location of congested capacitated facilities with distance-sensitive demand. *IIE Transactions*, *38*(3), 213–221.

Berman, O., & Drezner, Z. (2007). The multiple server location problem. *Journal of the Operational Research Society*, *58*(1), 91–99.

Berman, O., & Krass, D. (2015). Stochastic location models with congestion. In G. Laporte, S. Nickel, & F. S. da Gama (Eds.), *Location science* (pp. 443–486). Springer.

Berman, O., & Larson, R. C. (1985). Optimal 2-facility network districting in the presence of queuing. *Transportation Science*, *19*(3), 261–277.

Berman, O., & Odoni, A. R. (1982). Locating mobile servers on a network with markovian properties. *Networks*, *12*(1), 73–86.

Brännlund, U. (1995). A generalized subgradient method with relaxation step. *Mathematical Programming*, *71*(2), 207–219.

Brimberg, J., & Mladenovic, N. (1996). A variable neighbourhood algorithm for solving the continuous location-allocation problem. *Studies in Locational Analysis*, *10*, 1–12.

Brotcorne, L., Laporte, G., & Semet, F. (2003). Ambulance location and relocation models. *European Journal of Operational Research*, *147*(3), 451–463.

Bryan, D. (1998). Extensions to the hub location problem: Formulations and numerical examples. *Geographical Analysis*, *30*(4), 315–330.

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

Calık, H., Alumur, S. A., Kara, B. Y., & Karasan, O. E. (2009). A tabu-search based heuristic for the hub covering problem over incomplete hub networks. *Computers & Operations Research*, *36*(12), 3088–3096.

Camerini, P. M., Fratta, L., & Maffioli, F. (1975). On improving relaxation methods by modified gradient techniques. *Matehmatical Programming Study*, *3*, 26–34.

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

Campbell, J. F. (2009). Hub location for time definite transportation. *Computers & Operations Research*, *36*(12), 3107–3116.

Campbell, J. F. (2013). Modeling economies of scale in transportation hub networks. In *System sciences (hicss), 2013 46th hawaii international conference on* (pp. 1154–1163).

Campbell, J. F., Ernst, A., & Krishnamoorthy, M. (2005a). Hub arc location problems: Part II: Formulations and optimal algorithms. *Management Science*, *51*(10), 1556–1571.

Campbell, J. F., Ernst, A., & Krishnamoorthy, M. (2005b). Hub arc location problems:

part I: Introduction and results. *Management Science*, *51*(10), 1540–1555.

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

Canel, C., Khumawala, B. M., Law, J., & Loh, A. (2001). An algorithm for the capacitated, multi-commodity multi-period facility location problem. *Computers & Operations Research*, *28*(5), 411–427.

Castillo, I., Ingolfsson, A., & Sim, T. (2009). Social optimal location of facilities with fixed servers, stochastic demand, and congestion. *Production and Operations Management*, *18*(6), 721–736.

Castro, J., Nasini, S., & Saldanha-da Gama, F. (2017). A cutting-plane approach for large-scale capacitated multi-period facility location using a specialized interior-point method. *Mathematical Programming*, *163*(1-2), 411–444.

Cetiner, S., Sepil, C., & Sural, H. (2010). Hubbing and routing in postal delivery systems. *Annals of Operations Research*, *181*, 109–124.

Chardaire, P., Sutter, A., & Costa, M.-C. (1996). Solving the dynamic facility location problem. *Networks*, *28*(2), 117–124.

Cheong, M. L., Bhatnagar, R., & Graves, S. C. (2005). Logistics network design with differentiated delivery lead-time: Benefits and insights. *Proceedings of 2005 SMA conference*, 1–20.

Chrissis, J. W., Davis, R. P., & Miller, D. M. (1982). The dynamic set covering prblem. *Applied Mathematical Modelling*, *6*(1), 2 - 6.

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

Contreras, I. (2015). Hub location problems. In G. Laporte, F. Saldanha da Gama, & S. Nickel (Eds.), *Location science* (pp. 311–344). New York: Springer.

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

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

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

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

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

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

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

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

Contreras, I., Tanash, M., & Vidyarthi, N. (2016). Exact and heuristic approaches for the cycle hub location problem. *Annals of Operations Research*, *258*, 1–23.

Cook, W., Cunningham, W., Pulleybank, W., & Schrijver, A. (1998). *Combinatorial optimization*. John Wiley & Sons, New York.

Corberán, Á., Peiró, J., Campos, V., Glover, F., & Martí, R. (2016). Strategic oscillation for the capacitated hub location problem with modular links. *Journal of Heuristics*, *22*(2), 221–244.

Cornuéjols, G., Nemhauser, G. L., & Wolsey, L. A. (1983). *The uncapacitated facility* *location problem* (Tech. Rep.). Carnegie-mellon univ pittsburgh pa management sciences research group.

Correia, I., & Melo, T. (2016). Multi-period capacitated facility location under delayed demand satisfaction. *European Journal of Operational Research*, *255*(3), 729–746.

Correia, I., & Melo, T. (2017). A multi-period facility location problem with modular capacity adjustments and flexible demand fulfillment. *Computers & Industrial Engineering*, *110*, 307–321.

Correia, I., Melo, T., & Saldanha-da Gama, F. (2013). Comparing classical performance measures for a multi-period, two-echelon supply chain network design problem with sizing decisions. *Computers & Industrial Engineering*, *64*(1), 366–380.

Cortinhal, M., Lopes, M., & Melo, M. (2015). Dynamic design and re-design of multi-echelon, multi-product logistics networks with outsourcing opportunities: A computational study. *Computers & Industrial Engineering*, *90*, 118–131.

Costa, A. M. (2005). A survey on benders decomposition applied to fixed-charge network design problems. *Computers & operations research*, *32*(6), 1429–1450.

Cunha, C. B., & Silva, M. R. (2007). A genetic algorithm for the problem of configuring a hub-and-spoke network for a ltl trucking company in brazil. *European Journal of Operational Research*, *179*(3), 747–758.

Current, J. R., & Schilling, D. A. (1994). The median tour and maximal covering tour problems: Formulations and heuristics. *European Journal of Operational Research*, *73*(1), 114–126.

Daskin, M. S. (2011). *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons, New York.

Davari, S., Zarandi, M. F., & Turksen, I. (2013). The incomplete hub-covering location problem considering imprecise location of demands. *Scientia Iranica*, *20*(3), 983–991.

de Camargo, R. S., de Miranda Jr, G., & Luna, H. P. L. (2009). Benders decomposition for hub location problems with economies of scale. *Transportation Science*, *43*(1), 86–97.

Delmelle, E. M., Thill, J.-C., Peeters, D., & Thomas, I. (2014). A multi-period capacitated school location problem with modular equipment and closest assignment considerations. *Journal of Geographical Systems*, *16*(3), 263–286.

Dias, J., Captivo, M. E., & Clímaco, J. (2006). Capacitated dynamic location problems with opening, closure and reopening of facilities. *IMA Journal of Management Mathematics*, *17*(4), 317–348.

Dias, J., Captivo, M. E., & Clímaco, J. (2007). Dynamic location problems with discrete expansion and reduction sizes of available capacities. *Investigação Operacional*, *27*(2), 107–130.

Eiselt, H. A., & Marianov, V. (2011). *Foundations of location analysis* (Vol. 155). Springer Science & Business Media.

Elhedhli, S. (2006). Service system design with immobile servers, stochastic demand, and congestion. *Manufacturing & Service Operations Management*, *8*(1), 92–97.

Ernst, A. T., & Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location science*, *4*(3), 139–154.

Ernst, A. T., & Krishnamoorthy, M. (1998a). Exact and heuristic algorithms for the uncapacitated multiple allocation p-hub median problem. *European Journal of Operational Research*, *104*(1), 100–112.

Ernst, A. T., & Krishnamoorthy, M. (1998b). An exact solution approach based on shortest-paths for p-hub median problems. *INFORMS Journal on Computing*, *10*(2), 149–162.

Farahani, R. Z., Hekmatfar, M., Arabani, A. B., & Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications.

Computers & Industrial Engineering, *64*(4), 1096–1109.

Festa, P., & Resende, M. (2011). GRASP: basic components and enhancements. *Telecommunication Systems*, *46*(3), 253-271.

Fong, C., & Srinivasan, V. (1986). The multiregion dynamic capacity expansion problem: an improved heuristic. *Management Science*, *32*(9), 1140–1152.

Fong, C. O., & Srinivasan, V. (1981). The multiregion dynamic capacity expansion problem, part I. *Operations Research*, *29*(4), 787–799.

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

Galvão, R. D., & Santibanez-Gonzalez, E. d. R. (1992). A lagrangean heuristic for the pk-median dynamic location problem. *European Journal of Operational Research*, *58*(2), 250–262.

Gendreau, M., Laporte, G., & Semet, F. (1997). The covering tour problem. *Operations Research*, *45*(4), 568–576.

Geoffrion, A. M., & Graves, G. W. (1974). Multicommodity distribution system design by benders decomposition. *Management Science*, *20*(5), 822–844.

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

Grinstead, C. M., & Snell, J. L. (2012). *Introduction to probability*. American Mathematical Soc.

Guignard, M. (2003). Lagrangean relaxation. *Top*, *11*(2), 151–200.

Gunawardane, G. (1982). Dynamic versions of set covering type public facility location problems. *European Journal of Operational Research*, *10*(2), 190–195.

Guta, B. (2003). *Subgradient optimization methods in integer programming with an application to a radiation therapy problem* (Unpublished doctoral dissertation). University of Kasierslautern, Germany.

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

Hakimi, S. L., & Maheshwari, S. (1972). Optimum locations of centers in networks. *Operations Research*, *20*(5), 967–973.

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

Hoff, A., Peiró, J., Corberán, Á., & Martí, R. (2016). Adaptive memory programming for the capacitated modular hub location problem. *Technical Report*.

Hoff, A., Peiró, J., Corberán, Á., & Martí, R. (2017). Heuristics for the capacitated modular hub location problem. *Computers & Operations Research*, *86*, 94–109.

Holmberg, K., & Hellstrand, J. (1998). Solving the uncapacitated network design problem by a lagrangean heuristic and branch-and-bound. *Operations research*, *46*(2), 247–259.

Horner, M. W., & O'Kelly, M. E. (2001). Embedding economies of scale concepts for hub network design. *Journal of Transport Geography*, *9*(4), 255–265.

Ilić, A., Urošević, D., Brimberg, J., & Mladenović, N. (2010). A general variable neighborhood search for solving the uncapacitated single allocation p-hub median problem. *European Journal of Operational Research*, *206*(2), 289–300.

Jena, S. D., Cordeau, J.-F., & Gendron, B. (2015). Dynamic facility location with generalized modular capacities. *Transportation Science*, *49*(3), 484–499.

Kelly, D. L., & Marucheck, A. S. (1984). Planning horizon results for the dynamic warehouse location problem. *Journal of Operations Management*, *4*(3), 279–294.

Khumawala, B., & Clay Whybark, D. (1976). Solving the dynamic warehouse location problem. *International Journal of Physical Distribution*, *6*(5), 238–251.

Kim, J.-G., & Tcha, D.-W. (1992). Optimal design of a two-level hierarchical network with tree-star configuration. *Computers & Industrial Engineering*, *22*(3), 273–281.

Kimms, A. (2006). Economies of scale in hub & spoke network design models: We have it all wrong. In *Perspectives on operations research* (pp. 293–317). Springer.

Klincewicz, J. (1998). Hub location in backbone/tributary network design: a review. *Location Science*, *6*(1), 307–335.

Klincewicz, J. G. (2002). Enumeration and search procedures for a hub location problem with economies of scale. *Annals of Operations Research*, *110*(1-4), 107–122.

Kolen, A. (1983). Solving covering problems and the uncapacitated plant location problem on trees. *European Journal of Operational Research*, *12*(3), 266–278.

Labbé, M., Laporte, G., Martín, R., & Salazar-González, J. J. (2004). The ring star problem: Polyhedral analysis and exact algorithm. *Networks*, *43*(3), 177–189.

Labbé, M., Laporte, G., Rodríguez Martín, I., & Salazar-González, J. J. (2005). Locating median cycles in networks. *European Journal of Operational Research*, *160*(2), 457–470.

Labbé, M., Laporte, G., & Rodríguez-Martín, I. (1998). Path, tree and cycle location. In *Fleet management and logistics* (pp. 187–204). Springer.

Labbé, M., & Yaman, H. (2004). Projecting the flow variables for hub location problems. *Networks*, *44*(2), 84–93.

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

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

Labbé, M., Yaman, H., & Gourdin, E. (2005). A branch and cut algorithm for hub location problems with single assignment. *Mathematical Programming*, *102*(2), 371–405.

Laporte, G., Nickel, S., & da Gama, F. S. (2015). *Location science*. Springer.

Laporte, G., & Rodríguez Martín, I. (2007). Locating a cycle in a transportation or a telecommunications network. *Networks*, *50*(1), 92–108.

Larson, R. C. (1974). A hypercube queuing model for facility location and redistricting in urban emergency services. *Computers & Operations Research*, *1*(1), 67–95.

Lawler, E. L. (1979). Fast approximation algorithms for knapsack problems. *Mathematics of Operations Research*, *4*(4), 339–356.

Lee, S.-B., & Luss, H. (1987). Multifacility-type capacity expansion planning: algorithms and complexities. *Operations Research*, *35*(2), 249–253.

Lee, Y., Chiu, S., & Sanchez, J. (1998). A branch and cut algorithm for the steiner ring star problem. *International Journal of Management Science*, *4*(1), 21–34.

Lee, Y., Lim, B. H., & Park, J. S. (1996). A hub location problem in designing digital data service networks: Lagrangian relaxation approach. *Location Science*, *4*(3), 185–194.

Lee, Y., Lu, L., Qiu, Y., & Glover, F. (1993). Strong formulations and cutting planes for designing digital data service networks. *Telecommunication Systems*, *2*(1), 261–274.

Liefooghe, A., Jourdan, L., & Talbi, E.-G. (2010). Metaheuristics and cooperative approaches for the bi-objective ring star problem. *Computers & Operations Research*, *37*(6), 1033–1044.

Luss, H. (1982). Operations research and capacity expansion problems: A survey. *Operations research*, *30*(5), 907–947.

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

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

Marianov, V. (2003). Location of multiple-server congestible facilities for maximizing expected demand, when services are non-essential. *Annals of Operations Research*, *123*(1-4), 125–141.

Marianov, V., & Ríos, M. (2000). A probabilistic quality of service constraint for a location model of switches in atm communications networks. *Annals of Operations Research*, *96*(1), 237–243.

Marianov, V., & Serra, D. (1998). Probabilistic, maximal covering location–allocation models for congested systems. *Journal of Regional Science*, *38*(3), 401–424.

Marianov, V., & Serra, D. (2001). Hierarchical location–allocation models for congested systems. *European Journal of Operational Research*, *135*(1), 195–208.

Marianov, V., & Serra, D. (2002). Location–allocation of multiple-server service centers with constrained queues or waiting times. *Annals of Operations Research*, *111*(1), 35–50.

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

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

Martins de Sá, E., de Camargo, R., & de Miranda, G. (2013). An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, *226*, 185–202.

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

Melo, M. T., Nickel, S., & Da Gama, F. S. (2006). Dynamic multi-commodity capacitated facility location: a mathematical modeling framework for strategic supply chain planning. *Computers & Operations Research*, *33*(1), 181–208.

Min, H., & Melachrinoudis, E. (1999). The relocation of a hybrid manufacturing/distribution facility from supply chain perspectives: a case study. *Omega*, *27*(1), 75–85.

Mirchandani, P. B., & Francis, R. L. (Eds.). (1990). *Discrete location theory*. John Wiley

& Sons, New York.

Mirchandani, P. B., & Odoni, A. R. (1979). Locations of medians on stochastic networks. *Transportation Science*, *13*(2), 85–97.

Mirzaghafour, F. (2013). *Modular hub location problems* (Unpublished master's thesis). Concordia University, Montreal, Canada.

Moon, I. D., & Chaudhry, S. S. (1984). An analysis of network location problems with distance constraints. *Management Science*, *30*(3), 290–307.

Moore, G. C., & ReVelle, C. (1982). The hierarchical service location problem. *Management science*, *28*(7), 775–780.

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

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

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

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.

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

O'Kelly, M. E., & Bryan, D. (1998). Hub location with flow economies of scale. *Transportation Research Part B: Methodological*, *32*(8), 605–616.

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

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

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

Ortiz-Astorquiza, C., Contreras, I., & Laporte, G. (2015). The minimum flow cost Hamiltonian cycle problem: A comparison of formulations. *Discrete Applied Mathematics*, *187*, 140–154.

Podnar, H., Skorin-Kapov, J., & Skorin-Kapov, D. (2002). Network cost minimization using threshold-based discounting. *European Journal of Operational Research*, *137*(2), 371–386.

Polus, A. (1979). A study of travel time and reliability on arterial routes. *Transportation*, *8*, 141–151.

Racunica, I., & Wynter, L. (2005). Optimal location of intermodal freight hubs. *Transportation Research Part B: Methodological*, *39*(5), 453–477.

Rajagopalan, H. K., Saydam, C., & Xiao, J. (2008). A multiperiod set covering location model for dynamic redeployment of ambulances. *Computers & Operations Research*, *35*(3), 814–826.

Rastani, S., Setak, M., & Karimi, H. (2015). Capacity selection for hubs and hub links in hub location problems. *International Journal of Management Science and Engineering Management*, 1–11.

Roodman, G. M., & Schwarz, L. B. (1975). Optimal and heuristic facility phase-out strategies. *AIIE transactions*, *7*(2), 177–184.

Roodman, G. M., & Schwarz, L. B. (1977). Extensions of the multi-period facility phase-out model: New procedures and application to a phase-in/phase-out problem. *AIIE Transactions*, *9*(1), 103–107.

Sasaki, M., & Fukushima, M. (2003). On the hub-and-spoke model with arc capacity conatraints. *Journal of the Operations Research Society of Japan*, *46*(4), 409–428.

Silva, F. J. F., & De la Figuera, D. (2007). A capacitated facility location problem with constrained backlogging probabilities. *International Journal of Production Research*, *45*(21), 5117–5134.

Skorin-Kapov, D., Skorin-Kapov, J., & O'Kelly, M. (1996). Tight linear programming relaxations of uncapacitated p-hub median problems. *European Journal of Operational Research*, *94*(3), 582–593.

Smith, H. K., Laporte, G., & Harper, P. R. (2009). Locational analysis: highlights of growth to maturity. *Journal of the Operational Research Society*, *60*(1), S140–S148.

Sun, D., Farris, D., Cote, P., Shoults, R., & Chen, M. (1982). Optimal distribution substation and primary feeder planning via the fixed charge network formulation. *Power Apparatus and Systems, IEEE Transactions on*(3), 602–609.

Syam, S. S. (2000). Multiperiod capacity expansion in globally dispersed regions. *Decision Sciences*, *31*(1), 173–195.

Syam, S. S. (2008). A multiple server location–allocation model for service system design. *Computers & Operations Research*, *35*(7), 2248–2265.

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

Van Roy, T. J., & Erlenkotter, D. (1982). A dual-based procedure for dynamic facility location. *Management Science*, *28*(10), 1091–1105.

Wang, Q., Batta, R., & Rump, C. M. (2002). Algorithms for a facility location problem with stochastic customer demand and immobile servers. *Annals of operations Research*, *111*(1), 17–34.

Wang, Q., Batta, R., & Rump, C. M. (2004). Facility location models for immobile servers with stochastic demand. *Naval Research Logistics (NRL)*, *51*(1), 137–152.

Wesolowsky, G. O., & Truscott, W. G. (1975). The multiperiod location-allocation problem with relocation of facilities. *Management Science*, *22*(1), 57–65.

Wilhelm, W., Han, X., & Lee, C. (2013). Computational comparison of two formulations for dynamic supply chain reconfiguration with capacity expansion and contraction. *Computers & Operations Research*, *40*(10), 2340–2356.

Xu, J., Chiu, S. Y., & Glover, F. (1999). Optimizing a ring-based private line telecommunication network using tabu search. *Management Science*, *45*(3), 330–345.

Yaghini, M., & Akhavan, R. (2012). Multicommodity network design problem in rail freight transportation planning. *Procedia-Social and Behavioral Sciences*, *43*, 728–739.

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

Yaman, H. (2008). Star p-hub median problem with modular arc capacities. *Computers & Operations Research*, *35*(9), 3009–3019.

Yaman, H. (2009). The hierarchical hub median problem with single assignment. *Transportation Research Part B: Methodological*, *43*(6), 643–658.

Yaman, H., & Carello, G. (2005). Solving the hub location problem with modular link capacities. *Computers & Operations Research*, *32*(12), 3227–3245.

Yaman, H., & Elloumi, S. (2012). Star p-hub center problem and star p-hub median problem with bounded path lengths. *Computers & Operations Research*, *39*(11), 2725–2732.

Yoon, M.-G., & Current, J. (2006). The hub location and network design problem with fixed and variable arc costs: formulation and dual-based solution heuristic. *Journal of the Operational Research Society*, *59*(1), 80–89.