Multi-level facility location problems

Camilo Ortiz Astorquiza

A Thesis
in
the Department
of
Mechanical and Industrial Engineering

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

April 2017

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Camilo Ortiz Astorquiza**

Entitled: **Multi-level facility location problems**

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

Doctor of Philosophy (Industrial Engineering)

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

Signed by the final examining committee:

Dr. Jia Yuan Yu	Chair
Dr. Bernard Gendron	External Examiner
Dr. Masoumeh Kazemi Zanjani	Examiner
Dr. Ali Akgunduz	Examiner
Dr. Brigitte Jaumard	Examiner
Dr. Ivan Contreras	Supervisor
Dr. Gilbert Laporte	Supervisor

Approved by ______________________________________
Chair of Department or Graduate Program Director

April 2017 ______________________________________
Dean of Faculty of Engineering and Computer Science

ABSTRACT

Multi-level facility location problems

Camilo Ortiz Astorquiza Ph.D.

Concordia University, 2017

We study of a class of discrete facility location problems, called multi-level facility location problems, that has received major attention in the last decade. These problems arise in several applications such as in production-distribution systems, telecommunication networks, freight transportation, and health care, among others. Moreover, they generalize well-known facility location problems which have been shown to lie at the heart of operations research due to their applicability and mathematical structure. We first present a comprehensive review of multi-level facility location problems where we formally define and categorize them based on the types of decisions involved. We also point out some gaps in the literature and present overviews of related applications, models and algorithms. We then concentrate our efforts on the development of solution methods for a general multi-level uncapacitated facility location problem. In particular, based on an alternative combinatorial representation of the problem whose objective function satisfies the submodularity property, we propose a mixed integer linear programming formulation. Using that same representation, we present approximation algorithms with constant performance guarantees for the problem and analyze some special cases where these worst-case bounds are sharper. Finally, we develop an exact algorithm based on Benders decomposition for a slightly more general problem where the activation of links between level of facilities is also considered part of the decision process. Extensive computational experiments are presented to assess the performance of the various models and algorithms studied. We show that the multi-level extension of some fundamental problems in operations research maintain certain structure that allows us to develop more efficient algorithms in practice.

To Maria Cristina, Julio Vicente, Elisa and Ana Lucía

Everything I am and will be

Nothing in the world takes place without optimization, and there is no doubt that all aspects of the world that have a rational basis can be explained by optimization methods.[1]

[1]Taken from Grötschel [79], who translated the statement by Leonhard Euler from 1744

Acknowledgements

I am grateful to the people who touched my life and contributed towards reaching this goal. Keeping the motivation through the tough days of the PhD program is one of the challenges and finding one's own path in science requires very subtle guidelines from those we trust. I will try to mention those who helped me maintain the goal on sight in one way or the other, but memory sometimes can play games with our mind.

Foremost, I am grateful to my supervisors, Dr. Ivan Contreras and Dr. Gilbert Laporte who have been my advisors and my first-hand examples of worldwide top researchers. Their kind words, scientific curiosity, knowledge, and vast experience have transformed these years into a true learning and life-changing journey. Their guidelines were essential. I am grateful to Ivan, who has been my mentor and above all my friend. I thank him for initiating me into this beautiful world of Operations Research, for his patience, advice, commitment with this work and for the opportunity to begin this venture. My sincerest words of gratitude also go to Gilbert, for his precise and truly mind-setting help, for his support and understanding throughout these years. It has been a real privilege and a pleasure to work alongside both of you.

I would also like to thank my family who have my constant gratitude. At every step I take I am only standing on their shoulders to reach higher. In times like these, written words are the only means available to show them my appreciation. I am especially grateful to my parents, Maria Cristina and Julio Vicente, who have always been there to support me. They have encouraged my dreams and set my ethical guidelines through their own life example. Your teachings go beyond any academic program and your endless demonstrations of affection are certainly one of my main motivations in life. I also wish to thank my sister Catalina for her support and for having spent a year with us in Montreal. Your company is truly valuable to me. To my aunts for their support and unconditional help, and to my grandfather, Abolo, from whom I learned to love numbers as much as words and to love family more than anything in the world.

I will be forever grateful to my beloved wife without whom I would not have been able to carry out any of this work. This accomplishment is also hers. Thank you Ana! You made it possible from the very first day and never had doubts of our capabilities. Your love, kindness and company are indispensable to me.

I owe a debt of gratitude to all those friends who are available and ready to offer a hand when needed, especially to my extended family in Montreal, Maria and Felipe, who were always there to support us. I deeply appreciate your help but above all, your company and encouragement. I am also grateful to all the friends that helped me and my family to find our way when we first arrived in Montreal. Some of them have been my friends since I can remember; with others, this was the perfect excuse for starting a long-lasting friendship. Thank you Juan F. Garzón, Mariana Obando, Felipe González, Gerardo Jurado, and many other good friends for your generosity and support. I am also grateful to Andrés Acevedo and Diana Rodríguez, who offered me their unconditional friendship. To Andrés who bore with me during all the years of the PhD program, sometimes with a good sense of humor... essential to survive any adversity in life. To Carlos Zetina, who has become a real friend putting up with my long discussions and a true colleague sharing experiences, successes, and failures.

Looking further back in time, I cannot forget those who planted the seed of pursuing graduate studies in the first place. I am grateful to Fernando Novoa, Eddy Pariguan, Rafael Diaz, and Pedro Olaya for their encouragement and kind advice.

I would like to thank the Canadian Natural Sciences and Engineering Research Council under grants 418609-2012 and 2015-06189, to the Fonds de Recherche du Québec - Nature et Technologies under grant 2014-NC-172906, and to Colciencias Colombia under grant Doctorados Exterior 646, for partly funding this research.

Finally, to all the people that I did not mention by name but who have shown interest in this work and motivated me to get to this point: thank you!

Contribution of Authors

This dissertation is presented under the *manuscript-based* format. It contains four articles that have been accepted for publication or are under revision in different journals. Chronologically as submitted, they are as follows. The first article entitled "Multi-level facility location as the maximization of a submodular set function" was published in June 2015 in the *European Journal of Operational Research*. The second one "Formulations and approximation algorithms for multi-level uncapacitated facility location" received the final acceptance for publication in February 2017 from the *INFORMS Journal on Computing*. The third paper entitled "An exact algorithm for multi-level uncapacitated facility location" was submitted for publication to *Transportation Science* in January 2017 and the fourth paper corresponds to an invited review for the *European Journal of Operational Research* entitled "Multi-level facility location problems" and submitted in March 2017.

All four articles are co-authored with Dr. Ivan Contreras and Dr. Gilbert Laporte who established research guidelines and reviewed the papers before submission. The author of this thesis acted as the principal researcher with the corresponding duties such as mathematical proofs, development of formulations and algorithms as well as programming of solution methods and analysis of computational results along with writing the first drafts of the articles.

Contents

List of Figures

List of Tables

Main Abbreviations

MILP	Mixed-integer linear programming
LP	Linear programming
PBF	Path-based formulation
ABF	Arc-based formulation
FLP	Facility location problem
UFLP	Uncapacitated facility location problem
CFLP	Capacitated facility location problem
p-MP	p-median problem
HFLP	Hierarchical facility location problem
GNDP	General network design problem
MLFLP	Multi-level facility location problem
TUFLP	Two-level uncapacitated facility location problem
MUFLP	Multi-level uncapacitated facility location problem
MUFLP-E	MUFLP with fixed costs on the edges
MUpLP	Multi-level uncapacitated p-location problem
MUpLP-E	MUpLP with fixed costs on the edges
MFLDP	Multi-level facility location design problem
MpMP	Multi-level p-median problem

Chapter 1

Introduction

Facility location problems (FLPs) constitute a major area of interest for researchers and practitioners in operations research (OR). FLPs have emerged since the second half of the twentieth century, and are relevant to business, public service and theoretical research. The common idea behind location problems is "the notion of optimal choice within a spatial context" [96]. Thus, several applications find their place in this extensively studied field. Moreover, the mathematical structure of some FLPs has proven beneficial to develop solution methodologies that are broadly used today in OR. Numerous books and surveys are witnesses of the importance of this area, both theoretically and application-wise [see, for instance 52, 104]. In particular, discrete FLPs have played a major role in locational analysis [127], that is, FLPs where the location decisions are restricted to the vertices of an underlying network.

In this thesis we study of a class of discrete FLPs that has attracted increasing attention in the last decade, called *multi-level facility location problems* (MLFLPs). Perhaps, an important motivation for this interest in MLFLPs is the fact that they generalize some fundamental FLPs such as the uncapacitated facility location problem (UFLP) [46, 100] and the p-median problem (p-MP) [83], while retaining several mathematical properties. In Chapter 2 we present a comprehensive review of the existing literature on MLFLPs, consisting of more than 60 articles among which more than 40 date within the last ten years. We also discuss some of the gaps that appear in the field that also motivated this work. Although the review paper attempts to cover the entire spectrum of MLFLPs, in the other chapters of this thesis we concentrate on the uncapacitated cases only.

In particular, we study a general class of MLFLP denoted *multi-level uncapacitated p-location problems* (MUpLPs) along special cases and extensions. The MUpLP can be defined as follows. Let $I = \{1, \cdots, m\}$ be the set of customers, $V_1, \cdots, V_k$ be the sets of sites among which facilities of levels 1 to k can be selected (or opened), with $V = \cup_{r=1}^{k} V_r$. Also, consider $c_{ij_1 \cdots j_k}$ to be the profit associated with the allocation of customer i to the sequence of facilities $j_1, \cdots, j_k$, where $j_r \in V_r$. Now, let $p = (p_1, \cdots, p_k)$ be a vector of positive integers, and let f_{j_r} be the non-negative fixed cost associated with opening facility j_r at level r. The MUpLP consists of selecting a set of facilities to open, such that no more than p_r facilities are opened at level r and of assigning each customer to a set of open facilities, exactly one at each level, while maximizing the total profit minus the setup cost of the open facilities.

Note that the single-level version (i.e. $k = 1$) of the MUpLP corresponds to the well-known uncapacitated p-location problem (UpLP) [45] which in turn subsumes the UFLP and the p-MP. Thus, multi-level extensions of the UFLP and the p-MP are also special cases of the MUpLP. Namely, the multi-level uncapacitated facility location problem (MUFLP) [90, 153] is obtained when all cardinality constraints are redundant, i.e. when $p_r = |V_r|$ for all r, and a generalization of the p-MP, called the multi-level p-median problem (MpMP), is derived when all setup costs are set to zero, that is, $f_{j_r} = 0$.

Yet another important class of MLFLPs arises when network design decisions such as the activation of links in a network are considered. Thus, we also study a slightly more general version of the MUpLP which includes additional costs for opening edges between levels of facilities. This problem is denoted as the MUpLP with edge costs (MUpLP-E). Contreras and Fernández [39] review single-level problems where non-trivial location and network design decisions are part of the optimization process. The authors denoted them as General network design problems (GNDP). To the best of our knowledge, the MUpLP and therefore the MpMP and the MUpLP-E have not been defined before. Moreover, among the numerous publications relating MLFLPs only a few are concerned with the development of exact algorithms for the MUFLP. The objectives of this thesis can be summarized as follows.

- To define a general class of MLFLPs by identifying their main characteristics and differentiating factors from related areas.
- To provide a comprehensive review on MLFLPs and an appropriate classification scheme.
- To investigate the viability of extending some theoretical results previously obtained for single-level FLPs to the MLFLPs.
- To introduce and computationally compare mathematical formulations for the general MUpLP.
- To devise approximate and exact algorithms that efficiently attain optimal or near-optimal solutions for instances of the MUpLP and the extended MUpLP-E.

This thesis contains five more chapters, four of which correspond to the articles that have been published or submitted for revision in OR-related journals, followed by the conclusions chapter. Hence, each chapter is self-contained and special caution should be taken with the notation going between chapters. Nevertheless, we have made some minor modifications to the notation as well as tables and figures from the original versions of the papers in order to improve the coherence throughout this manuscript. However, some repetition can be found between chapters especially in the corresponding introductory sections.

The remainder of this document is organized as follows. In Chapter 2 we present a comprehensive review of MLFLPs. We first discuss the main characteristics of these problems and show some similarities and differences with well-known related areas. Based on the types of decisions that are involved in the optimization process, we identify three different categories of MLFLPs. We thus present overviews of formulations, solution methods, applications and the historical development of the field. In Chapter 3, we introduce an alternative combinatorial representation for the MUFLP. An interesting observation is that the real-valued set function associated with the classical representation of the problem does not satisfy the submodular property, whereas the set function associated with the new representation does satisfy this property. This corrects a previous conclusion stating that the MUFLP is not submodular. In Chapter 4 we generalize this result to the MUpLP and exploit this characterization to derive worst-case bounds for a greedy heuristic. We also obtain sharper bounds for the special case of the MpMP. Moreover, we introduce a mixed integer linear programming formulation for the problem based on the submodularity property. We present results of computational experiments to assess the performance of the greedy heuristic and that of the formulation and compare the models with previously studied formulations. In Chapter 5 we study the more general MUpLP-E where the selection of links between levels of facilities is part of the decision process. We propose an exact algorithm based on a Benders reformulation to solve large-scale instances of the problem and some particular cases. Extensive computational experiments were carried out to assess the performance of several different variants of the Benders algorithm. Conclusions follow in Chapter 6.

Chapter 2

Multi-level facility location problems

The content of this chapter was submitted for publication under the Invited Review series entitled "Multi-level Facility Location Problems", *European Journal of Operational Research*, March 2017 [138].

Abstract

We conduct a comprehensive review on multi-level facility location problems which extend several classical facility location problems and can be regarded as a subclass within the well-established field of hierarchical facility location. We first present the main characteristics of these problems and discuss some similarities and differences with related areas. Based on the types of decisions involved in the optimization process, we identify three different categories of multi-level facility location problems. We present overviews of formulations, algorithms and applications, and we trace the historical development of the field.

2.1 Introduction

Discrete facility location problems (FLPs) constitute a major area of interest for researchers and practitioners in operations research (OR). The mathematical structure of some FLPs, which has proven fruitful to the development of solution methodologies broadly used today in OR, combined with their applicability to real-life problems, have made FLPs a core topic that has led to a vast number of publications, including several books and surveys [see, for example 52, 56, 104, 127]. A subclass of FLPs called multi-level facility location problems (MLFLPs) has attracted increasing attention in the last two decades. However, to the best of our knowledge, no recent publication consolidates the available material on this particular subject. Thus, we felt that the time was adequate to discuss the main aspects of MLFLPs in order to differentiate them from related topics and classify this rapidly emerging area. In this article we review the most representative MLFLPs as well as their historical development, models, solution methods and applications. For this purpose we survey over 60 OR-related studies published since the late 1970s, among which more than 40 have appeared in the last decade.

In an MLFLP we are given a set of customers that have a service or product requirement and a set of potential facilities partitioned into k levels. The goal is to determine which facilities to open simultaneously at each level, so that customers are assigned to one or multiple sequences of opened facilities, while optimizing an objective function. Some of these problems generalize fundamental FLPs such as the uncapacitated facility location problem (UFLP) [46, 100]. For example, in one of the first papers on MLFLPs, Kaufman et al. [90] introduced the so-called warehouse and plant location problem. Later, a slightly different version of that problem was presented and denoted as the two-level uncapacitated facility location problem (TUFLP). A natural extension to more than two levels of facilities corresponds to the multi-level uncapacitated facility location problem (MUFLP).

MLFLPs can also be viewed as a special case of an important class of problems called hierarchical facility location problems (HFLPs), where systems involving different types of interacting facilities that provide services to a set of customers are studied. Applications of HFLPs arise naturally in supply chain management (SCM) [123], where the interactions between plants, warehouses, distribution centers, and retail stores play a major role, and in health care systems [143] in which users must be served from different levels of clinics and hospitals. Other examples arise in hierarchical telecommunication networks [34, 77], freight transportation [68, 70], and solid waste management systems [22]. The two surveys of Şahin and Süral [47] and Zanjirani Farahani et al. [168] provide classifications and overviews of models, applications, and algorithms for HFLPs. Reference [47] covers the literature until 2004. Reference [168] is more recent but does not present most of the papers on MLFLPs in the broader context of HFLPs. Perhaps, one of the reasons for the exclusion of some of these problems is that they are known under different names and can be confused with similar, out-of-scope, problems. When preparing this survey, we have found that the terms *multi-echelon*, *multi-stage*, *multi-level*, *hierarchical*, and *multi-layer* facility location problems have all been used to refer to what we call MLFLPs.

The main contribution of this article is twofold. First, we formally define MLFLPs in order to present a unified framework for this still-growing area of research, and to differentiate it from other related areas within the field of facility location. Second, we consolidate the main contributions in the context of MLFLPs with a comprehensive review dating back to 1977 but with an emphasis on the last two decades. The paper is organized as follows. Section 2.2 establishes the types of decisions that pertain to MLFLPs and discusses the main characteristics of these problems. It also relates them with well-known areas of research and describes some of the applications that have been most relevant to MLFLPs. In Section 2.3 we present some of the historical milestones of the area and identify the main categories of MLFLPs that have been studied. We also discuss some variants and summarize the main references. Sections 2.4 to 2.6 are divided following the proposed classification scheme for MLFLPs. In each of the latter sections we provide overviews of the corresponding models and algorithms. Conclusions follow in Section 2.7. To facilitate reading, Table 2.1 summarizes the main abbreviations used throughout the paper.

OR: Operations research	TUFLP: Two-level uncapacitated facility location problem
MILP: Mixed-integer linear programming	TEUFLP: Two-echelon uncapacitated facility location problem
ILP: Integer linear programming	TFLDP: Two-level facility location design problem
LP: Linear programming	TECFLP-S: Two-echelon CFLP with single assignment constraints
PBF: Path-based formulation	TCFLP: Two-level capacitated facility location problem
ABF: Arc-based formulation	TUFLP-S: TUFLP with single assignment constraints
FLP: Facility location problem	TCFLP-E: TCFLP with edge set-up costs
HFLP: Hierarchical facility location problem	MUFLP: Multi-level uncapacitated facility location problem
GNDP: General network design problem	MUFLP-E: MUFLP with edge set-up costs
UFLP: Uncapacitated facility location problem	MUpLP: Multi-level uncapacitated p-location problem
CFLP: Capacitated facility location problem	MUpLP-E : MUpLP with edge set-up costs
p-MP: p-median problem	MFLDP: Multi-level facility location design problem
MLFLP: Multi-level facility location problem	MpMP: Multi-level p-median problem

Table 2.1: Summary of the main abbreviations

2.2 Decisions, related problems and applications

We first discuss the types of decisions that are involved in an MLFLP. For this purpose and for the sake of clarity when referring to these decisions, we introduce some notation that is used to model an MLFLP. Let $G = (V \cup I, E)$ be a graph with vertex set $V \cup I$ and edge set E. The set I corresponds to the customers, and the set V is partitioned into $\{V_1, \cdots, V_k\}$, corresponding to the sets of potential facilities at levels 1 to k. The edges always link two different levels. An MLFLP involves some of the following decisions.

Design decisions: facility location and edge activation The *location decisions* determine where to open the facilities. Given an underlying network G, facilities may be located at both the vertices or the edges of the network. This review focuses on discrete location problems, where it is assumed that facilities can only be located at the vertices of G. We refer to [75, 84, 163] for generalizations of results of Hakimi [83] on node optimality properties for HFLPs. The UFLP [46] and the p-MP [83] are well-known examples where facility location decisions are involved. The *network design decisions* select the edges to be activated. These edges are used to provide transportation services between customers and facilities of the first level, and facilities between different levels. Fixed-charged network design problems [110] are well-known problems involving network design decisions, among others.

Tactical decisions: allocation and routing The *allocation decisions* determine which facilities will be used to serve each customer. In FLPs, two types of allocation strategies have been considered. In single allocation, each customer is assigned to exactly one facility, whereas in multiple allocation each customer is allowed to be assigned to more than one facility, if beneficial. The *routing decisions* indicate the routes (or paths) on G that will be used to satisfy the customer demands. We use the term route to indicate the sequence of edges used to send flows between pairs of vertices. These types of decisions commonly appear in network flow problems which have been widely studied [11]. Since we consider different levels of facilities (V_r), the allocation decisions can also be viewed as the assignment of customers to open facilities of the first level and that of open facilities from one level to the next, sequentially. That is, a path between customers and highest-level facilities is associated with a multi-level allocation structure. Finally, observe that the network design and routing decisions are also interrelated, since the edges that can be used in the paths are determined by the network design decisions.

Both of the above types of decisions are directly related to the fixed and variable costs. For example, when a vertex $j_r \in V_r$ is selected to locate a facility, a set-up cost f_{j_r} is incurred. Analogously, when an edge $\{j_1, j_2\} \in E$ is activated a set-up cost

$d_{j_1 j_2}$ must be paid. The tactical decisions are affected by variable costs. A common example is transportation costs which are generally related to the distances between the vertices. Transportation (or distribution) costs $c_{ij_1 \cdots j_k}$ are variable since they also depend on the customer's demands d_i and the sequence of used facilities $j_1, \cdots, j_k$. Some classical problems such as the UFLP involve set-up costs for opening facilities and transportation costs for assigning customers directly to facilities.

In order to better define the scope of this survey, we further discuss the above types of decisions in the context of MLFLPs. First, it is required that non-trivial facility location decisions be taken at every level of the hierarchy, simultaneously. Other problems involve two or more levels of facilities but only in one of them is the selection of facilities considered. We present some examples of this type of problems in Section 2.2.1.2. Depending on the application, network design and routing decisions may be explicitly considered or not, that is, the activation of edges and flow patterns are not necessarily non-trivial decisions. More importantly, this type of decisions should not be confused with routing decisions commonly encountered in similar problems such as location-routing problems [12, 49], where tours or paths between vertices of the same level in the network are considered. In the case of MLFLPs, there is no direct interaction between customers, and no horizontal interactions between facilities of the same level. This can be seen from the definition of the set E which corresponds to links between facilities and customers of different levels. Typically the edges between facilities of different levels are defined sequentially, i.e., for $r = 1, \cdots, k-1$, let $E_r = \{\{a, b\} \in E : a \in V_r \text{ and } b \in V_{r+1}\}$, and let $E_0 = \{\{i, b\} \in E : i \in I \text{ and } b \in V_1\}$. When this is the case we require a sequence of exactly one open facility at each level. As we will discuss later in this section, this feature corresponds to what is called a single flow pattern in the context of HFLPs. However, some problems with multi-flow patterns are also considered as MLFLPs. These assign customers to sequences of open facilities that can skip levels. Most of these multi-flow pattern problems can be modeled as single-flow-patterns by simply adding dummy vertices in the corresponding missing levels [145, 153], at the expense of increasing the instance size.

A common requirement in MLFLPs is that every served customer must be allocated to an open facility of the k^{th} level either directly or through a sequence of open facilities, and every open facility of level r must be connected to an open facility of level $r+1$, except those of level k. When flow patterns are considered, the flow between levels must go in one direction and there ought to be only one type of arc available. Some HFLPs, especially those that arise in the framework of waste management systems, consider flows in two directions or more than one type of arc (see, for instance[22, 130]). These types of problems lie outside the scope of this paper.

Another important feature that differentiates MLFLPs from similar problems is that the set of vertices $V \cup I$ consisting of potential sites and customers is partitioned from the input into $k+1$ levels. This means that the set V is also partitioned into k subsets, one for each level of facilities. Notably, in early works the partitioning of the set V did not necessarily consist of pairwise disjoint sets [90, 145]. However, most of the more recent papers assume pairwise disjoint sets. In any case, in contrast to some HFLPs where one can open different facilities at any vertex of the network, including those that model customer zones, in MLFLPs the sets V_r differ from V_{r+1} for all r. This also means that in MLFLPs the number of levels is not part of the decision process and facilities of type r can only be located in V_r, i.e. the hierarchy is given as an input of the problem. Note also that the hierarchy is imposed only on the vertices and not on the edges, in contrast for instance to multi-level network design problems where usually the network design decisions are predominant [19, 76]. Finally, in terms of the objective function we restrict this review to those MLFLPs with median and fixed charge objective (minisum) functions. We note that in recent years variations of some MLFLPs allow the planner to have the option of incurring a penalty instead of serving all customers. Such penalties are included in the objective function and take into account the benefit of deciding which customers to serve. Therefore, we do not restrict MLFLPs to require each customer to be allocated to a sequence of open facilities.

2.2.1 Related problems

Different classes of FLPs are related to MLFLPs. We next discuss some of the areas that we consider to be most relevant to this review and we point out the main differences and similarities with MLFLPs.

2.2.1.1 Hierarchical facility location problems

We have already discussed some applications, definitions and references [47, 168] for this class of FLPs. In particular, since we consider MLFLPs as a special case of HFLPs, we have mentioned some of the differences between the two types of problems. We now emphasize other relevant differences between them. Hence we use the classification scheme and terminology of HFLPs given in [47] in order to categorize MLFLPs in that context. It is based on four criteria: *flow pattern*, *service availability*, *spatial configuration* and *objective*. A flow pattern refers to the way in which a facility at a given level receives or offers services or products to another facility at a different level and is either single-flow or multi-flow. In a network with single-flow patterns, the flow from or to the customers must pass through all higher levels until it reaches its point of origin or destination, whereas in an multi-flow pattern, facilities of some level may receive or send flow directly from or to any higher level. Service availability specifies whether a higher-level facility provides all services offered by its lower-level facilities plus another one (nested), or whether facilities at each level provide different services (non-nested). In the spatial configuration category a network can be coherent or non-coherent. In a coherent network, an open facility of a lower-level must receive or send service from or to exactly one higher-level facility. Non-coherent systems allow more than one higher-level facility serving a given lower-level facility. Median, covering and fixed charge objectives are considered in HFLPs. Therefore, for an MLFLP we have noted that the single-flow pattern is more common and in principle a non-nested structure is considered. In terms of the coherency criterion some papers have included certain assumptions while others simply impose single assignment constraints which in both cases imply a coherent structure [for example, 34, 70, 136].

Three main differences thus arise between MLFLPs and HFLPs apart from those mentioned above, namely the type of objective function, the type of demands and the service availability criterion. First, we note that other HFLPs that consider covering or pure median objectives typically appear in the context of having the same set V as potential sites for all types of facilities. In MLFLPs it is common to observe fixed-charge-type objective functions. On the other hand, the service availability criterion which was first discussed by Narula [131], is strictly interrelated with the presence of different types of demand. In some HFLPs the requirements from the customers are services, and the same customer can demand different types of service offered by certain types of facilities in the hierarchy. These problems are generally motivated by health care applications where geographical zones require service from regional hospitals, local hospitals or clinics. Examples of this feature are provided in [126, 155]. In contrast, in MLFLPs there is only one type of demand, more in the spirit of a production-distribution system where for instance, plants serve warehouses which in turn serve customers. Therefore, we assume a non-nested configuration for MLFLPs since we refer to different types of facilities instead of services, although this is also application-dependent.

2.2.1.2 Multi-echelon location-routing problems

The term multi-level is not the only one used in the context of MLFLPs. For example, we found *multi-echelon*, *multi-stage*, *multi-layer* and *multi-tier* among the more common terminologies. Typically the terms *layer* and *level* are used as synonyms, referring to the sets V_r or types of facilities as we did above. On the other hand, the term *echelon* is generally associated with distribution networks where products are transported between each pair of levels. Such pairs are called echelons [21, 67, 107]. Multi-echelon FLPs are thus very similar to MLFLPs. In Section 2.3 we highlight the main steps in the evolution of both terms. In fact, some of the papers that we review as MLFLPs denote their problems as multi-echelon. There are two main characteristics that we can use to differentiate the terminology in this case. The first one is that although all of the multi-echelon problems involve a multi-level environment, not all of them require facility location decisions at every level. For example, Geoffrion [72] Klose [91, 92] and Li et al. [107] study two-echelon FLPs in which facilities to be opened are only selected at one of the levels. This is partially because the predominant decisions are made at the echelons, and these typically involve routing variables. Indeed, the second differentiating feature lies precisely in the routing patterns. In MLFLPs we are concerned with problems where facility, and sometimes network design decisions, are predominant with no routing decisions between vertices of the same level involved. Cuda et al. [49] recently reviewed two-echelon routing problems. Another term that is generally related to echelons is the word *tier*, which has mainly been used in the context of freight transportation systems and city logistics [113, 114]. These problems also involve vehicle routing decisions and are therefore out of the scope of this paper. The term *stage* has also been used in the MLFLPs context. This is probably the most elusive one when trying to associate it to something in particular. To mention a few MLFLPs references, in [103, 118, 129] the term stage is used when referring to what we denote as levels. However, in other papers it has been used in the sense of what we identified as echelons [e.g. 91, 92, 107]. Finally, the term stages may also apply for dynamic FLPs and stochastic programs. In this review we attempt to select those that are concerned with MLFLPs.

2.2.1.3 General network design problems

We note that the types of decisions involved in MLFLPs mentioned above are very much in the spirit of those identified by Contreras and Fernández [39], who classified a broader class of optimization problems referred to as general network design problems (GNDPs), where both the facility location and network design decisions are predominant and non-trivial. Thus, MLFLPs can also be seen as a special case of the more general class of GNDPs. However, these authors concentrated on single-level problems, excluding MLFLPs from their study. Nonetheless, their classification of GNDPs based on the type of demand can be useful for our study of MLFPLs. Contreras and Fernández [39] present two main categories of GNDPs: problems that involve User-Facility demands (UF), and those with User-User demands (UU). In UF, facilities are the service providers to users and typically there are no interactions between facilities. Therefore, demands are routed from facilities to users. On the other hand in UU, facilities consolidate commodities that are routed from origins to destinations and thus, they are used as intermediate locations. The network design and routing decisions influence the optimal solution structure by deciding how to connect users to facilities and facilities to each other. This means that in most UU cases facilities interact with each other. An example of GNDPs with UU demands in which there is a multi-level environment are the so-called hierarchical hub location problems [14, 167]. In MLFLPs, from the perspective of GNDPs, we restrict our attention to those problems that have a UF demand and incorporate non-trivial network design decisions.

2.2.1.4 Supply chain network design problems

MLFLPs also relate to the well-studied area of supply chain management [16, 150]. There has been a great effort to establish the importance of location problems in SCM [50, 115, 123]. For instance, Melo et al. [123] review facility location models in the context of SCM and identify features that such models must capture in order to be consistent with the strategic decisions involved in SCM. In particular, the authors discuss the importance of having a different types of facilities, very much like an MLFLP, where the strategic decisions of the SCM system are considered. However, SCM usually involve decisions on the inventory, procurement, production, routing, etc, and thus, reviewing such a general class of problems is beyond the scope of this paper. Nevertheless, MLFLPs can be considered as a simple version of a supply chain network design problem where most of the tactical and operational decisions are not involved.

2.2.2 Applications

Two types of applications arise frequently in MLFLPs-related papers. The first one is concerned with production-distribution systems where customers require a product that must be provided from first-level facilities (warehouses) which in turn is sent from production plants. This line of research naturally evolved from some early works where the warehouse and plant location problem was introduced [90, 145]. Some variations include additional levels in the distribution network such as retail stores or distribution centers and more sophisticated models in freight transportation [36, 68, 70]. Also note that the applications that motivate this type of MLFLP generally do not exceed more than three levels of facilities. Some examples where a production-distribution system is studied are [60, 67, 81, 103, 118, 119, 129, 140, 141, 153, 157, 164, 166]. However, other papers more involved in the development of approximation algorithms for the general k-level case also make reference to distribution systems [9, 28, 35]. Moreover, an interesting variant of the problem arises when decision maker determines whether to provide the service to each customer or to pay a penalty for those that are not served [15, 29].

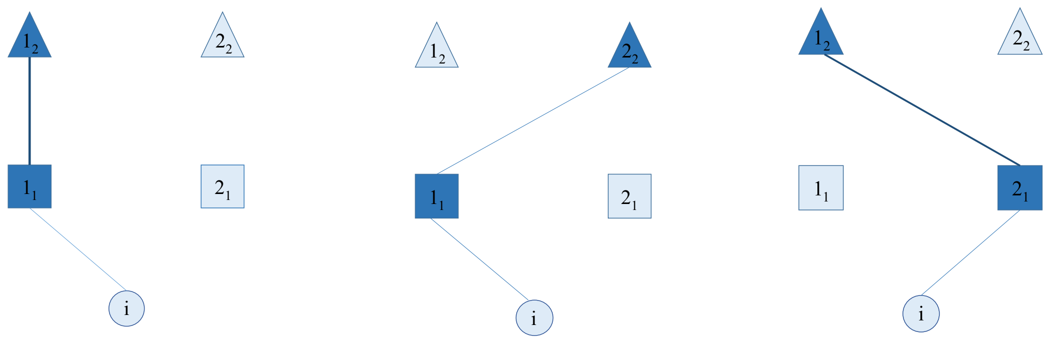
The second major application area emerges from the telecommunications industry and the design of computer networks. In this case, one must decide where to locate devices such as routers and multiplexers and how to allocate customers (terminals) to a sequence of devices. Examples of references where this type of application is discussed are [6, 7, 34, 51, 77, 80, 88, 89, 128]. Finally, other studies in the context of MLFLPs have been motivated by applications in different fields. For instance, waste disposal systems [21, 25, 26, 159], supply chain of disaster relief system [73], and health care systems [48].

2.3 A classification scheme and overview of the related literature

We now present a classification scheme for MLFLPs based on the types of decisions involved and on the different possible combinations of them. On the one hand, design decisions correspond to (*i*) opening of facilities and (*ii*) activating edges, while on the other hand, (*iii*) allocation and routing decisions are made to satisfy customer demands. Since selecting which facilities are opened at each level is a requirement of every MLFLP, we are left with the three possible combinations of (*ii*) and (*iii*) to define our categories. Given that non-trivial decisions are closely related to the types of costs (or profits) considered in the definition of each problem, we could also refer to the corresponding category by type of cost. We have selected one fundamental problem from each category in order to identify them more easily as follows. When there are only design decisions (*i*) and (*ii*) involved, we refer to them as multi-level facility location design problems (MFLDPs). When there are facility location and tactical decisions (i) and (iii), we refer to the MUFLP. Finally, when all three types of decisions are present, we refer to the MUFLP with edge set-up costs (MUFLP-E). The latter is clearly a more general version combining the former two. In the following example we sketch an instance of the three problems in a two-level environment in order to illustrate this categorization. We summarize the main notation used throughout the paper in Table 2.2.

Example 2.1. *Consider an underlying network consisting of* $I = \{i\}$, $V_1 = \{1_1, 2_1\}$ *and* $V_2 = \{1_2, 2_2\}$ *and all edges between* I *and* V_1 *as well as those between* V_1 *and* V_2 *exist. The fixed costs for opening facilities are* $f_{1_1} = 5$, $f_{2_1} = 10$, $f_{1_2} = 20$, *and* $f_{2_2} = 25$ *and we assume that there are no fixed costs for opening edges between* I *and* V_1*. We analyze three scenarios, one for each of the aforementioned representative problems in this two-level context (TFLDP, TUFLP, TUFLP-E). For the TFLDP and the TUFLP-E, consider edge set-up cost* $d^1_{1_1 1_2} = 5$, $d^1_{1_1 2_2} = 10$, $d^1_{2_1 1_2} = 3$, *and* $d^1_{2_1 2_2} = 5$*. For the TUFLP and the TUFLP-E let* $c_{i1_1 1_2} = 10$, $c_{i1_1 2_2} = 1$, $c_{i2_1 1_2} = 5$, *and* $c_{i2_1 2_2} = 5$ *be the corresponding transportation costs. Therefore, we obtain three different optimal solutions, one for each problem. For the TFLDP the optimal value is 30, opening facilities* 1_1 *and* 1_2 *as well as the edge* $\{1_1, 1_2\}$*. This solution is depicted in Figure 2.1a. We have represented with darker colors open facilities and links. Similarly, for the TUFLP and the TUFLP-E we have optimal values equals to 31 and 38, respectively. The corresponding solutions are shown in Figures 2.1b and 2.1c.*

In Sections 2.4 to 2.6 we present formal definitions of the problems and discuss the related variants and references in more detail for each category of MLFLPs. We also present what we consider to be milestones of the field, and the trends that they have defined. On our historical path towards defining those most representative MLFLPs, we introduce some commonly used MILP formulations within each category, and thus we illustrate the differences and relationships with each other as well as



(a) TFLDP solution (b) TUFLP solution (c) TUFLP-E solution

Figure 2.1: Three examples of two-level FLPs

their variations. Table 2.3 summarizes the main MLFLP publications and includes side criteria such as capacitated/uncapacitated and the solution approach (exact or approximate) that was applied in the corresponding reference. Some references may therefore appear in more than one box of the table. We also include papers containing polyhedral studies or introducing MILP formulations only in the "exact" columns.

From Table 2.3 we can observe that certain areas have received considerably more attention than others. For example, in the uncapacitated cases an important number of publications are concerned with the development of approximation algorithms, except for the MUFLP-E variant. Thus, more research must be carried out to further investigate whether adding fixed costs on the edges changes the problem drastically from an approximation perspective. Also, in the uncapacitated case, we see that models and exact algorithms have been proposed for almost all categories listed in the table. However, only recently was an exact solution method designed for large-scale instances of the general MUFLP-E with $k > 2$ [137]. On the other hand, in the capacitated versions, the effort appears to have focused on the two-level variants. This leaves aside only a few references where approximation algorithms have been designed for the general case where $k > 2$.

Table 2.2: Summary of main notation

	Notation	Definition
Sets	$G = (V \cup I, E)$	graph with vertices $V \cup I$ and edges E
	V	set of potential sites
	I	set of customers
	V_r	set of potential sites of level r, for $r = 1, \cdots, k$
	E_r	set of edges between V_r and V_{r+1}, for $r = 1, \cdots, k-1$
	E_0	set of edges between I and V_1
	Q	set of possible paths of facilities, exactly one from each level e.g. $q = j_1, \cdots, j_k \in Q$
Parameters	k	number of levels
	f_{j_r}	fixed cost for opening facility $j_r \in V_r$, for $r = 1, \cdots, k$
	$c_{ij_1 \cdots j_k}$	variable cost (or profit) for serving customer i through the sequence $j_1 \cdots j_k$
	d^r_{ab}	fixed costs for opening edge $\{a, b\} \in E_r$ for $r = 1, \cdots, k-1$
	$d^0_{ij_1}$	fixed costs for opening edge $\{i, j_1\} \in E_0$
	p_r	maximum number of facilities to open at level r, for $r = 1, \cdots, k$
	h_i	demand of customer $i \in I$
	β_{j_1}, α_{j_2}	capacities at facilities $j_1 \in V_1$ and $j_2 \in V_2$, respectively
Variables	y_{j_r}	binary decision for opening facility $j_r \in V_r$, for $r = 1, \cdots, k$
	$x_{ij_1 \cdots j_k}$	binary (continuous) decision for assigning customer $i \in I$ to sequence $j_1 \cdots j_k$ with $j_r \in V_r$ for $r = 1, \cdots, k$
	w^r_{ab}	binary decision for opening edge $\{a, b\} \in E_r$ for $r = 0, \cdots, k-1$
	z^r_{iab}	binary decision determining whether the edge $\{a, b\} \in E_r$ is used to serve customer i, for $r = 1, \cdots, k$
	v_{ij_1}	binary decision if customer i is assigned to $j_1 \in V_1$. Also used as amount of flow between $i \in I$ and $j_1 \in V_1$
	$t_{j_r j_{r+1}}$	(fraction of) flow between $j_r \in V_r$ and $j_{r+1} \in V_{r+1}$, for $r = 0, \cdots, k-1$, with $V_0 = I$
	η^i	continuous variable for the profit of serving customer $i \in I$
	x_q	binary decision for opening path $q \in Q$

2.4 MLFLPs with tactical decisions

Perhaps the simplest version of an MLFLP, yet the most studied, is the TUFLP which can be defined as follows. Assuming that all facilities are uncapacitated and given fixed costs f_{j_r} for setting up facility j_r, for $r = 1, 2$, as well as distribution costs $c_{ij_1j_2}$ for serving customer i through the pair j_1, j_2, the problem consists of determining which facilities to open at each level so that every customer is served via a pair of open facilities (j_1, j_2), while minimizing the total cost. Consider the binary decision variables y_{j_r} equal to 1 if and only if facility $j_r \in V_r$ is open, and the continuous variable $x_{ij_1j_2}$ equal to the fraction of the demand of customer i satisfied by second-level facility j_2 through first-level facility j_1. The TUFLP can be formulated as

		Uncapacitated		Capacitated	
		Heuristics	Exact	Heuristics	Exact
MUFLP	$k = 2$	[69, 99, 128, 149, 169]	[3, 25, 48, 90, 103, 118, 145]	[7, 60, 81, 108, 129, 140, 141, 166]	[1, 2, 6, 26, 108, 119, 140, 141, 159, 164, 166]
	$k > 2$	[4, 9, 10, 15, 27–29, 55, 64, 66, 80, 94, 99, 106, 116, 117, 122, 125, 136, 153, 160, 161, 170]	[66, 97, 136, 153]	[9, 28, 35, 53]	
MUFLP-E	$k = 2$	[21, 67]	[21, 34, 67, 70, 71]	[73, 88, 156]	[73, 88, 157]
	$k > 2$		[137]		
MFLDP	$k = 2$	[128]	[18, 34]	[128]	[34]
	$k > 2$	[51, 64, 89]			

Table 2.3: Summary of MLFLPs references

$$
\begin{aligned}
\text{(F1-TUFLP)}\quad \text{minimize} \quad & \sum_{i\in I}\sum_{j_1\in V_1}\sum_{j_2\in V_2} c_{ij_1j_2}x_{ij_1j_2} + \sum_{r=1}^{2}\sum_{j_r\in V_r} f_{j_r}y_{j_r} \\
\text{subject to} \quad & \sum_{j_2\in V_2}\sum_{j_1\in V_1} x_{ij_1j_2} = 1 && i\in I && (2.1)\\
& \sum_{j_1\in V_1} x_{ij_1j_2} \le y_{j_2} && i\in I,\ j_2\in V_2 && (2.2)\\
& \sum_{j_2\in V_2} x_{ij_1j_2} \le y_{j_1} && i\in I,\ j_1\in V_1 && (2.3)\\
& x_{ij_1j_2} \ge 0 && i\in I,\ j_1\in V_1,\ j_2\in V_2 && (2.4)\\
& y_{j_r}\in\{0,1\} && j_r\in V_r,\quad r=1,2. && (2.5)
\end{aligned}
$$

Note that the variables $x_{ij_1j_2}$ are allowed to be declared as continuous since in the uncapacitated case they will take integer values in any case [3]. However, the earliest version of the problem that we were able to identify is a slightly different variant which was denoted as the warehouse and plant location problem in the seminal work of Kaufman et al. [90]. Assuming that $V_2 \subseteq V_1$, the authors imposed the additional constraint that with each open plant there must be an open warehouse in the same location:

$$y_{j_2} \leq y_{j_1} \qquad j_1 \in V_1, j_2 \in V_2.$$

A few years later, Ro and Tcha [145] introduced a modified version of this constraint by including a set of "adjunct" warehouses to each plant, thus enforcing the constraint that if a plant is opened the associated warehouses are opened, but not vice versa. When the sets of adjunct warehouses are empty, the problem corresponds to what we call the TUFLP. The same year, Tcha and Lee [153] presented a problem without this additional constraint, which is then a TUFLP. These authors also generalized the problem to k levels and denoted it as the MUFLP. They introduced an MILP for the MUFLP which is nowadays referred to as *path-based formulation* (PBF), where each sequence of facilities $j_1, j_2, \cdots, j_k$, with $j_r \in V_r$, is called a path, and every customer must be allocated to a path of open facilities. It is straightforward to derive the corresponding MILP formulation for the MUFLP by extending the decision variables y_{j_r} and $x_{ij_1 \cdots j_k}$ for $r = 1, \cdots, k$ from those of the F1-TUFLP. We thus select the MUFLP as the representative problem for those MLFLPs encompassed in this category. We divide this section into the uncapacitated and the capacitated cases. The former is in turn divided into three parts namely, formulations, exact algorithms and heuristics.

2.4.1 Uncapacitated case

As for the single-level case, we follow the uncapacitated/capacitated criterion for MLFLPs. This distinction is important since capacity constraints usually play a major role in models and algorithms.

2.4.1.1 Formulations

Two main families of MILP formulations are commonly used for MLFLPs. The first is related to the PBF explained before that extends from the F1-TUFLP using variables $x_{ij_1\cdots j_k}$ for the allocation of customers, and y_{j_r} variables for selecting facilities. The second type of formulation is the so-called *arc-based formulation* (ABF). In contrast with PBF, in an ABF the decision variables are in a sense split between levels, that is, the variables are associated with arcs instead of paths. At this point it is important to note that a path in k levels (a sequence of k facilities, one from each level), actually coincides with arcs in the two-level case. Also, we refer to arcs and edges indistinctly unless otherwise needed. Therefore, the initial formulation F1-TUFLP can be viewed as a PBF or as an ABF, from which the extended versions are derived. For example, Gabor and van Ommeren [66] introduced the following MILP for the MUFLP in which decision variables are associated to arcs instead of paths. Assuming that the sets E_r contain all possible edges between levels r and $r+1$, for $r = 1, \cdots, k-1$ and that $c_{ij_1\cdots j_k} = c_{ij_1} + \cdots + c_{j_{k-1}j_k}$, let $z^r_{iab} = 1$ if customer i uses the edge $\{a, b\} \in E_r$, y_{j_r} as defined before and $v_{ij_1} = 1$ if customer i is assigned to $j_1 \in V_1$. An ABF for the MUFLP is then (F2-MUFLP)

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in I} \sum_{j_1 \in V_1} c_{ij_1} v_{ij_1} + \sum_{i \in I} \quad \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} c_{ab} z_{iab}^r + \sum_{r=1}^{k} \sum_{j_r \in V_r} f_{j_r} y_{j_r} \\
\text{subject to} \quad & \sum_{j_1 \in V_1} v_{ij_1} = 1 && i \in I && (2.6) \\
& \sum_{b \in V_2} z_{ij_1 b}^1 = v_{ij_1} && \{i, j_1\} \in E_0 && (2.7) \\
& \sum_{b \in V_{r+1}} z_{iab}^r = \sum_{b' \in V_{r-1}} z_{ib'a}^{r-1} && i \in I,\ a \in V_r,\ r = 2, \cdots, k-1 && (2.8) \\
& v_{ij_1} \leq y_{j_1} && \{i, j_1\} \in E_0 && (2.9) \\
& \sum_{a \in V_{r-1}} z_{iab}^{r-1} \leq y_b && i \in I,\ b \in V_r,\ r = 2, \cdots, k && (2.10) \\
& v_{ij_1} \geq 0 && \{i, j_1\} \in E_0 && (2.11) \\
& z_{iab}^r \geq 0 && i \in I,\ \{a, b\} \in E_r,\ r = 1, \cdots, k-1 && (2.12) \\
& y_{j_r} \in \{0, 1\} && j_r \in V_r,\ r = 1, \cdots, k. && (2.13)
\end{aligned}
$$

Note that the variables v_{ij_1} can be eliminated from the model either by consolidating the sets of constraints (2.6) and (2.7) or by setting $z_{iij_1} = v_{ij_1}$, in which case (2.7) can be embedded in (2.8). However, we have opted to reproduce the model as presented in [66]. When $k = 2$ we obtain the initial formulation F1-TUFLP described above.

Other ABFs have been studied for the problem, in particular those that consider variables $w_{j_r j_{r+1}}$ representing the flow from facility $j_r \in V_r$ to facility $j_{r+1} \in V_{r+1}$, with $V_0 = I$. As we will discuss, this type of formulation is common in the capacitated cases. One example in the uncapacitated variant arises in the seminal work of Aardal et al. [3], which introduces an ABF for the TUFLP by defining $w_{ij_1} = \sum_{j_2 \in V_2} x_{ij_1 j_2}$ and $w_{j_1 j_2} = \sum_{i \in I} x_{ij_1 j_2}$. The authors compared the LP relaxations of the two formulations concluding that the bound of the F1-TUFLP formulation is always better than that obtained with their ABF. This result typically generalizes to k levels, but the size of a PBF grows much faster than that of an ABF. Aardal et al. [3] also conducted a polyhedral study of the associated polytope of F1-TUFLP. In particular, they developed a characterization of the extreme points of its LP relaxation as well as results extending all nontrivial facets of the single-level UFLP to the TUFLP. They proved, among other results, that (2.2)–(2.4) define facets of the convex hull of the associated polyhedral set of F1-TUFLP. Moreover, they introduced two classes of facet-defining inequalities for a modified version of the F1-TUFLP and stated conditions under which these inequalities also induce facets for the single-level case. However, these results have never been extended to the case $k > 2$.

Another example of an ABF using the variables $w_{j_r j_{r+1}}$ for the TUFLP was studied by Marín [118]. Landete and Marín [103] also used the disaggregated version of constraints (2.2) and (2.3) and introduced a reformulation of the TUFLP as a set packing problem for which the corresponding polyhedral study was developed, along with facet-defining inequalities and an algorithm. More recently, Kratica et al. [97] and Marić et al. [117] independently introduced a new ABF for the MUFLP, very much in the spirit of the ABF introduced in [3] for the two-level case. Kratica et al. [97] provided computational results comparing on general purpose solvers the performance of the new formulation with those of the PBF and of the F2-MUFLP.

Ortiz-Astorquiza et al. [136] presented a new type of MILP for a slightly more general MUFLP in which for given values of p_r, $r = 1, \cdots, k$, cardinality constraints ($\sum_{j_r \in V_r} y_{j_r} \leq p_r$) are imposed at each level. They called this problem the multi-level uncapacitated p-location problem (MUpLP) since it generalizes the well-known UpLP presented by Cornuéjols et al. [45], which in turn subsumes the UFLP and the p-median problem (p-MP) [83]. The multi-level version of the p-median problem is denoted by MpMP. In [136], the authors developed the new formulation of the maximization version of the problem based on an alternative combinatorial representation given in [135], in which the objective function satisfies the submodularity property. Thus, considering the variables η^i representing the profit (or cost) of serving customer $i \in I$, and Q the set of all possible paths $q = j_1, \cdots j_k$ having exactly one facility at each level, one can project out the variables x_{iq} to x_q from the PBF and obtain the submodular formulation (SF-MUpLP)

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in I} \eta^i \\
\text{subject to} \quad & \eta^i \leq c_{iq_t} + \sum_{q \in Q} (c_{iq} - c_{iq_t})^+ x_q && i \in I, \quad t = 0, \cdots, |Q| - 1, && (2.14) \\
& \sum_{q \in Q : j_r \in q} x_q \leq M_r y_{j_r} && j_r \in V_r, \quad r = 1, \cdots, k, && (2.15) \\
& \sum_{j_r \in V_r} y_{j_r} \leq p_r && r = 1, \cdots, k, && (2.16) \\
& x_q \in \{0, 1\} && q \in Q, && (2.17) \\
& y_{j_r} \in \{0, 1\} && j_r \in V_r, \quad r = 1, \cdots, k, && (2.18)
\end{aligned}$$

where for each $i \in I$, and $r = 1, \cdots, k$, $0 = c_{iq_0} \leq c_{iq_1} \leq \cdots \leq c_{iq_{|Q|}}$ and M_r are sufficiently large numbers. Note that the disaggregated version of constraints (2.16) may also be used. In [136] a computational comparison of formulations for the MUpLP was carried out. Because of the large number of constraints (2.14) the authors embedded the SF-MUpLP in a branch-and-cut framework exploiting an efficient way of solving the separation problem. For comparison purposes, they considered the PBF extension of the F1-TUFLP, the F2-MUFLP, the branch-and-cut SF-MUpLP and the ABF of [97]. Their results showed that when the cardinality constraints are predominant, the SF-MUpLP dominates the other three formulations in terms of CPU time spent to obtain the optimal solution, while in the case of $p_r = |V_r|$, i.e. for the MUFLP, there is no clear dominance of one model over the others. While the PBF grows considerably faster when $k > 2$, it is the one that yields the best LP bound, and therefore a smaller enumeration tree. The F2-MUFLP modified for the general problem seems to take much longer when the cardinality constraints are active, but it is rather efficient in the MUFLP case. The ABF of [97] and the submodular formulation achieve a balance between memory usage and computing time spent when the problem is more general. However, the experiments pointed to a better average performance for the submodular formulation. Instances with up to 2,000 customers, 200 potential facilities, and four levels of hierarchy were solved to optimality.

2.4.1.2 Exact algorithms

All of the early works on MLFLPs introduced exact algorithms for different versions of the problem. For example, the three ground-setting papers [90, 145, 153] presented branch-and-bound methods that extended those known for the single-level case. However, the algorithms of [145, 153] were based on the assumption that the submodular property extends directly from the single-level version. The correctness of such methods was later discussed by Barros and Labbé [20] who showed that the objective function of the representation of the corresponding problems does not satisfy this property. More recently, Ortiz-Astorquiza et al. [135] introduced an alternative combinatorial representation of the (maximization version of the problem) whose objective function does satisfies submodularity.

Tcha and Lee [153] presented a modified version for the MUFLP of the dual ascent procedure of Erlenkotter [57] known for the single-level UFLP. However, ever since these solution methods were proposed, only a few papers have dealt with the development of specialized exact solution algorithms for variants of the MUFLP. The papers of Marín [118], Landete and Marín [103], Gendron et al. [70] and Ortiz-Astorquiza et al. [137] are perhaps the exceptions. However, [70, 137] study the more general case where fixed costs on the edges are considered, so these contributions will be discussed in the following sections.

As already mentioned, an ABF for the TUFLP was proposed in [118] from which the authors studied several Lagrangian relaxations. The authors showed when the so-called Lagrangian bound satisfied the integrality property, that is, the case when the optimal value obtained from the Lagrangian dual coincides with that of the LP relaxation. Moreover, they presented several results in which dominance relationships between bounds of the different relaxations are given, and developed a bounding procedure based on the lower bounds obtained by applying a subgradient optimization procedure for one of the Lagrangian relaxations. They argued for the selection of a relaxation based on a balance between dominance and ease of solution. Landete and Marín [103] reformulated the TUFLP as a set packing problem and presented different classes of facet-defining inequalities for the reformulation. Based on these inequalities, they developed a branch-and-cut algorithm and compared its performance with that of a general purpose solver.

2.4.1.3 Heuristics

Most research efforts towards the development of algorithms for MUFLP-related problems have focused on heuristics. We can start by differentiating two main research streams: heuristics without a performance guarantee, and ρ-approximation algorithms i.e., polynomial-time heuristics that yield a feasible solution with an objective function value lying within a factor of ρ of the optimal value. Most of the work on heuristics has focused on the latter stream.

In the area of heuristics without a bounded worst-case ratio, Korać et al. [94], Marić [116], Marić et al. [117] presented algorithms for the MUFLP considering that the costs c are metric (i.e. nonnegative, symmetric and satisfying the triangle inequality) and additive with respect to the k levels. In [116] a genetic algorithm is presented including an implementation with a dynamic programming scheme to find the sequences of open facilities to satisfy customers demands. According to the authors, the dynamic programming component is the main ingredient that enables the genetic algorithm to solve large-scale instances within a short amount of time. Later, in [94, 117] improvements on the genetic algorithm were introduced. For instance, improving the implementation of the dynamic programming approach and incorporating local search procedures designed for the MUFLP, which are denoted as memetic algorithms. Another memetic algorithm was designed by Mišković and Stanimirović [128] to obtain solutions of the TUFLP using the formulation introduced in [103]. Gendron et al. [69] developed a metaheuristic for the two-level uncapacitated facility location problem with single assignment constraints, denoted TUFLP-S. In the TUFLP-S the restrictions that each open first-level facility can be connected to at most one open second-level facility are required. The authors developed what they called a multi-layer variable neighborhood search metaheuristic for the TUFLP-S and a similar variant with modular costs. The term multi-layer comes from partitioning the neighborhood structures into several layers, where for each layer a variable neighborhood search scheme is applied. They compared the performance of their algorithm with that of a MILP solved using a general purpose solver, and with that of a slope-scaling heuristic based on the same formulation.

There also exist approximation algorithms with performance guarantee. However, since there are two versions of the MUFLP and its variants namely, a maximization and a minimization version, we must review them separately. The reason for this additional classification comes from the fact that from an approximation perspective, the maximization and minimization versions of an optimization problem are not necessarily comparable. This was discussed in [86, 149] for the single-level case and in [136, 170] for the multi-level case.

a) *Maximization version*

We note that in the maximization version of the MUFLP, the values of $c_{ij_1 \cdots j_k}$ correspond to the profit of serving customer i through path $j_1 \cdots j_k$. This can be thought as $c_{iq} = b_i - D_{iq}$, where b_i is the price that client $i \in I$ pays for the service, and D_{iq} is the total operational cost of serving client i through path $q = j_1 \cdots j_k$. Observe also that adding to c_{iq} a constant γ_i for every possible path q does not change the optimal solution. This is because in the MUFLP one must serve every client and thus, having new values of c defined as $c'_{iq} = c_{iq} + \gamma_i$ changes the cost of every feasible solution by the same amount. This property is well known for the single-level case [46]. The price b_i can thus be seen as the corresponding constant γ_i and therefore, only the costs are relevant for the decision, yielding the minimization version of the problem. This is why from an optimization point of view, it seems to be more common to work with the minimization version than with its maximization counterpart. Moreover, note that the objective function

$$z = \sum_{i \in I} \sum_{r=1}^{k} \sum_{j_r \in V_r} c_{ij_1 \cdots j_k} x_{ij_1 \cdots j_k} - \sum_{r=1}^{k} \sum_{j_r \in V_r} f_{j_r} y_{j_r}$$

can take positive or negative values and thus, a correcting factor in the definition of measure of relative deviation for approximate solutions must be added [45].

Let z^* be the optimal value of the problem and let z_R be a sufficiently small number, typically defined depending on the input of the problem, such that $z^* \leq z \leq z_R$, where z is the value of any feasible solution. Bumb [27] assumed that all costs and profits are non-negative and presented an approximation algorithm based on the technique of independently randomized rounding which yields a solution Z satisfying

$$\frac{Z - z_R}{z^* - z_R} \geq 0.47.$$

This worst-case bound was soon improved to 0.5 by Zhang and Ye [170]. Recently, based on an alternative representation of the MUFLP [135] in which the objective function satisfies the submodularity property, Ortiz-Astorquiza et al. [136] were able to extend to the k-level case the constant-performance guarantees of Cornuéjols et al.

[45] and Nemhauser et al. [134] derived for the single-level case. In [136], they studied the MUpLP which includes as special cases the MUFLP and the Mp-MP. The authors showed that when the profits c are additive, a polynomial time greedy algorithm always yields a solution satisfying

$$\frac{Z - z_R}{z^* - z_R} \geq 1 - \frac{1}{e} \approx 0.63.$$

Based on the foreseen difficulties of extending their algorithm to the general case of k levels, Bumb [27] questioned whether there exists an approximation algorithm with performance guarantee independent on the number of levels for the maximization version, as was the case at the time for the minimization counterpart. The recent result of [136] answers this question in a positively manner.

b) *Minimization version*

Since almost all the related papers assumed that the costs c are induced by a metric on $V \cup I$ and are additive with respect to the levels as already mentioned, in the remainder of this section we retain these assumptions unless otherwise stated. We observe that Shmoys et al. [149] and Aardal et al. [4] were the first to present approximation algorithms with constant-performance guarantees for the two-level and multi-level cases, respectively. These papers set the ground for a rich line of research. In [149] a 3.16-approximation algorithm was introduced which was soon improved in [4] to a 3-approximation algorithm for the general k-level case. However, the drawback of these algorithms seems to be that they are based on randomized rounding of the optimal solution of an LP relaxation. Even if the algorithms have polynomially bounded running times, the LP relaxation contains an exponential number of variables and thus, solving it may be difficult in practice. Guha et al. [80], Meyerson et al. [125] were the first to design efficient combinatorial algorithms capable of finding a solution within a factor of $O(\log|I|)$ and 9.2 of the optimal value, respectively. They presented these results for the MUFLP as a special case of more general network design problems. These worst-case bounds were improved by Bumb and Kern [28] who developed a dual ascent algorithm for the MUFLP with a performance guarantee of 6, and by Ageev [9] and Ageev et al. [10], using the result of Edwards [55] who proved that any ρ-approximation algorithm for the UFLP leads to a 3ρ-approximation algorithm for the MUFLP. This yielded combinatorial 4.83- and 3.27-approximation algorithms for any $k \geq 2$ with worst-case bounds of 2.8446 and 3.1678 for $k = 2$ and $k = 3$, respectively. Zhang [169] later combined techniques such as randomized rounding, dual fitting and a greedy procedure to obtain the best-to-date 1.77-approximation algorithm for the TUFLP. Moreover, the author also obtained an $O(\ln|I|)$-approximation algorithm for the non-metric TUFLP. In the same year, Fleischer et al. [64] published their results which consisted of an $O(\ln^k|I|)$-approximation algorithm for the non-metric MUFLP. A few years later, Gabor and van Ommeren [66] described a 3-approximation algorithm for the MUFLP based on LP-rounding using a new MILP formulation. The importance of this model lies in its polynomial number of variables and constraints in contrast with the previous formulation by [4]. Finally, since many techniques used for the development of such algorithms extend from those applied to the single-level versions, a natural question is whether the MUFLP is computationally harder than the UFLP. This question remained open until recently when Krishnaswamy and Sviridenko [99] proved inapproximability results which showed that there exists no approximation algorithm with performance guarantee better than 1.539 for the TUFLP unless $P = NP$. They also showed that for the general case of $k > 2$, when k tends to infinity, the hardness factor is 1.61.

Similarly, approximation algorithms with performance guarantee were developed for variants of the MUFLP. For example, Wang et al. [160, 161] proposed a 4-approximation algorithm based on LP-rounding techniques for the stochastic MUFLP, that is, when demands are uncertain. Melo et al. [122] improved the performance guarantee to $4 - o(1)$. Another variant of the MUFLP that has received attention in the last few years is the so-called MUFLP with penalties [15, 29, 106], in which the decision maker determines whether to provide the service to each customer or to pay a penalty for those that are not served. In particular, Byrka et al. [29] presented the MUFLP as a special case and provides the best known constant-performance guarantee for $k > 2$ which tends to 3 when k is sufficiently large.

2.4.2 Capacitated case

Several features of single-level FLPs, including applications, solution methods and variants of the problems have been extended to MLFLPs. However, since the number of publications suggests that the uncapacitated cases have attracted more attention, defining streams of research for the capacitated case seems more challenging. Here we discuss the main contributions corresponding to the capacitated variant of the TUFLP, called the two-level capacitated facility location problem (TCFLP). Analogously to the uncapacitated case, it seems that the TCFLP is the one that has been the most studied among capacitated MLFLPs. In this problem, capacities in one or both levels of facilities are imposed, denoted by α_{j_2} and β_{j_1}. From the early works on the TCFLP we note that of Aardal [1], who presented MILP formulation for the problem and a polyhedral study. The same author [2] later introduced a reformulation along with computational results. In contrast with the TUFLP, in this case a more common formulation involves ABF or also called two-index formulations. Let h_i be the demand value for each $i \in I$, and let v_{ij_1} and $t_{j_1j_2}$ be continuous variables representing the flow to customer i from j_1 and that of the plant j_2 to warehouse j_1, respectively. Denoting by c_{ij_1} and $c_{j_1j_2}$ the unit transportation costs from i to j_1 and from j_1 to j_2, respectively, the TCFLP can be formulated as

$$
\begin{aligned}
\text{(TCFLP)}\quad \text{minimize}\quad & \sum_{i\in I}\sum_{j_1\in V_1} c_{ij_1}v_{ij_1} + \sum_{j_1\in V_1}\sum_{j_2\in V_2} c_{j_1j_2}t_{j_1j_2} + \sum_{r=1}^{2}\sum_{j_r\in V_r} f_{j_r}y_{j_r} \\
\text{subject to}\quad & \sum_{j_1\in V_1} v_{ij_1} \geq h_i && i\in I && (2.19)\\
& \sum_{j_2\in V_2} t_{j_1j_2} \geq \sum_{i\in I} v_{ij_1} && j_1\in V_1 && (2.20)\\
& \sum_{j_1\in V_1} t_{j_1j_2} \leq \alpha_{j_2}y_{j_2} && j_2\in V_2 && (2.21)\\
& \sum_{j_2\in V_2} t_{j_1j_2} \leq \beta_{j_1}y_{j_1} && j_1\in V_1 && (2.22)\\
& y_{j_r}\in\{0,1\} && j_r\in V_r,\ r=1,2 && (2.23)\\
& v_{ij_1}\geq 0,\ t_{j_1,j_2}\geq 0 && i\in I,\ j_1\in V_1,\ j_2\in V_2. && (2.24)
\end{aligned}
$$

Marín and Pelegrín [119] compared a two-index and a three-index formulation for the development of an exact algorithm for the TCFLP based on Lagrangian relaxations. More recently, Litvinchev and Ozuna Espinosa [108], Wildbore [164] developed exact and approximate algorithms mainly based on Lagrangian relaxations along with the corresponding computational results obtained for the TCFLP. Fernandes et al. [60] introduced a genetic algorithm, while Guo et al. [81] proposed a hybrid evolutionary algorithm for the same version of the problem. Chen and Wang [35] designed an approximation algorithm for the general k-level version. As in the uncapacitated case, assuming that the values of c are induced by a metric, for k levels, Ageev [9], Bumb and Kern [28], and Du et al. [53] developed ρ-approximation algorithms with values of ρ equal to 12, 9 and $k+2+\sqrt{k^2+2k+5}+\epsilon$, respectively.

Other authors have studied some variations of the TCFLP. For example, Bloemhof-Ruwaard et al. [26] solved a slightly different version of the problem in the context of a waste disposal system, while Pirkul and Jayaraman [140] presented a MILP formulation and heuristic methods for a multi-commodity, single-source TCFLP in which cardinality constraints are imposed at both levels. Pirkul and Jayaraman [141] considered the case without the single-source requirements. Yet another variant introduced by Addis et al. [6, 7] is the TCFLP with single source constraints at both levels and dimensioning of the facilities, that is, with modular capacities. The authors provided exact and heuristic algorithms to solve instances with up to 200 customers and 50 potential sites of facilities. A similar version of the problem was studied by Wu et al. [166] who developed a Lagrangian relaxation-based procedure. Finally, Wang and Yang [159] and [129] considered variations of the TCFLP under uncertainty.

2.5 MLFLPs with network design and tactical decisions

In the early 1990s, Gao and Robinson [67] introduced the term *echelon* in the context of MLFLPs by presenting a new MILP for a variant of the TUFLP, denoted as two-echelon uncapacitated facility location problem (TEUFLP). This formulation was mainly motivated by the desire to extend the dual adjustment procedures of [57] for the TUFLP, which Tcha and Lee [153] had previously been unable to achieve. This modification of the problem can be viewed as if the fixed costs for opening warehouses also depend on the plants from which they are served, that is, there is a fixed cost associated with each pair of facilities from levels one and two, i.e. operating together. Equivalently, this variant can be seen as having fixed costs for opening edges between facilities of different levels associated with the selection of facilities at the first level. Thus, the authors consider $d^1_{j_1 j_2}$ to be the fixed costs for opening warehouse j_1 and supplying it from plant j_2, and the binary variables $w^1_{j_1 j_2} = 1$ if j_1 is opened and simultaneously served from j_2. The TEUFLP can be formulated as

$$
\begin{aligned}
\text{(F1-TEUFLP)} \quad \text{minimize} \quad & \sum_{i\in I}\sum_{j_1\in V_1}\sum_{j_2\in V_2} c_{ij_1j_2}x_{ij_1j_2} + \sum_{j_2\in V_2} f_{j_2}y_{j_2} + \sum_{j_2\in V_2}\sum_{j_1\in V_1} d_{j_1j_2}w^1_{j_1j_2} \\
& (2.1), (2.4) \\
& w^1_{j_1j_2} \le y_{j_2} \qquad j_1\in V_1,\ j_2\in V_2, \qquad (2.25) \\
& x_{ij_1j_2} \le w^1_{j_1j_2} \qquad i\in I,\ j_1\in V_1,\ j_2\in V_2, \qquad (2.26) \\
& y_{j_2} \in \{0,1\} \qquad j_2\in V_2, \qquad (2.27) \\
& w^1_{j_1j_2} \in \{0,1\} \qquad j_1\in V_1,\ j_2\in V_2. \qquad (2.28)
\end{aligned}
$$

Soon after, Barros and Labbé [21] introduced a general version of the problem that subsumes both the TUFLP and the TEUFLP. The authors considered fixed costs associated with opening facilities at both levels as well as those for activating edges between facilities of different levels. Fixed costs for opening edges between customers and first-level facilities are not considered in the problem. We denote this variant as the TUFLP-E. The authors discussed three variants of a MILP formulation for the problem and studied the relationships between the corresponding LP relaxations. The TUFLP-E of [21] is formulated as

$$
\begin{aligned}
\text{(F1-TUFLP-E)} \quad \text{maximize} \quad & \sum_{i\in I}\sum_{j_1\in V_1}\sum_{j_2\in V_2} c_{ij_1j_2}x_{ij_1j_2} - \sum_{r=1}^{2}\sum_{j_r\in V_r} f_{j_r}y_{j_r} - \sum_{j_2\in V_2}\sum_{j_1\in V_1} d^1_{j_1j_2}w^1_{j_1j_2} \\
\text{subject to} \quad & (2.1), (2.4), (2.5), (2.25), (2.26), (2.28) \\
& w^1_{j_1j_2} \le y_{j_1} \qquad j_1\in V_1,\ j_2\in V_2, \qquad (2.29)
\end{aligned}
$$

or equivalently, exchanging constraints (2.25) and (2.29) by (2.2) and (2.3) when the fixed costs are non-negative. They proved in [21] that using the sets of constraints (2.2) and (2.3) yields a better LP bound. We also note that the authors formulated the problem in the maximization form, perhaps for the first time for MLFLPs. The values of $c_{ij_1j_2}$ in this case actually correspond to profits, instead of transportation costs. Indeed, as discussed by Cornuéjols et al. [46] for the single-level and by Ortiz-Astorquiza et al. [136] for the multi-level case, the maximization and minimization versions of the UFLP are equivalent from an optimization point of view. However, from an algorithmic perspective, especially for approximation algorithms, this is not the case as mentioned in Section 2.4.1.3. This version of the problem can be generalized to the case of k levels, even including link activation costs between customers and the set V_1. We denote it as MUFLP-E, which is the representative problem of this category.

We next review the main contributions to MLFLPs that involve non-trivial network design and tactical decisions. We divide this section into the uncapacitated and capacitated cases.

2.5.1 Uncapacitated case

Barros and Labbé [21] seem to have been the first to study the general version of the problem. They developed a branch-and-bound procedure using the corresponding upper and lower bounds obtained from different Lagrangian relaxations of two of the formulations discussed, and those obtained from an extension of the greedy heuristic proposed for the UFLP. This method also benefits from the efficient solution of a particular Lagrangian relaxation which coincides with a min-cut problem. The authors presented comparative computational results with the previous special cases of [67] and [153]. They observed that solving the proposed Lagrangian relaxations provides an easier way to obtain better bounds than those yielded by the modified dual ascent method for MLFLPs. However, this category of MLFLPs was put aside for some years until very recently when Gendron et al. [70, 71] studied a more general version of the problem of Chardaire et al. [34]. The MLFLP studied in [34] considers only design costs in a two-level FLP where single assignment constraints between levels of facilities are imposed. The problem addressed in [70] additionally includes transportation costs c since it appears as a subproblem in a more sophisticated MLFLP in the context of freight transportation [68]. We refer to this variant as the TUFLP with edge costs and single assignment constraints (TUFLP-E-S) which can be formulated as F1-TUFLP-E-S

$$
\begin{aligned}
\text{minimize} \quad & \sum_{i \in I} \sum_{j_1 \in V_1} \sum_{j_2 \in V_2} c_{ij_1j_2} x_{ij_1j_2} \qquad + \sum_{r=1}^{2} \sum_{j_r \in V_r} f_{j_r} y_{j_r} + \sum_{j_2 \in V_2} \sum_{j_1 \in V_1} d^1_{j_1j_2} w^1_{j_1j_2} \\
\text{subject to} \quad & \sum_{j_2 \in V_2} \sum_{j_1 \in V_1} x_{ij_1j_2} = 1 \qquad i \in I \qquad (2.30) \\
& \sum_{j_1 \in V_1} x_{ij_1j_2} \le y_{j_2} \qquad i \in I,\ j_2 \in V_2, \qquad (2.31) \\
& \sum_{j_2 \in V_2} x_{ij_1j_2} \le y_{j_1} \qquad i \in I,\ j_1 \in V_1, \qquad (2.32) \\
& x_{ij_1j_2} \le w^1_{j_1j_2} \qquad i \in I,\ j_1 \in V_1,\ j_2 \in V_2, \qquad (2.33) \\
& \sum_{j_2 \in V_2} w^1_{j_1j_2} \le 1 \qquad j_1 \in V_1, \qquad (2.34) \\
& w^1_{j_1j_2} \le y_{j_1} \qquad j_1 \in V_1,\ j_2 \in V_2 \qquad (2.35) \\
& x_{ij_1j_2} \ge 0 \qquad i \in I,\ j_1 \in V_1,\ j_2 \in V_2, \qquad (2.36) \\
& y_{j_r} \in \{0,1\} \qquad j_r \in V_r, \quad r = 1,2. \qquad (2.37)
\end{aligned}
$$

As noted in [70], if the fixed costs are non-negative, one can project out the variables y_{j_1} for $j_1 \in V_1$ based on the set of constraints (2.34). Thus, $y_{j_1} = \sum_{j_2 \in V_2} w^1_{j_1j_2}$ and the fixed costs f_{j_1} can be embedded within the new edge costs $l_{j_1j_2} = d^1_{j_1j_2} + f_{j_2}$ for each $j_1 \in V_1$. Constraints (2.35) are actually redundant but allow relaxing the integrality conditions on the y_{j_2} variables in addition to improving the LP bound. After projecting out the variables y_{j_1} the objective function becomes

$$\text{minimize} \quad \sum_{i\in I}\sum_{j_1\in V_1}\sum_{j_2\in V_2} c_{ij_1j_2}x_{ij_1j_2} + \sum_{j_2\in V_2} f_{j_2}y_{j_2} + \sum_{j_2\in V_2}\sum_{j_1\in V_1} l_{j_1j_2}w^1_{j_1j_2},$$

which coincides with that of the TEUFLP described in [67]. Also, when the set-up costs on the edges $d^1_{j_1j_2}$ are zero, we obtain the TUFLP-S version of the problem. However, in this case the single assignment constraints can be dropped under some conditions on the costs $c_{ij_1j_2}$ [70, 136], yielding a class of instances for which the TUFLP and the TUFLP-E are equivalent. In [70] a branch-and-bound procedure is also developed based on specialized branching rules and a Lagrangian relaxation that was not previously studied in [21, 34].

All the papers relating to MUFLP-E mentioned so far consider the two-level version of the problem. Ortiz-Astorquiza et al. [137] recently introduced a general k-level setting where all three types of costs are considered and cardinality constraints are imposed at each level. This problem is denoted as MUpLP-E. In comparison with the other two categories of MLFLP, little research has been carried out in this category, especially in what regards the development of exact solution methods. These authors developed an exact Benders-based algorithm decomposition scheme for the solution of large-scale instances. The algorithm exploits the structure of the extended F2-MUFLP formulation for the MUpLP-E in which the subproblems can be efficiently solved. The authors conducted an extensive computational study on the impact of different variations of the Benders decomposition procedure, such as implementing Pareto-optimal cuts or using alternative feasibility cuts.

2.5.2 Capacitated case

The first articles to consider a TCFLP with fixed costs for opening edges are [156, 157]. Tragantalerngsak et al. [156] developed several Lagrangian heuristics for a two-level CFLP with single source constraints (single assignment in the TUFLP) and capacities at the first-level facilities only. The MILP formulation described by the authors resembles that of the TEUFLP of [67], where fixed costs on the edges between facilities of different levels replace those of opening facilities in one level. In a sequel paper [157], the same authors presented an exact algorithm for the problem based on the previous Lagrangian relaxations. We refer to this problem as the TECFLP-S. It can be formulated as

$$\text{(TECFLP-S) minimize} \sum_{i\in I}\sum_{j_1\in V_1}\sum_{j_2\in V_2} c_{ij_1j_2}x_{ij_1j_2} + \sum_{j_1\in V_1}\sum_{j_2\in V_2} d_{j_1j_2}w_{j_1j_2}\sum_{j_2\in V_2} f_{j_2}y_{j_2}$$

$$\text{subject to} \sum_{j_1\in V_1}\sum_{j_2\in V_2} x_{ij_1j_2} = 1 \qquad i\in I \qquad (2.38)$$

$$\sum_{i\in I} h_i x_{ij_1j_2} \le \beta_{j_1} \qquad j_1\in V_1,\ j_2\in V_2 \qquad (2.39)$$

$$\sum_{j_2\in V_2} w_{j_1j_2} \le 1 \qquad j_1\in V_1 \qquad (2.40)$$

$$x_{ij_1j_2} \le w_{j_1j_2} \qquad j_1\in V_1,\ \ j_2\in V_2 \qquad (2.41)$$

$$w_{j_1j_2} \le y_{j_2} \qquad (2.42)$$

$$x_{ij_1j_2},\ w_{j_1,j_2},\ y_{j_2}\in\{0,1\} \quad i\in I,\ \ j_1\in V_1,\ j_2\in V_2. \quad (2.43)$$

Other related problems have also been studied. These are more general and typically include additional requirements. For example, Ignacio et al. [88] presented a two-level capacitated facility location problem with edge costs (in E_0 and E_1) and single-source constraints (TCFLP-E-S) in a computer network environment. There, both levels of facilities, routers and concentrators, have capacities and fixed costs for opening facilities at the two levels are considered. The authors designed an exact solution method based on a Lagrangian relaxation an a tabu search heuristic. Another example of a TCFLP-E arises in the context of a disaster relief facility location system in [73]. Ghezavati et al. [73] considered a more general version of the problem where capacities are also imposed on the edges and studied the problem under some uncertain parameters. Finally, some related problems were addressed in [36, 68, 85]. Çinar and Yaman [36] introduced two variants of the so-called vendor location problem as special cases of capacitated MLFLPs. The work of Gendron and Semet [68] was motivated by a freight transportation problem. It set the ground for the study of different variants of the TUFLP which can be seen as a capacitated MLFLP. The authors considered a multi-commodity two-level facility location problem with single-source constraints, capacities in the arcs and modular transportation costs. Hinojosa et al. [85] studied a multi-period TCFLP.

2.6 MLFLPs with network design decisions

The last category of MLFLPs that we review is concerned with non-trivial network design decisions, but in which no tactical decisions are explicitly considered. We call the MFLDP this "design-only" version of the problem, which can also be viewed as a special case of the MUFLP-E. This problem is relevant to strategic supply chain management. In such a scenario, only the design decisions are involved through the fixed cost on facilities and edges and the allocation of customers is implicitly given by the opening of the corresponding edge. MLFLPs belonging to this category, either for two or k levels, have been studied by several authors [18, 34, 51, 64, 89]. Remarkably, with the exception of [34], none of the above references presents an exact algorithm and all date from the last decade; three of them develop approximation algorithms with performance guarantee, while [18] presents a polyhedral study for a ILP formulation. In particular, the latter paper provides three families of valid inequalities and extends to the TFLDP non-trivial facet defining inequalities for the single-level UFLP. Moreover, the authors study integrality conditions of the polytope associated with the TFLDP, that is, they introduce conditions on the graph $G = (V \cup I, E)$ so that the LP relaxation of the problem outputs an integral solution.

They also show how to determine whether a given graph G satisfies such conditions using a polynomial time algorithm developed for the single-level case. For the two-level case, we reproduce the version presented in [18] which uses the sets of arcs $A_r \subseteq V_r \times V_{r+1}$ between levels of facilities, considering $A_0 = I$, to introduce their ILP formulation:

$$\text{(F1-TFLDP) minimize} \sum_{(i,j_1)\in A_0} d^0_{ij_1} w^0_{ij_1} + \sum_{(j_1,j_2)\in A_1} d^1_{j_1j_2} w^1_{j_1j_2} + \sum_{r=1}^{2} \sum_{j_r \in V_r} f_{j_r} y_{j_r}$$

$$\sum_{(i,j_1)\in A_0} w^0_{ij_1} = 1 \qquad i \in I \tag{2.44}$$

$$w^0_{ij_1} \leq y_{j_1} \qquad (i, j_1) \in A_0, \tag{2.45}$$

$$\sum_{(j_1,j_2)\in A_1} w^1_{j_1j_2} = y_{j_1} \qquad j_1 \in V_1, \tag{2.46}$$

$$w^1_{j_1j_2} \leq y_{j_2} \qquad (j_1, j_2) \in A_1, \tag{2.47}$$

$$w^0_{ij_1}, w^1_{j_1j_2} \in \{0, 1\} \qquad (i, j_1) \in A_0,\ (j_1, j_2) \in A_1, \tag{2.48}$$

$$y_{j_r} \in \{0, 1\} \qquad j_r \in V_r,\ r = 1, 2, \tag{2.49}$$

where $w^0_{ij_1}$ and $d^0_{ij_1}$ are the decision variables and costs for opening a link between customer i and first-level facility j_1, respectively. We make two remarks on the above formulation. First, it corresponds to a more general version in which arcs are considered between levels of facilities instead of taking the sets $V_r \times V_{r+1}$. This slightly more general version of the problem could also be reproduced for the MUFLP-E. Second, constraints (2.46) ensure the allocation of open facilities of the first level to those of the second one, and also enforce single assignment for open facilities of the first level. This is important because in this case the number of edges adjacent to an open facility yields a capacitated version of the problem. This follows from the fact that there are no flow or transportation variables, but only design-type variables. For this reason we exclude the uncapacitated/capacitated subdivision from this category.

Fleischer et al. [64] developed a $\ln^k |I|$-approximation algorithm for the k-level extension of the problem. The authors use general costs h, that is, they also consider the non-metric case. Later, Drexl [51] presented a $3/2(3^k - 1)$-approximation algorithm. They assumed that the costs h are induced by a metric and that the values of f_{j_r} are non-negative. Kantor and Peleg [89] studied a similar version of the problem in which edges need to be opened as well as facilities at each of the k levels, but only in one level is there an associated fixed cost for opening facilities f_{j_k}. The authors developed a $(1 + 3^\beta)(3^{\beta+1})^{k-1}$-approximation algorithm, where $\beta \geq 1$ is a parameter used to define the values of the costs h from the distances between vertices j_r, j_{r+1} in the graph.

Finally, Chardaire et al. [34], mainly motivated from a telecommunications application, studied a variant of the two-level problem in which single-assignment constraints between levels of facilities are enforced. That is, each open facility of the first level can be assigned to at most one open facility of the second level (coherent structure in the HFLP classification). Following the notation of this article,

$$\sum_{j_2 \in V_2} w^1_{j_1 j_2} \leq 1 \qquad j_1 \in V_1.$$

The authors presented two MILP formulations and obtained lower bounds via a Lagrangian relaxation, thus improving one of the formulations with a family of facet-defining inequalities. They also developed a simulated annealing algorithm to improve the upper bounds returned by the Lagrangian relaxation. Mišković and Stanimirović [128] used the model of [34] to design a metaheuristic.

2.7 Conclusions

We have identified the main characteristics of MLFLPs, an important class of discrete location problems that has received increasing attention in the last two decades. We have pointed out the main differences and similarities with well-known related areas in an attempt to delimit the borders of this class of problems and thus the scope of the survey. In the context of MLFLPs, we have identified three main categories based on the types of decisions involved in the optimization process: MLFLPs with tactical decisions, MLFLPs with network design and tactical decisions, and MLFLPs with network design decisions only. These decisions are closely related to the types of input costs to the problem. Using this classification scheme we have presented a comprehensive review of the most relevant publications and we have discussed the variations between problems along with formulations and algorithms. We have also considered the uncapacitated/capacitated distinction to further identify where most of the efforts in the area have been expended. We first observed that with one exception [137], all papers concerned with the development of exact algorithms (or polyhedral studies) refer to the special case where $k = 2$. Thus, all contributions related to the most general versions of the problems arise from the approximate algorithms context. In particular, a large number of papers have been published on the development of approximation algorithms with performance guarantee for the MUFLP. Notably, this same category of MLFLPs is the one that has received the most attention in comparison with the other two.

Some recent publications have concentrated on different variants and extensions of the main MLFLPs. For instance, we have mentioned some articles in which uncertain parameters are included, as well as dynamic facility location problem where facilities can be opened and closed at each time period, and MLFLPs with service penalties where customers may not to be serviced. In many cases, fundamental MLFLPs arise as subproblems of these more general versions. Other sophisticated models in SCM and HFLPs also present MLFLPs as subproblems. Therefore, efficient algorithms for MLFLPs may help solve related problems. The fact that MLFLPs generalize well-known single-level FLPs while retaining several of their mathematical properties can be further exploited in the development of such algorithms. One important step towards a more systematic growth of the field is the incorporation of a common set of MLFLP instances, which would allow fairer algorithmic comparisons. Finally, we consider that MLFLPs constitute a very promising research area, not only from a theoretical and modeling point of view, but also in terms of devising efficient algorithms.

Chapter 3

Multi-level facility location as the maximization of a submodular set function

The content of this chapter is published as "Multi-level Facility Location as the Maximization of a Submodular Set Function", *European Journal of Operational Research*, 247(3), 1013-1016, 2015 [135].

Abstract

In this paper we model the multi-level uncapacitated facility location problem as two different combinatorial optimization problems. The first model is the classical representation of the problem which uses a set of vertices as combinatorial objects to represent solutions whereas in the second model we propose the use of a set of paths. An interesting observation is that the real-valued set function associated with the first combinatorial problem does not satisfy the submodular property, whereas the set function associated with the second problem does satisfy this property. This illustrates the fact that submodularity is not a property intrinsic to an optimization

problem but rather to its mathematical representation.

3.1 Introduction

In this paper we introduce an alternative combinatorial representation for the *multi-level uncapacitated facility location problem* (MUFLP) whose objective function satisfies the submodularity property. The contribution of this paper is twofold. First, we correct a previous conclusion stating that the MUFLP is not submodular and also, we illustrate the fact that submodularity is not an intrinsic property of an optimization problem but rather of the set function of its mathematical representation. To the best of our knowledge, this has not been clarified before in the literature and has led to a misuse of the terminology and to misleading results.

Generally speaking *multi-level facility location problems* (MFLPs) consist of finding the best set of facilities to open at each of the different levels (or type of facilities) in order to maximize the total profit while satisfying the demand of every customer. Several variants of MFLPs have been studied over the last four decades. Early works addressing MFLPs are those of Calvo and Marks [31] and Kaufman et al. [90], who introduced this family of discrete location problems and described mathematical programming formulations to represent them. However, since most MFLPs are NP-hard, several approximation algorithms have been developed for different versions of this family of problems [see for instance, 4, 7, 10, 55]. Moreover, even for the single level case, very recent approximation and exact algorithms for solving particular cases are presented in [5] and [105], respectively, showing the continued interest in this fundamental class of problems for the location science community. For recent overviews of models, classification criteria, solution techniques and references of Hierarchical Facility Location Problems, a more general view of MFLPs, we refer the reader to [47] and [168].

We focus our attention on a class of MFLPs originally introduced by Kaufman et al. [90] and then extended by Ro and Tcha [145] and Barros and Labbé [20]. Let $I = \{1, \cdots, m\}$ be the set of customers and $V_1, V_2, \cdots, V_k$ the sets of sites where facilities of levels 1 to k can be selected (or open) and $V = \cup_{i=1}^{k} V_i$. Also, there are fixed costs f_{j_r} associated with opening facility j_r at level r and a profit $c_{ij_1 \cdots j_k}$ obtained from allocating customer $i \in I$ to the sequence of facilities $j_1, \cdots, j_k$. The MUFLP consists of selecting a set of facilities to open at each of the k levels and of assigning each customer to a set of facilities, exactly one at each level, while maximizing the difference of the total profit minus the setup cost for opening the facilities. The MUFLP can be modeled as the following combinatorial optimization problem. For each nonempty subset $R \subseteq V$ define

$$\begin{aligned} q(R) &= -\sum_{r=1}^{k} \sum_{j_r \in R_r} f_{j_r} \\ w(R) &= \sum_{i \in I} w^i(R) = \sum_{i \in I} \max_{j_1 \in R_1, \cdots, j_k \in R_k} c_{ij_1 \cdots j_k} \end{aligned}$$

and

$$\begin{aligned} v(R) &= w(R) + q(R) \\ &= \sum_{i \in I} \max_{j_1 \in R_1, \cdots, j_k \in R_k} c_{ij_1 \cdots j_k} - \sum_{r=1}^{k} \sum_{j_r \in R_r} f_{j_r}, \end{aligned}$$

where $R = \cup_{r=1}^{k} R_r$, with $R_1 \subseteq V_1, \cdots, R_k \subseteq V_k$. The MUFLP can then be stated as the problem of selecting a set of nodes $R \subseteq V$ such that $v(R)$ is maximum, i.e.,

$$\max_{R \subseteq V} \{v(R)\}. \tag{3.1}$$

Note that when $k = 1$, the MUFLP reduces to the classical *uncapacitated facility location problem* (UFLP), a central problem in location theory [46]. A promising avenue of research has thus been the extension of well-known properties of the UFLP for the multi-level case (i.e. $k > 1$). For instance, Aardal et al. [3] show that all non-trivial facet defining inequalities for the UFLP also define facets for the MUFLP when $k = 2$. [4], [28] and [169] use ideas previously developed for the UFLP, such as dual ascent and adjustment techniques [57], in order to develop approximation algorithms for the MUFLP. Frieze [65] and Babayev [17] show that the natural set function representation $v(R)$ is submodular and Nemhauser et al. [134] show that $w(R)$ is submodular and nondecreasing when $k = 1$. We recall the definition of submodular and nondecreasing set functions [see, 134].

Let N be a finite set and f be a real-valued function defined on the set of subsets of N and $\rho_e(S) = f(S \cup \{e\}) - f(S)$ be the incremental value of adding element e to the set S when evaluating the set function f.

Definition 3.1. .

a) f *is submodular if* $\rho_e(S) \geq \rho_e(T), \quad \forall S \subseteq T \subseteq N$ *and* $e \in N \setminus T$.

b) f *is nondecreasing if* $\rho_e(S) \geq \rho_e(T) \geq 0, \quad \forall S \subseteq T \subseteq N$ *and* $e \in N$.

Although the maximization of a submodular set function is known to be NP-hard [133], the submodular property has allowed the analytic study and development of exact and approximate algorithms for several classes of optimization problems [74]. One milestone in facility location is the paper of Cornuéjols et al. [45] in which the authors use this property to obtain worst-case bounds for greedy and local improvement heuristics and for enumeration algorithms for the maximization version of the location problem. Later, a result by Feige [59] implied that the best possible approximation guarantee for the UFLP was given precisely by the greedy algorithm, unless $P = NP$. More generally, the submodularity property is used by Nemhauser et al. [134] in order to extend the results of the greedy and local heuristics for the problem of maximizing a submodular set function subject to a cardinality constraint.

Nemhauser and Wolsey [132] presented a mixed integer programming formulation and a cutting-plane algorithm to solve this class of problems based on submodularity. Wolsey [165] applied this formulation to the UFLP and discussed its connections with a Benders reformulation. Spielberg [151] studied simplification rules to reduce the number of explored nodes in an enumeration tree when solving the UFLP to optimality by branch-and-bound. More recently, Calinescu et al. [30] presented a randomized approximation algorithm with worst-case bound for the problem of maximizing a submodular function subject to an arbitrary matroid and Kulik et al. [101] introduced an approximation algorithm for the maximization of a non-decreasing submodular function subject to multiple linear constraints. Also, Gupta and Könemann [82] presented a survey of different approximation algorithms for network design, including the results of the greedy heuristic based on submodularity. Contreras and Fernández [40] showed how a general class of hub location problems can be modeled as the minimization of a supermodular set function and used this representation to develop mixed integer programming formulations and approximation algorithms to solve these problems.

A similar approach to that of Spielberg [151] is used in Ro and Tcha [145] and Tcha and Lee [153], where the submodularity property is assumed for the MUFLP, and branch-and-bound methods using similar simplification rules are presented for the two-level and the multi-level cases, respectively. However, Barros and Labbé [20] provided a counter-example showing that the submodular property does not hold for the set function $v(R)$, and thus the authors concluded that part of the results obtained in [145] and [153] were unfortunately wrong, also concluding that the MUFLP is not submodular.

3.2 An Alternative Combinatorial Representation of the MUFLP

We next provide an alternative combinatorial representation of the MUFLP whose objective set function is submodular. Note that for the case of problem (3.1), the mathematical model is defined on the set of subsets of V. Such a model arises naturally given that the MUFLP seeks to open a subset of facilities in V and to assign each customer to exactly one facility at each level. However, an alternative combinatorial representation is to use the set of paths connecting facilities between different levels to represent the *allocation paths* of customers to their allocated facility in each level.

Let $G = (V \cup I, E)$ be a graph with a vertex set $V \cup I$ partitioned into $k+1$ levels where, as before, I represents the set of customers and $V_1, V_2, \cdots, V_k$ are the sets of facilities from level 1 to level k, respectively. The set of edges E is also partitioned as $E = \{E_1, \cdots, E_k\}$, where $E_i = \{e \in E : e = (s,t) \text{ with } s \in V_{i-1} \text{ and } t \in V_i\}$ for $i = 2, \cdots, k$, and $E_1 = \{e \in E : e = (s,t) \text{ with } s \in I \text{ and } t \in V_1\}$. Without loss of generality, we assume that for each $i = 2, \cdots, k$, the graphs induced by $V_{i-1} \cup V_i$ are complete, otherwise we could complete the graph by adding edges such that the total profits of the corresponding new paths are zero. Moreover, we denote by P the set of all possible simple paths $(j_1, j_2, \cdots, j_k)$ in G starting at a vertex $j_1 \in V_1$, finishing at a vertex $j_k \in V_k$, and having exactly one element from each level. Let $N = P \cup V$ be a finite set containing both the set of paths P and the subset of vertices V of G. Also, consider the set $N_r(S)$ to be the set of vertices of level r associated with the paths of set S. Moreover, with an abuse of notation, we refer to each nonempty subset of N as the pair (S, R) thus, $(S, R) \subseteq N$, where $S \subseteq P$ and $R \subseteq V$. Note that (S, R) is not a couple but a subset of N, which we denote as a pair in order to clearly differentiate the elements taken from P and those taken from V. Now, we define

$$
\begin{aligned}
f(S,R) &= -\sum_{r=1}^{k}\sum_{j_r\in R_r} f_{j_r} \\
c(S,R) &= \sum_{i\in I} c^i(S,R) = \sum_{i\in I} \max_{(j_1,j_2,\cdots,j_k)\in S} c_{ij_1j_2\cdots j_k}
\end{aligned}
$$

and

$$
\begin{aligned}
z(S,R) &= c(S,R) + f(S,R) \\
&= \sum_{i\in I} \max_{(j_1,j_2,\cdots,j_k)\in S} c_{ij_1j_2\cdots j_k} - \sum_{r=1}^{k}\sum_{j\in R_r} f_{j_r},
\end{aligned}
$$

where, as before, $R = \cup_{r=1}^{k} R_r$, with $R_1 \subseteq V_1, \cdots, R_k \subseteq V_k$. The MUFLP can now be stated as the problem of selecting a set of paths $S \subseteq P$ and a set of nodes $R \subseteq V$ such that $z(S,R)$ is maximum, i.e.,

$$
\max_{(S,R)\subseteq N} \{z(S,R) : N_r(S) = R_r, \quad \text{for } r = 1,\cdots,k\}, \tag{3.2}
$$

where $N_r(S) = \{j \in V_r : j \text{ is a vertex of some path } p \in S\}$. Observe that the constraints of (3.2) state that for each edge $e = (j,k)$ formed from a path $p \in S$, the corresponding facilities $j \in V_{i-1}$ and $k \in V_i$ must be open.

Before proving the submodularity of $z(S,R)$, we recall some results on submodular and nondecreasing set functions [see, 134].

Lemma 3.1. *Let d_j be the weight of $j \in N$. Then, the linear set function $f(S) = -\sum_{j\in S} d_j$ is submodular.*

Lemma 3.2. *A positive linear combination of submodular functions is submodular.*

Using lemmas (3.1) and (3.2) we can now prove that the objective function of the alternative combinatorial representation for the MUFLP satisties submodularity.

Proposition 3.1.

a) $c(S,R) = \sum_{i\in I} c^i(S,R)$ *is submodular and nondecreasing.*

b) $z(S,R) = c(S,R) + f(S,R)$ *is submodular.*

Proof.

a) Let $(S,R) \subseteq (T,Q) \subseteq N$, with $S \subseteq T \subseteq P$ and $R \subseteq Q \subseteq V$, and $q = (q_1, q_2, \cdots, q_k) \in P \backslash T$. For each $i \in I$, we have

$$\begin{aligned}
\max_{(j_1,\cdots,j_k)\in S\cup\{q\}} c_{ij_1\cdots j_k} - \max_{(j_1,\cdots,j_k)\in S} c_{ij_1\cdots j_k} &= \max\{0, c_{iq_1\cdots q_k} - \max_{(j_1,\cdots,j_k)\in S} c_{ij_1\cdots j_k}\} \\
&\geq \max\{0, c_{iq_1\cdots q_k} - \max_{(j_1,\cdots,j_k)\in T} c_{ij_1\cdots j_k}\} \\
&= \max_{(j_1,\cdots,j_k)\in T\cup\{q\}} c_{ij_1\cdots j_k} - \max_{(j_1,\cdots,j_k)\in T} c_{ij_1\cdots j_k},
\end{aligned}$$

where the inequality follows from $\max_{(j_1,\cdots,j_k)\in S} c_{ij_1\cdots j_k} \leq \max_{(j_1,\cdots,j_k)\in T} c_{ij_1\cdots j_k}$. Summing over $i \in I$ and by Lemma 1(b) we obtain that $c(S,R)$ is submodular. Moreover, given that

$$\max_{(j_1,\cdots,j_k)\in T\cup\{q\}} c_{ij_1\cdots j_k} - \max_{(j_1,\cdots,j_k)\in T} c_{ij_1\cdots j_k} \geq 0,$$

for each $T \subseteq P$ and $q \in P \setminus T$, $c(S,R)$ is nondecreasing.

b) The function $f(S,R)$ is submodular as a direct consequence of Lemma 3.1 and when combined with Lemma 3.2, the submodularity of $z(S,R)$ follows.

□

The above result shows that the MUFLP can be stated as the maximization of a submodular set function subject to a set of constraints that ensure that the set of selected paths S are associated with a set of open facilities $R \subseteq V$. Observe that these constraints can be easily modeled by using a system of linear equations and thus, problem (3.2) is a particular case of the more general problem of maximizing a submodular function subject to a linear set of constraints studied by Nemhauser and Wolsey [132].

3.3 Example

To illustrate the structural differences between the vertex and the path-allocation representations of the MUFLP we use the following example introduced in [20]. Let $k = 2$, $I = \{1\}$, $V_1 = \{1_1, 2_1\}$, $V_2 = \{1_2, 2_2\}$, $f_j = 0$ for all $j \in V$ and the following profit matrix:

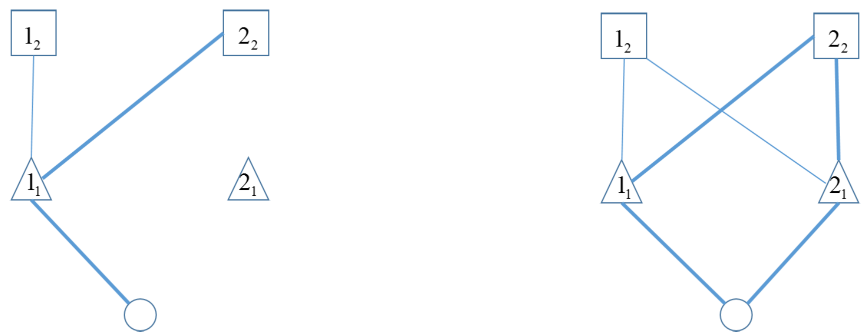
c	1_2	2_2
1_1	1	1
2_1	100	1

An optimal solution for this two-level instance is obtained when facilities 2_1 and 1_2 are open with a total profit of 100. In the case of the set function $v(R)$, note that when $R = \{1_1, 2_2\}$, $Q = \{1_1, 1_2, 2_2\}$, and $j = 2_1$, we have

$$w(R \cup \{j\}) - w(R) = 1 - 1 = 0 \leq 99 = 100 - 1 = w(Q \cup \{j\}) - w(Q).$$

That is, for $R \subset Q$ and $j \in V \setminus Q$, adding j to R does not increase w more than by adding j to Q and thus, the submodularity property is not satisfied by w.

Figure 3.1 illustrates the vertex representation in which the possible allocations of the customer to facilities are shown for R and Q. In this case, a simple path going from the customer to a second level facility corresponds to an allocation path. The thick lines represent the possible links between the customer and the first level facilities and between the first and second level facilities induced by the set R (thereby, also some of Q since $R \subseteq Q$). The thin lines represent the additional links induced by the set $Q \setminus R$. Figure 3.1(a) depicts the available allocation paths associated with the initial sets R and Q. Note that path $(1, 1_1, 1_2)$ induced by Q is the only one not contained in the set of paths induced by R. However, from Figure 3.1(b) we observe that when the element 2_1 is added to R and Q (i.e. 2_1 is open), two new allocation paths become available for $Q \cup \{2_1\}$ but only one of them is available for the set $R \cup \{2_1\}$. This means that the incremental could be $\rho_e(R) < \rho_e(Q)$, because of the additional possible selection that is offered to the set $Q \cup \{2_1\}$, which is precisely the case of this example.



(a) $Q = \{1_1, 1_2, 2_2\}$ and $R = \{1_1, 2_2\}$ (b) $Q = \{1_1, 1_2, 2_2, 2_1\}$ and $R = \{1_1, 2_2, 2_1\}$

Figure 3.1: Vertex representation of the MUFLP

The above observation motivates the use of the allocation paths as the combinatorial objects in the objective function $z(S, R)$, instead of vertices. This will ensure that when one path q is added to the set S, exactly one allocation path becomes available for both sets, S and T, with $S \subseteq T \subseteq P$, and therefore,

$$c(S \cup \{q\}) - c(S) \geq c(T \cup \{e\}) - c(T) \geq 0 \quad \text{for each } \ q \in P \backslash T.$$

For example, when $S = \{(1_1, 2_2)\}$, $T = \{(1_1, 1_2), (1_1, 2_2)\}$, $R = \{1_1, 2_2\}$, $Q = \{1_1, 1_2, 2_2\}$, and $q = (2_1, 1_2)$, we have

$$c(S \cup \{q\}, R) - c(S, R) = 100 - 1 = 99 \geq 99 = 100 - 1 = c(T \cup \{j\}, Q) - c(T, Q).$$

That is, for $S \subset T$ and $q \in P \setminus T$, adding q to S increased c as much as adding q to T.

Figure 3.2 illustrates the path-allocation representation, where the meaning of the thin and thick lines is the same as before but for the corresponding sets S and T in this case. In Figure 3.2(a) we have the configuration of possible patterns using the initial sets $S = \{(1_1, 2_2)\}$ and $T = \{(1_1, 2_2), (1_1, 1_2)\}$. Figures 3.2(b) and 3.2(c) represent the updated allocation patterns when adding the elements $(2_1, 1_2)$ and $(2_1, 2_2)$ to both S and T, respectively.

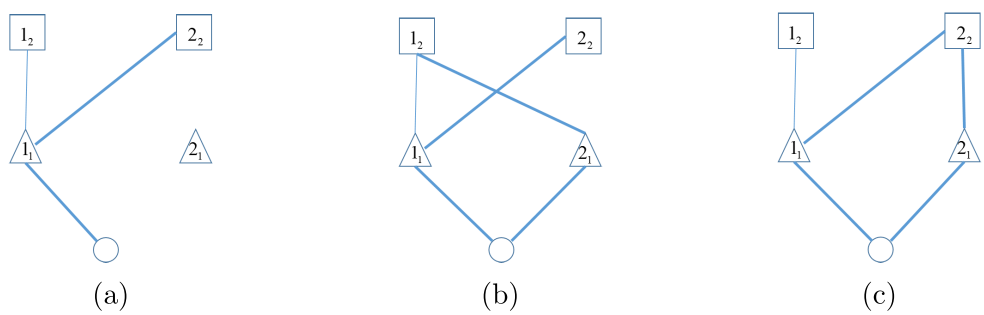


Figure 3.2: Path-allocation representation of the MUFLP

Note that in the vertex representation, when an element is added to the sets R and Q, the change in the objective function can vary abruptly, since opening a new facility implies that we can take any possible combination with the open facilities of another level for the allocation of customers. However, in the path-allocation representation, the change in the objective function when adding an element to S and T is controlled by the fact that only one new possibility of allocation is being offered to the customers, either through S or T.

Chapter 4

Formulations and approximation algorithms for multi-level uncapacitated facility location

The content of this chapter received the final acceptance for publication in February 2017 and is reprinted by permission. Originally entitled "Formulations and approximation algorithms for multi-level uncapacitated facility location", *INFORMS Journal on Computing*. Forthcoming. [136]

Abstract

This paper studies multi-level uncapacitated p-location problems, a general class of facility location problems. We use a combinatorial representation of the general problem where the objective function satisfies the submodular property, and we exploit this characterization to derive worst-case bounds for a greedy heuristic. We also obtain sharper bounds when the setup cost for opening facilities is zero and the

allocation profits are non-negative. Moreover, we introduce a mixed integer linear programming formulation for the problem based on the submodularity property. We present results of computational experiments to assess the performance of the greedy heuristic and that of the formulation. We compare the models with previously studied formulations.

4.1 Introduction

Hierarchical facility location problems (HFLPs) constitute an important class of *facility location problems* (FLPs) that consider different hierarchies of facilities and their interactions. Applications of HFLPs arise naturally in supply chain management [123] and logistics [148], where the interactions between warehouses, distribution centers and retail stores play a major role, and in health care systems [143] which typically require serving users from different levels of clinics and hospitals. Other examples arise in hierarchical telecommunication networks [34, 77]. The two recent surveys of [47] and [168] provide overviews of classification, models, applications, and algorithms for HFLPs.

Here we study a general class of HFLPs called *multi-level uncapacitated p-location problems* (MUpLPs), which can be defined as follows. Let $I = \{1, \cdots, m\}$ be the set of customers, $V_1, \cdots, V_k$ be the sets of sites among which facilities of levels 1 to k can be selected (or opened), with $V = \cup_{r=1}^{k} V_r$. Also, consider $c_{ij_1 \cdots j_k}$ to be the profit associated with the allocation of customer i to the sequence of facilities $j_1, \cdots, j_k$, where $j_r \in V_r$. Now, let $p = (p_1, \cdots, p_k)$ be a vector of positive integers, and let f_{j_r} be the non-negative fixed cost associated with opening facility j_r at level r. The MUpLP consists of selecting a set of facilities to open, such that no more than p_r facilities are opened at level r and of assigning each customer to a set of open facilities, exactly one at each level, while maximizing the total profit minus the setup cost of the open facilities.

The MUpLP subsumes the *uncapacitated p-location problem* (UpLP) [45] when $k = 1$, which in turn, contains as special cases both the *uncapacitated facility location problem* (UFLP) [100] and the *p-median problem* (p-MP) [83]. Thus, multi-level extensions of the UFLP and the p-MP are also special cases of the MUpLP. Namely, the well-known *multi-level uncapacitated facility location problem* (MUFLP) [90] is obtained when all cardinality constraints are redundant, i.e. when $p_r = |V_r|$ for all r, and to the best of our knowledge, a new generalization of the p-MP, called the *multi-level p-median problem* (MpMP), is obtained when all setup costs are set to zero, that is, $f_{j_r} = 0$.

The main contribution of this article is twofold. First, we state the MUpLP as the maximization of a set function satisfying the submodular property, subject to a set of linear constraints. This representation is used to obtain worst-case performance results of a greedy heuristic for the MUpLP. Sharper bounds are obtained for the case of the MpMP, in which the objective function is also non-decreasing. In particular, we obtain a $(1 - 1/e)$-approximation algorithm under some conditions on the profits c. This bound is known to be the optimal approximation bound for the single-level case if $P \neq NP$. Second, we introduce a mixed integer linear programming (MILP) formulation for the MUpLP also based on submodularity. A series of computational experiments are performed with a general purpose solver to compare the proposed formulation with respect to other MILP formulations previously introduced for special cases. Computational results on benchmark instances show the benefits and limitations of our formulation when embedded into a standard cutting plane algorithm for the general MUpLP and some special cases.

It is important to clarify that throughout this article we work with the maximization version of these problems. Similar to the case of the MUFLP, the maximization and minimization versions of the MUpLP are equivalent from an optimization point of view but not from the approximation algorithms perspective [see, 149, 170]. Thus, our results for the MUpLP can be adapted to the corresponding minimization version, except for those pertaining to the worst-case bounds of the greedy heuristics.

The remainder of the paper is organized as follows. Section 4.2 reviews the most relevant literature on the MUpLP and on submodularity. Section 4.3 provides a representation of the MUpLP as a combinatorial optimization problem and describes some fundamental properties of this representation. The worst-case bounds of the greedy heuristics are introduced in Section 4.4. In Section 4.5, we introduce a MILP formulation for the MUpLP based on submodularity, and in Section 4.6 we present computational experiments to compare the efficiency and limitations of the formulations. Conclusions follow in Section 4.7.

4.2 Literature Review

We use the classification scheme of HFLPs given by [47] in order to categorize the MUpLP. The classification is based on four criteria: *flow pattern*, *service availability* (or varieties), *spacial configuration* and *objective*. Other schemes may consider extra conditions such as capacity constraints or horizontal relationships between facilities of the same level. A *flow pattern* refers to the way in which a facility at a given level receives or offers services or products to another facility at a different level and is either single-flow (SF) or multi-flow (MF). In a network with SF pattern, the flow from or to the customers must pass through all higher levels until it reaches the point of origin or destination, while in a MF pattern, facilities of some level may receive or send flow directly from or to any higher level. *Service availability* specifies whether a higher-level facility provides all services provided by its lower-level facilities plus another one (nested), or whether facilities at each level provide different services (non-nested). In the *spacial configuration* category a network can be coherent or non-coherent. In a coherent network, facilities of lower-level must receive or send service from or to one and the same higher-level facility. Non-coherent systems allow more than one higher-level facility serving a given lower-level facility. Median, covering and fixed charge objectives are considered. Thus, for the MUpLP we identify an SF pattern and in principle a non-coherent structure. However, throughout the paper we make an assumption on the values of c which implies a coherent structure on the optimal solution. This is discussed in more detail in Section 4.3. The service availability criterion is application-dependent. Since we refer to different types of facilities instead of services that have an SF pattern, we can assume a non-nested configuration in this case. Moreover, what differentiates multi-level problems within HFLPs is that the initial set of potential facilities is partitioned in the input, and facilities of type r can only be opened in those potential sites of the set V_r. In a general setting of a HFLP, different hierarchical services are sometimes assigned to facilities that are not necessarily partitioned beforehand.

One of the most studied problems in this context is the MUFLP. Barros and Labbé [21] present MILP formulations and a branch-and-bound algorithm based on Lagragian relaxations for a more general two-level facility location including costs for opening edges. Also, [70] and [34] study the two-level facility location problem with single assignment constraints (coherent structure) including setup costs for the edges. Aardal et al. [3] show that all non-trivial facet defining inequalities for the UFLP also define facets for the two-level uncapacitated facility location problem. Aardal et al. [4], Bumb and Kern [28] and Zhang [169] use ideas previously developed for the UFLP, such as dual ascent and adjustment techniques, in order to develop approximation algorithms for the MUFLP. Ageev et al. [10] present approximation algorithms with worst-case bounds for the MUFLP. More recently, Krishnaswamy and Sviridenko [98] presented inapproximability results for the MUFLP and showed that in the general case, the two-level FLP is computationally harder than the UFLP. It is important to mention that in some of these references, the results are based on the equivalent minimization version of the MUFLP in which the values of c are interpreted as the costs of assigning customers to sequences of open facilities. However, recall that from the perspective of approximation algorithms, the two versions of the MUFLP are not equivalent [86, 149].

To the best of our knowledge, the definitions of MpMP and of the MUpLP just presented are new. However, some closely related problems, including those defined in the more general framework of HFLPs, have been studied. For instance, Teixeira and Antunes [154], Weaver and Church [162] and Hodgson [87] mainly discuss nested hierarchical p-median models. Serra and ReVelle [146, 147] and Alminyana et al. [13] discuss a nested and coherent hierarchical structure combining two p-median problems referred to as the pq-median problem. Edwards [55] studies a multi-level p-median problem in which the cardinality constraint is only required at the highest level of the facilities (V_k) and presents approximation results for the minimization version of the problem.

Cornuéjols et al. [45] presented important results for single level FLPs, namely worst-case bounds for greedy and local improvement heuristics for the maximization version of the UpLP, including the UFLP and the p-MP as special cases. These results were later generalized in the sequel of papers by Nemhauser et al. [134] and [63] for the maximization of a non-decreasing submodular set function subject to a cardinality constraint, and further to an independence system constraint. A result by Feige [59] implies that the worst-case approximation bound of $1 - 1/e$ given by the greedy heuristic for the combinatorial representation of the p-MP is the best possible approximation guarantee, unless $P = NP$. Moreover, it is the best possible guarantee for the maximization of a non-negative submodular function subject to a cardinality constraint.

Relevant results involving submodularity include those of Nemhauser and Wolsey

[132] who presented a MILP formulation, a cutting-plane and branch-and-bound algorithms to solve the maximization of a submodular set function subject to a cardinality constraint, using the p-MP as an example. Wolsey [165] applied this MILP formulation to the UFLP and discussed its connections with a Benders reformulation. More recently, Sviridenko [152] obtained the same worst-case bound of $1-1/e$ for the problem of maximizing a non-negative submodular set function subject to a knapsack constraint. Later, Calinescu et al. [30] presented a randomized approximation algorithm with worst-case bound for the problem of maximizing a monotone submodular function subject to an arbitrary matroid, and Kulik et al. [101, 102] introduced approximation algorithms for the maximization of a non-decreasing and non-negative submodular function subject to multiple linear and knapsack constraints. Contreras and Fernández [40] showed some of the benefits of representing a class of hub location problems as the minimization of a supermodular function subject to at most two cardinality constraints.

Some of the first articles discussing submodularity for the development of solution methods for the MUFLP are those of Ro and Tcha [145] and Tcha and Lee [153] who assumed that the submodularity property extends directly from the single-level cases. The correctness of such results was later discussed by Barros and Labbé [20] who concluded that the combinatorial representation of the MUFLP did not satisfy submodularity. However, other equivalent combinatorial optimization problems modeling the MUFLP have an objective function that actually satisfies submodularity, as was recently shown by Ortiz-Astorquiza et al. [135]. For a review on submodular optimization we refer the reader to [158] and [74].

4.3 Problem Definition and Submodular Properties

Let $G = (V \cup I, E \cup E_I)$ be a graph with a vertex set $V \cup I$ partitioned into $k+1$ levels where I represents the set of customers and $V_1, \cdots, V_k$ are the sets of potential facilities from levels 1 to k. The set E consists of edges between V_r and V_{r+1} for $r = 1, \cdots, k-1$, and E_I of those between I and V_1. We assume, without loss of generality, that all possible edges between V_r and V_{r+1}, as well as those between I and V_1, are in $E \cup E_I$. Now, let Q be the set of all possible simple paths having exactly one vertex from each level, starting from some vertex $j_1 \in V_1$, finishing at some vertex $j_k \in V_k$ and $N = Q \cup V$. Moreover, abusing the notation, we refer to each nonempty subset of N as the pair (S, R) thus, $(S, R) \subseteq N$, where $S \subseteq Q$ and $R \subseteq V$. Note that (S, R) is not a couple but a subset of N, which we denote as a pair in order to clearly differentiate the elements taken from Q and those taken from V. Now, we define

$$f(S,R) = -\sum_{r=1}^{k} \sum_{j_r \in R_r} f_{j_r}, \qquad h(S,R) = \sum_{i \in I} h^i(S,R) = \sum_{i \in I} \max_{(j_1,\cdots,j_k) \in S} c_{ij_1 \cdots j_k}$$

and

$$z(S,R) \quad = \quad h(S,R) + f(S,R) = \sum_{i \in I} \max_{(j_1,\cdots,j_k) \in S} c_{ij_1 \cdots j_k} - \sum_{r=1}^{k} \sum_{j \in R_r} f_{j_r},$$

where, $R = \cup_{r=1}^{k} R_r$, with $R_1 \subseteq V_1, \cdots, R_k \subseteq V_k$. The MU$p$LP can then be represented as the problem of selecting a (nonempty) set of paths $S \subseteq Q$ and a set of vertices $R \subseteq V$ satisfying the cardinality constraints such that $z(S,R)$ is maximum, that is,

$$\max_{(S,R)\subseteq N} \left\{ z(S,R) : N_r(S) = R_r, \quad \text{and} \quad |R_r| \leq p_r \quad \text{for } r = 1, \cdots, k \right\}, \tag{4.1}$$

where $N_r(S) = \{j \in V_r : j \text{ is a vertex in some path } q \in S\}$, is the set of vertices of level r associated with the paths of set $S \subseteq Q$. Observe that the first set of constraints of (4.1) state that for each vertex j_r on a path $q \in S$, the corresponding facility $j_r \in V_r$ must be open. The second set of constraints are the cardinality constraints on the number of open facilities at each level r. Also, note that adding a constant γ_i, for every $q \in Q$ to c_{iq} does not change the optimal solution. This result follows because in the MUpLP one must serve every client and thus, having new values of c, defined as $c'_{iq} = c_{iq} + \gamma_i$ changes every feasible solution in the same amount. This property is well known for the single level case [see, 46]. Moreover, we can model the profits as $c_{iq} = b_i - D_{iq}$, where b_i is the price that client i pays for the service and D_{iq} is the total operational cost of serving client i through path q. The price b_i can then be seen as the corresponding constant γ_i and therefore, only the costs are relevant for the decision, yielding the minimization version of the problem. From an optimization point of view, this is one of the reasons why it seems to be more common to work with the minimization version than with its maximization counterpart.

A fundamental property of z is that of submodularity. Before formally stating this result, we recall the definition of submodular and non-decreasing set functions [134]. Let X be a finite set and g be a real-valued function defined on the set of subsets of X, and let $\rho_e(W) = g(W \cup \{e\}) - g(W)$ be the incremental value of adding e to the set W when evaluating the set function g.

Definition 4.1.

- *g is submodular if $\rho_e(W) \geq \rho_e(U)$, $\quad \forall W \subseteq U \subseteq X$ and $e \in X \setminus U$.*

- *g is submodular and non-decreasing if $\rho_e(W) \geq \rho_e(U) \geq 0$, $\forall W \subseteq U \subseteq X$,*

$e \in X$.

The following result was proved by Ortiz-Astorquiza et al. [135].

Proposition 4.1.

- $h(S,R) = \sum_{i \in I} h^i(S,R)$ *is submodular and non-decreasing.*
- $z(S,R) = h(S,R) + f(S,R)$ *is submodular.*

The MUpLP can thus be stated as the maximization of a submodular set function subject to a set of constraints ensuring that: *i)* the selected paths in S are associated with a set of open facilities $R \subseteq V$, and *ii)* the number of open facilities at each level r does not exceed the predetermined value p_r. These constraints can be modeled by using a system of linear equations and thus, problem (4.1) is actually a particular case of the more general problem of maximizing a submodular function subject to a linear set of constraints [see, 132].

We next present some special cases of the MUpLP that are of particular interest.

- When we eliminate the cardinality constraints on the facilities at every level, i.e., $p_r = |V_r|$ for $r = 1, \ldots, k$, the MUpLP reduces to the MUFLP:

$$\max_{(S,R) \subseteq N} \left\{ z(S,R) : N_r(S) = R_r, \quad \text{for } r = 1, \cdots, k \right\}. \tag{4.2}$$

- When we eliminate the setup costs for the location of the facilities, i.e., $f_{j_r} = 0$ for each $j \in V_r$ and $r = 1, \ldots, k$, the MUpLP reduces to the MpMP:

$$\max_{S \subseteq Q} \left\{ h(S,R) : |N_r(S)| \leq p_r, \quad r = 1, \cdots, k \right\}. \tag{4.3}$$

Note that for the MpMP no subsets of vertices from V must be selected but only a subset of paths having an associated set of vertices on which the cardinality constraints are imposed. Thus, for this case instead of writing $(S, R) \subseteq N$ we will only write $S \subseteq Q$.

As in previous works, we assume that the profit (or cost) c is additive with respect to the profits on the edges. Typically, this assumption is made for the minimization version of the problem, where c corresponds to costs or more specifically distances [for instance, 4]. However, since we consider $c_{iq} = b_i - D_{iq}$, where D_{iq} is the total operational cost (e.g. distance) of assigning client i to path q, having additive costs D_{iq} relates directly with the additivity of the function c.

Assumption 4.1. *We assume that c is additive. Thus, for each $i \in I$ and $j_r \in V_r$ for $r = 1, \cdots, k$ we have $c_{ij_1\cdots j_k} = c_{ij_1} + c_{j_1j_2} + \cdots + c_{j_{k-1}j_k}$.*

The above assumption holds throughout the paper unless otherwise stated. We will also discuss some consequences on the results obtained when relaxing it. The following propositions are direct consequences of it and the proofs are found in the Online Supplement section. Also, we will present sharper bounds for a greedy heuristic when we assume that c is non-negative.

Proposition 4.2. *Under Assumption 4.1, there exists an optimal solution to the MUpLP in which every open facility at level r is assigned to exactly one facility at level $r+1$, for $r = 1, \ldots, k-1$ (i.e. coherent structure).*

Proposition 4.3. *Under Assumption 4.1, there exists an optimal solution to the MUpLP in which at most p_1 paths are used.*

Recall that the paths $q \in Q$ are sequences of vertices from V_1 to V_k and do not include vertices from the set of clients I.

4.4 Worst-Case Bounds for Greedy Heuristics

We now present worst-case bounds of greedy heuristics for the MUpLP, as well as some particular cases. Similar results are proved in [134] for the maximization of submodular functions subject to a single cardinality constraint. Recall that Assumption 4.1 holds throughout the paper unless otherwise stated.

4.4.1 A Greedy Heuristic for the MUpLP

We next describe a greedy heuristic for the MUpLP. Let $(S, R)^t$ denote the current solution at iteration t, with S^t and R^t its corresponding subsets of paths and vertices, respectively. Also, let $\rho_A(S, R) = z((S, R) \cup A) - z(S, R)$ be the incremental value of adding subset A to the set (S, R) when evaluating the set function z and denote ρ_t the maximum possible increment at iteration t. First, we note that a heuristic that takes one element of N at iteration t as candidate to $(S, R)^{t+1}$ does not necessarily terminate with a feasible solution since in N there exist both paths and vertices. We therefore consider a heuristic that constructs a feasible solution by adding at each iteration a subset of elements of N satisfying the feasibility conditions, i.e. $N_r(S) = R_r$, and $|R_r| \leq p_r$ for $r = 1, \cdots, k$, while increasing z the most. This is done by considering as candidate subsets those containing exactly one path $q \in Q$ with its corresponding vertices $N(\{q\}) = \cup_{r=1}^{k} N_r(\{q\})$, that are not yet in the solution. We define such subsets as $A_q(R^{t-1})$, where in general $A_q(R) = \{N(\{q\}) \setminus R\} \cup \{q\}$ for $q \in Q$ and $R \subseteq V$. In what follows, in order to simplify the notation, we will write $N_r(q)$ instead of $N_r(\{q\})$, and A_q instead of $A_q(R)$ when the selection of R is obvious. Moreover, we define $z(\emptyset)$ as the worst possible value of z, i.e., $z(\emptyset) = \sum_{i \in I} \min_{q \in Q} c_{iq} - p_1 \left(\max_{q \in Q} \sum_{r : j_r \in N_r(q)} f_{j_r} \right)$, ensuring that at the first iteration there is a positive change ρ_0. The procedure is outlined in Algorithm 1.

Before proving the main results for the worst-case bound obtained for this greedy heuristic we compute its running time.

Proposition 4.4. *The greedy heuristic for the MUpLP can be executed in* $O\left(p_1 |V_1| \left(|V| \log |V| + |E| + |I|\right)\right)$ *time.*

Algorithm 1 Greedy Heuristic for the MUpLP.

Let $(S,R)^0 \leftarrow \emptyset$, $N^0 \leftarrow N$ and $t \leftarrow 1$
while $t < p_1 + 1$ **do**
  Select $A_{q^*}(R^{t-1}) \subseteq N^{t-1}$ for which $\rho_{A_{q^*}}((S,R)^{t-1}) = \max_{A_q \subseteq N^{t-1}} \rho_{A_q}((S,R)^{t-1})$ with ties broken arbitrarily. Set $\rho_{t-1} \leftarrow \rho_{A_{q^*}}((S,R)^{t-1})$
  if $\rho_{t-1} \leq 0$ **then**
    Stop with $(S,R)^{t-1}$ as the greedy solution
  else
    Set $(S,R)^t \leftarrow (S,R)^{t-1} \cup A_{q^*}(R^{t-1})$ and $N^t \leftarrow N^{t-1} \setminus A_{q^*}(R^{t-1})$
  end if
  for r such that $|N_r(S^t)| = p_r$ **do**
    Set $N^t \leftarrow N^t \setminus \{q = (j_1, \cdots, j_k) : \exists j_r \in V_r \setminus R^t_r \text{ and } j_r \in q\}$
  end for
  $t \leftarrow t + 1$
end while
Stop with $(S,R)^{t-1}$ as the greedy solution

Proof. At iteration t the subset $A_{q^*}(R^{t-1}) \subseteq N^{t-1}$ can be efficiently identified by solving a series of shortest path problems as follows. We consider the auxiliary directed acyclic graph $G^t = (V^t, ARC^t)$, where $V^t = \cup_{r=1}^k V_r^t$,

$$ARC^t = \left\{(j_r, j_{r+1}) : j_r \in V_r^t, j_{r+1} \in V_{r+1}^t, r = 1, \ldots, k-1\right\},$$

and V_r^t is the set of vertices of level r that are either in the current solution or that are available to enter the solution at iteration t, that is, those vertices of level r for which the cardinality constraint is not binding. For each $a \in ARC^t$, we define its length as $w_{j_r j_{r+1}} = f_{j_{r+1}} - c_{j_r j_{r+1}}$ if $j_{r+1} \notin R^{t-1}$ and $w_{j_r j_{r+1}} = -c_{j_r j_{r+1}}$ if $j_{r+1} \in R^{t-1}$. This operation takes $O(|E|)$ time. We then compute a candidate path q, and its corresponding subset $A_q(R^{t-1})$, associated with each facility $j \in V_1 \setminus R_1^{t-1}$ by solving a shortest path problem between j and all nodes in V_k. This can be done in $O(|V|\log|V| + |E|)$ time using the Fibonacci heap implementation of Dijkstra's algorithm [11]. Finally, we evaluate $\rho_{A_q(R^{t-1})}((S,R)^{t-1})$ for each candidate path q. This takes $O(|I|)$ time for each of the at most $|V_1|$ paths. Therefore, each iteration of the algorithm takes a total of $O(|V_1|\,(|V|\log|V| + |E| + |I|))$ time. Given that there are at most p_1 iterations in the algorithm, the result follows. □

Now, we have the following result which follows directly from Proposition 4.1.

Proposition 4.5. *For* $(S, R) \subseteq (S', R') \subseteq N$ *and any subset* $A \subseteq N\backslash(S', R')$, $\rho_A(S, R) \geq \rho_A(S', R')$.

Moreover, since the set N is finite and given the definition of the function z, there exists a $\theta \geq 0$ for which $\rho_A(S, R) \geq -\theta$ for $(S, R) \subseteq N$ and $A \subseteq N \setminus (S, R)$. In the case of having a non-decreasing set function (e.g. h) $\theta = 0$.

Proposition 4.6. *For all* $(S, R), (T, W) \subseteq N$ *such that* $\cup_{r=1}^{k} N_r(S) = R$ *and* $\cup_{r=1}^{k} N_r(T) = W$,

$$z(T, W) \leq z(S, R) + \sum_{q \in T\backslash S} \rho_{A_q}(S, R) + |S\backslash T|\theta.$$

Proof. Let $(S, R), (T, W) \subseteq N$, with $|S\backslash T| = \beta$, $|T\backslash S| = \alpha$, such that $\cup_{r=1}^{k} N_r(S) = R$ and $\cup_{r=1}^{k} N_r(T) = W$. Consider the sets $A_q(R)$ with $q \in T\backslash S$ and similarly $B_s(W)$ with $s \in S\backslash T$, as defined before. Recall that for simplicity we only write A_q for $A_q(R)$ when it is possible. Also, with an abuse of notation, we enumerate the paths $q \in T \backslash S$ as $q = 1, \cdots, \alpha$ and similarly those in $S \setminus T$ as $s = 1, \cdots, \beta$. Then

$$z((S, R) \cup (T, W)) - z(S, R) \leq \sum_{q \in T\backslash S} \rho_{A_q}(S, R), \tag{4.4}$$

$$\text{since} \qquad z((S,R)\cup(T,W)) - z(S,R) =$$

$$z((S,R)\cup A_1) - z(S,R) + z((S,R)\cup A_1\cup A_2) - z((S,R)\cup A_1)$$

$$+\cdots + z((S,R)\cup A_1\cup\cdots\cup A_\alpha) - z((S,R)\cup A_1\cup\cdots\cup A_{\alpha-1})$$

$$= \sum_{i=1}^{\alpha} \rho_{A_i}((S,R)\cup A_1\cup\cdots\cup A_{i-1}) \leq \sum_{i=1}^{\alpha}\rho_{A_i}(S,R) = \sum_{q\in T\setminus S}\rho_{A_q}(S,R)$$

where the inequality follows from Proposition 4.5. Similarly, we obtain

$$z((S,R)\cup(T,W)) - z(T,W) \geq \sum_{s\in S\setminus T}\rho_{B_s}((T,W)\cup(S,R)\setminus B_s). \tag{4.5}$$

Subtracting (4.5) from (4.4), we obtain

$$z(T,W) \leq z(S,R) + \sum_{q\in T\setminus S}\rho_{A_q}(S,R) - \sum_{s\in S\setminus T}\rho_{B_s}((T,W)\cup(S,R)\setminus B_s).$$

Since $\rho \geq -\theta$, it follows that $z(T,W) \leq z(S,R) + \sum_{q\in T\setminus S}\rho_{A_q}(S,R) + \beta\theta$. □

Let Z be the optimal solution value of an instance of the MUpLP and let Z^G be the value of a solution obtained using Algorithm 1. Thus, $Z^G = z(\emptyset) + \rho_0 + \rho_1 + \cdots + \rho_{t^*-1}$, with $t^* \leq p_1$.

Proposition 4.7. *If the greedy heuristic for the MUpLP stops after t^* iterations then*

$$Z \leq z(\emptyset) + \sum_{i=0}^{t-1}\rho_i + p_1\rho_t + t\theta \qquad t = 0,\cdots,t^*-1, \tag{4.6}$$

and also

$$Z \leq z(\emptyset) + \sum_{i=0}^{t^*-1}\rho_i + t^*\theta \qquad \textit{if } t^* < p_1.$$

Proof. By Proposition 4.6 we have $z(T,W) \leq z(S,R) + \sum_{q\in T\setminus S}\rho_{A_q(R)}(S,R) + |S\setminus T|\theta$. Now consider $(T,W) \subseteq N$ to be the optimal solution (i.e. $Z = z(T,W)$) and $(S,R) =$

$(S,R)^t$. Then, since for $q \in T \setminus S^t$ and iteration t, $\rho_{A_q(R^t)}(S,R)^t \leq \rho_t$, $\theta \geq 0$, $|S^t \setminus T| \leq t$, $|T \setminus S^t| \leq p_1$, and $z((S,R)^t) = z(\emptyset) + \sum_{i=0}^{t-1} \rho_i$, we have

$$Z = z(T,W) \leq z(\emptyset) + \sum_{i=0}^{t-1} \rho_i + p_1 \rho_t + t\theta \quad \text{for } t = 0, \cdots, t^* - 1.$$

If $t^* < p_1$ and $(S,R) = (S,R)^{t^*}$ then,

$$Z \leq z(\emptyset) + \sum_{i=0}^{t^*-1} \rho_i + t^* \theta,$$

since $\rho_{t^*} \leq 0$. □

Thus, if the greedy heuristic is applied to MUpLP, using $t = 0$ in (4.6) and the fact that in this case $Z^G = z(\emptyset) + \rho_0$, we obtain

$$\frac{Z - Z^G}{Z - z(\emptyset)} \leq \frac{p_1 - 1}{p_1}.$$

A more general result for $t^* > 0$ can be obtained by using the results described above, as well as those of Lemma (4.1) and Theorem (4.1) part (a) from [134]. The proofs are omitted because they can be followed with simple modifications.

Proposition 4.8. *[see, Theorem 4.1 134] If the greedy heuristic for the MUpLP terminates after t^* iterations,*

$$\frac{Z - Z^G}{Z - z(\emptyset) + p_1 \theta} \leq \frac{t^*}{p_1} \left(\frac{p_1 - 1}{p_1} \right)^{p_1} \leq \left(\frac{p_1 - 1}{p_1} \right)^{p_1}.$$

4.4.2 A Greedy Heuristic for the MpMP

Sharper bounds can be obtained for the particular case of the MpMP in which the setup costs of the facilities are all equal to zero, i.e. $f(S,R) = 0$ and when the profits c are non-negative. We consider an adaptation of the previous greedy heuristic for the MpMP which consists of finding a greedy solution by adding at each iteration exactly one path $q \in Q$ that increases h the most. The procedure is outlined in the Online Supplement section.

Algorithm 2 has the same running time as Algorithm 1. Given that h is submodular and non-decreasing and $f(S, R) = 0$ in the MpMP, the results of Section 4.4.1 hold for $\theta = 0$. Let H be the value of an optimal solution of MpMP and H^G be the value of a particular solution obtained with Algorithm 2. Then, $H^G = h(\emptyset) + \rho_0 + \rho_1 + \cdots + \rho_{t^*-1}$, with $t^* \leq p_1$. Moreover, if we consider $c \geq 0$, we have $h(\emptyset) = 0$.

Proposition 4.9. *If the greedy heuristic for the MpMP stops after $t < p_1$ steps, then the greedy solution is optimal.*

The proof follows directly from the second part of Proposition 4.7 with $\theta = 0$. Similarly, in the trivial case in which $p_r = 1$ for all $r = 1, \cdots, k$, the greedy heuristic yields an optimal solution. More importantly,

Proposition 4.10. *If $c \geq 0$ and the greedy heuristic is applied to MpMP, then*

$$\frac{H - H^G}{H} \leq \left(\frac{p_1 - 1}{p_1}\right)^{p_1},$$

and the bound is tight.

It then follows that $H^G/H \geq 1 - \left(\frac{p_1-1}{p_1}\right)^{p_1} \geq 1 - 1/e$, which coincides with the best worst-case bound for the single level p-MP. Thus, the same instances presented in [45] that prove the tightness of the bound for one level can be used in this more general setting.

4.4.3 A Greedy Heuristic for the MpMP with General Costs

We conclude this section by providing a worst-case bound of a greedy heuristic for the MpMP when Assumption 4.1 is relaxed. We first note that MpMP is actually a particular case of the more general case studied in [63] of maximizing a submodular function over an independence system described as the intersection of a finite number of matroids. We recall the definitions of matroid and of independence system.

Definition 4.2. *A matroid $\mathscr{M}$ is a pair $(X, \mathscr{F})$ where X is a finite ground set of elements and $\mathscr{F}$ is a collection of subsets of elements of X, satisfying*

- $A \in \mathscr{F}$ *and* $B \subseteq A$ *then* $B \in \mathscr{F}$.
- $A, B \in \mathscr{F}$ *with* $|A| > |B|$ *then* $\exists e \in A \backslash B$ *such that* $B \cup \{e\} \in \mathscr{F}$.

Sets in $\mathscr{F}$ satisfying only the first condition are referred to as independence systems.

For the case of the MpMP represented in (4.3), we have seen that the objective function satisfies submodularity and it is a non-decreasing set function. In terms of the constraints, note that we have k cardinality constraints and it is straightforward to see that in general, the pair $(Q, \mathscr{F})$ with $\mathscr{F} = \{S \subseteq Q : |N_r(S)| \leq p_r \quad \text{for} \quad r = 1, \cdots, k\}$ does not form a matroid. However, the pair $(Q, \mathscr{F})$ satisfies the first part of Definition 4.2. Thus, the combinatorial problem (4.3) of MpMP is a particular case of the problem of maximizing a submodular non-decreasing set function subject to an independence system. Given that Korte and Hausman [95] [see, 37] have shown that every independence system can be written as the finite intersection of matroids, problem (4.3) can actually be seen as a particular case of the problem studied by Fisher et al. [63]. An interesting consequence of this is that one can directly obtain worst-case bounds of a greedy heuristic presented in [63] that do not depend on the number of matroids, but only on the cardinality of the smallest dependent set and on that of the largest independent set in the independence system.

Proposition 4.11. *If the greedy heuristic given in [63] is applied to MpMP, then*

$$\frac{H - H^G}{H - h(\emptyset)} \leq \left(\frac{B-1}{B}\right)^b,$$

where, B is the cardinality of the largest independent set in $(Q, \mathscr{F})$ and $b+1$ is the cardinality of the smallest dependent set.

If we relax Assumption 4.1, we have

$$\frac{H - H^G}{H - h(\emptyset)} \leq \left(\frac{p_1 \cdots p_k - 1}{p_1 \cdots p_k} \right)^{\min_r \{p_r\}}.$$

Note that if we consider Assumption 4.1, then $B = p_1$. Together with the modifications previously presented for the greedy heuristic for the MpMP, we get $b = p_1$. Thus, we obtain the same result as in Proposition 4.10.

4.5 Formulations for the MUpLP

We next introduce a MILP formulation for the MUpLP that exploits the properties of a submodular function. First we present the results required for the formulation of the MpMP and then we extend these results for the more general MUpLP [see, 132, 165]. Recall that a given instance of the MUpLP can be transformed by adding a constant γ_i to every path $q \in Q$, $c'_{iq} = c_{iq} + \gamma_i$, without affecting the optimal solution. Therefore, in the following section we assume, without loss of generality, that $c_{iq} \geq 0$ for every $i \in I$ and $q \in Q$.

4.5.1 A Submodular Formulation for the MpMP

Recall that in general $h(S, R) = \sum_{i \in I} h^i(S, R) = \sum_{i \in I} \max_{(j_1, \cdots, j_k) \in S} c_{ij_1 \cdots j_k}$, but since h does not depend on R, in this section we use the notation $h(S)$. Consider the polyhedron X defined as

$$\{(\eta, x, y_1, \cdots, y_k) : \eta \leq h(S) + \sum_{q \in Q \setminus S} \rho_q(S) x_q,\ S \subseteq Q,\ x \in \{0,1\}^{|Q|},\ y_r \in \{0,1\}^{|V_r|},\ \eta \in \mathbb{R}\},$$

where the binary variables x_q can be interpreted as $x_q = 1$ if the path $q \in Q$ is open and 0 otherwise, and y_r corresponds to the characteristic vector for each level r of the facilities that are open.

Proposition 4.12. *Let $T \subseteq Q$, $N_r(T) \subseteq V_r$ for all r, and $(\eta, x^T, y_1^T, \cdots, y_k^T)$ where $x^T, y_1^T, \cdots, y_k^T$ are the incidence vectors of T and $N_r(T)$, respectively. Then, $(\eta, x^T, y_1^T, \cdots, y_k^T) \in X$ if and only if $\eta \leq h(T)$.*

The proof is given the Online Supplement section

Consider the following MILP formulation of MpMP:

$$\begin{aligned}
\text{(SF) maximize} \quad & \eta && \\
\text{subject to} \quad & \eta \leq h(S) + \sum_{q \in Q \setminus S} \rho_q(S) x_q && \forall \;\; S \subseteq Q && (4.7)\\
& \sum_{q \in Q : j_r \in q} x_q \leq M_r y_{j_r} && \forall \;\; j_r \in V_r, \quad r = 1, \cdots, k && (4.8)\\
& \sum_{j_r \in V_r} y_{j_r} \leq p_r && r = 1, \cdots, k && (4.9)\\
& x_q \in \{0, 1\} && \forall \;\; q \in Q && (4.10)\\
& y_{j_r} \in \{0, 1\} && \forall \;\; j_r \in V_r, \quad r = 1, \cdots, k. && (4.11)
\end{aligned}$$

Inequalities (4.7) are called the submodular constraints and compute the profit of every $S \subseteq Q$. Constraints (4.8) are the linking constraints between x and y, and inequalities (4.9) ensure the cardinality restrictions for each level. We have chosen to present the aggregated version of inequalities (4.8) since we can exploit the structure of the problem and take a "good" value for M_r, knowing that the optimal solution will not have more than p_1 paths (Proposition 4.3). Thus, we select $M_r = \min\{p_1, |Q|/|V_r|\}$ in order to have a tighter formulation. Further comments on the selection of the aggregated constraints are given in the computational experiments. Additionally, note that as in the single level p-MP, we can drop the integrality constraints on the x variables.

Proposition 4.13. $(\eta^*, x^*, y^*) = (h(T^*), x^{T^*}, y_1^{T^*}, \cdots, y_k^{T^*})$ *is an optimal solution to SF if and only if* T^* *is an optimal solution to Problem* (4.3).

The proof is given in the Online Supplement section.

Also, note that since $h(S)$ is the sum of $|I|$ submodular set functions, one for each $i \in I$, we can obtain a tighter formulation by replacing the objective function η by $\sum_{i \in I} \eta^i$ and constraints (4.7) with

$$\eta^i \leq h^i(S) + \sum_{q \in Q \setminus S} \rho_q^i(S) x_q \qquad \forall \;\; i \in I, \;\; S \subseteq Q, \tag{4.12}$$

where $\rho_q^i(S) = h^i(S \cup \{q\}) - h^i(S)$. Moreover, most of these inequalities are redundant. First, note that for $S \subseteq Q$ and $i \in I$ given, the right-hand side of their associated constraint (4.12) does not change if the summation is taken over all $q \in Q$. Also, $h^i(S) = c_{is_1 \cdots s_k}$ for some $s_1, \cdots, s_k \in S$. For simplicity, we write c_{is} for $s \in S \subseteq Q$. Then, $\rho_q^i(S) = c_{iq} - c_{is}$ if $c_{iq} > c_{is}$ or $\rho_q^i(S) = 0$ if $c_{iq} \leq c_{is}$. For any S, its associated constraint (4.12) can thus be written as $\eta^i \leq c_{is} + \sum_{q \in Q} (c_{iq} - c_{is})^+ x_q$, for some $s \in S$ and $\chi^+ = \max\{0, \chi\}$. Therefore, if for each $i \in I$ we consider the ordering $0 = c_{iq_0} \leq c_{iq_1} \leq \cdots \leq c_{iq_{|Q|}}$, we may select only the sets $S_q = \{q\}$ with $q = q_0, \cdots, q_{|Q|-1}$ in constraints (4.12). We prove this result in the following proposition.

Proposition 4.14. *The MpMP can be formulated as*

$$\text{(SFD) maximize} \sum_{i \in I} \eta^i$$

$$\text{subject to } (4.8) - (4.11)$$

$$\eta^i \leq c_{iq_t} + \sum_{q \in Q} (c_{iq} - c_{iq_t})^+ x_q \;\; \forall \;\; i \in I, \quad t = 0, \cdots, |Q| - 1. \tag{4.13}$$

Proof. Since constraints (4.13) are a subset of constraints (4.7), we only need to show that if (ζ, x^T, y^T) does not satisfy constraints (4.7) (i.e $\zeta > h^{\hat{i}}(T)$ for some $\hat{i}$, by Proposition 4.12) for a given $T \subseteq Q$, then (ζ, x^T, y^T) is also infeasible with respect to constraints (4.13). Thus, suppose $h^{\hat{i}}(T) = \max_{q \in T} c_{iq} = c_{\hat{i}q_t}$, then the associated t^{th} inequality (4.13) would be

$$\zeta \leq c_{\hat{i}q_{t-1}} + \sum_{q \in Q} (c_{\hat{i}q} - c_{\hat{i}q_{t-1}})^+ x_q^T = c_{\hat{i}q_{t-1}} + c_{\hat{i}q_t} - c_{\hat{i}q_{t-1}} = c_{\hat{i}q_t} = h^{\hat{i}}(T),$$

which contradicts $\zeta > h^{\hat{i}}(T)$ and the result follows. □

Finally, we consider the additional constraint

$$\sum_{q \in Q} x_q \leq p_1, \tag{4.14}$$

which explicitly incorporates Proposition 4.3 into the formulation. Even though this constraint is redundant for SFD, preliminary computational experiments have shown that it can help reduce the CPU time of a branch-and-cut algorithm. Note that the LP relaxation is strengthened when this valid inequality is added. Thus, if we consider the polyhedral sets associated with the LP relaxation of the constraints of SFD and that obtained after the addition of equation (4.14), clearly the latter is contained in the former. Moreover, consider the following fractional feasible solution values for SFD, when $M_r = p_1$ for all r and $\min_r p_r \geq 2$. Let $x_q = (p_1 + \epsilon)/|Q|$ for every $q \in Q$ and $y_{j_r} = p_r/|V_r|$ for all $j_r \in V_r$, with $\epsilon \in (0, 1)$, then it is not difficult to show that such solution does not satisfy equation (4.14) and therefore the corresponding polyhedral set is strictly contained in the initial one.

4.5.2 A Submodular Formulation for the MUpLP

Since $z(S,R) = h(S,R) + f(S,R)$ where h is submodular and non-decreasing and f is a linear function, we can reformulate the MUpLP as follows [see, 165]:

$$\begin{aligned} \text{(SFML) maximize} \quad & \sum_{i \in I} \eta^i - \sum_{r=1}^{k} \sum_{j_r \in V_r} f_{j_r} y_{j_r} \\ \text{subject to} \quad & (4.8) - (4.11), (4.13), (4.14). \end{aligned}$$

By removing constraints (4.9) from SFML we obtain a MILP for the MUFLP.

4.5.3 A Branch-and-Cut Algorithm

Typically, the instances of the well-known MUFLP consider relatively small values of k (i.e. $k = 2, 3, 4$). Therefore, if we consider k a fixed value, then the total number of paths is bounded by a polynomial in k, that is, $|Q| = |V_1| \times \cdots \times |V_k| \leq (\max_r |V_r|)^k$. However, one of the drawbacks of the SFD and SFML models is that even though they contain a polynomial number of variables and constraints in the input of the problem when k is fixed, the submodular constraints (4.13) are actually very dense. Therefore, handling such constraints efficiently in a general purpose solver to solve these models is desirable. Our aim is not to propose a specialized exact solution algorithm for the MUpLP (or MpMP) but rather to treat constraints (4.13) in an efficient way. The idea is to use a standard branch-and-cut algorithm in which only a few constraints are initially considered. In our case, we have selected the least dense constraints which correspond to $t = |Q| - 1$ for every $i \in I$. Then, we add additional constraints (4.13) only when they are violated at fractional and integer solutions of the enumeration tree.

Given a solution $(\bar{\eta}, \bar{x}, \bar{y})$ of the LP relaxation of formulation (4.8)–(4.11), and (4.14), a separation problem must be solved for inequalities (4.13). A naive solution of the separation problem would be to inspect each of the remaining constraints. For each $i \in I$, this sequential inspection can be carried out in $O(|Q|^2)$ time, but since the right-hand side of (4.13) is a piecewise concave function, the separation problem can be solved more efficiently.

Proposition 4.15. *For each $i \in I$, the separation problem of inequalities* (4.13) *can be solved in $O(|Q|)$ time.*

The proof is given in the Online Supplement section.

4.6 Computational Experiments

We have conducted a computational study in order to assess the empirical performance of the greedy heuristics of Section 4.4 and of the SFD and SFML formulations of Section 4.5 with a general purpose solver. All algorithms were coded in C and run on an Intel Xeon E3 1240 V2 processor at 3.40 GHz and 24GB of RAM under a Windows 7 environment. The formulations were implemented using the callable library of CPLEX 12.6.2.

For our computational experiments, we have transformed benchmark instances of the closely related UFLP to the multi-level case. In particular, we have used instances with 1,000 customers and 100 potential facilities such as the *capa*, *capb* and *capc* instances from the OR-Library [23]. The modification to multi-level instances was carried out as follows. We have generated instances in which $|V_1| \geq \cdots \geq |V_k|$ to represent the level hierarchy of facilities, and we have selected the first $|V_r|$ facilities of the UFLP instance for the r^{th} level facilities, with $k = 2$ and 3. The setup costs for opening facilities were modified in order to be dependent on its level. Thus, a value of f_j is multiplied by r, so that higher level facilities are more expensive than lower level facilities. We have also used the profits of customer allocations. However, the OR-Library instances do not provide distances (profits) between facilities but only between customers and facilities. Therefore, we have constructed inter-facility distances by taking the minimum two-hop distance between facilities via a customer [as in 55]. We then transformed these distances into profits by simply subtracting them from a sufficiently large constant. For every multi-level *cap* instance, three values of $p = (p_1, \cdots, p_k)$ were selected. One relatively small or tight, one of medium value and one with the same values of the potential facilities configuration, that is, making the cardinality constraints redundant (as in the MUFLP).

We also considered another set of randomly generated instances with larger numbers of customers and potential facilities. We first generated the coordinates of potential facilities and customers in the plane. We then computed the setup costs f_{j_r} using $RAND(50, 500) \times r$, where $RAND(a, b)$ outputs a random real value between a and b and r is the level of the facility. The values of the profits c were obtained after computing the Euclidean distances and subtracting them from a sufficiently large number. The $c_{j_r j_{r+1}}$ values range from 25 to 125. These instances contain between 500 and 2,000 customers, between 100 and 200 potential facilities, and between two and four levels. For these instances we considered four different values of p. Finally, we also present three instances for which we illustrate that the bound obtained for the greedy heuristic for the MpMP is tight. These instances are called *tightC2*, *tightC3* and *tightC4*, and were constructed using the profit matrices of [45]. All instances used in these experiments are available at *https://users.encs.concordia.ca/ icontrer/web/instances.html*.

For comparison purposes, we have adapted three known MILP formulations for the MUFLP to our more general MUpLP. These are a path-based formulation (PBF) [4, 55], an arc-based formulation (ARB) [3, 66] and a flow-based formulation (FBF) [97]. These formulations are provided in the Online Supplement.

All MILP formulations were executed using a standard branch-and-cut search with single thread to make the comparisons as fair as possible. We also turned off the CPLEX heuristics, since this seems to yield a better performance irrespective of the formulation. We provide CPLEX with the lower bound value obtained using the greedy heuristic. The remaining parameters were set at their default values. It is also important to mention that we present the results obtained for the SFD and the SFML with the aggregated version of the linking constraints (4.8). While the disaggregated version of the linking constraints, i.e. $x_q \leq y_{j_r}, j_r \in V_r, r = 1, \cdots, k, q \in Q, j_r \in q$, yields a better LP gap and sometimes outperforms the aggregated version, the latter configuration solved more instances within the time limit. We also note that, for the PBF the aggregated version performs better than the disaggregated one, but consumes a considerable amount of memory. However, the aggregated version solved more instances within the time limit and with the available memory than its disaggregated counterpart.

Table 4.1 summarizes the comparison of SFD, FBF, ARB, PBF, and Algorithm 2 for the MpMP. A more detailed comparison is presented in Section 4 of the Online Supplement. Table 4.1 describes the proportion of instances solved to optimality by the corresponding formulation within a time limit of $1,000$ seconds. Each column represents different types of instances solved. For example, in the column $k = 2$ we provide the number of two level instances solved to optimality, whereas in the column *cap* we provide the same information for the case of the multi-level *cap* instances. Also, in the last three columns we have the average %gap and the shifted geometric mean (SGM) for the CPU time and number of nodes explored in the enumeration tree. For the formulations, we used %gap $= 100(LP_X - OPT)/(LP_X)$, where OPT is the optimal value and LP_X is the LP bound obtained with formulation X. For the heuristic, we used %gap $= 100(BEST - OPT)/(OPT)$, where $BEST$ is the solution value obtained with the greedy algorithm. We computed the two SGM values by only considering those instances solved by the three formulations that solved the most instances within the time limit, (i.e. SFD, ARB and FBF) since the PBF failed in most instances. For the computation of these values we used $SGM = \prod_{i=1}^{L}(t_i + s)^{1/L} - s$, with $s = 10$ and L equal to the number of instances considered.

Table 4.1: Summary of the computational results for the MpMP.

	$k=2$	$k=3$	$k=4$	$\|I\|=500$	$\|I\|$=1,000	$\|I\|$=1,500	$\|I\|$=2,000	cap	Total	SGM sec	SGM nodes	Avg. %gap
SFD	37/39	23/25	12/12	20/20	33/33	13/16	3/4	21/21	72/76	2.55	11.74	1.24
FBF	28/39	18/25	11/12	17/20	29/33	6/16	2/4	21/21	57/76	8.35	12.11	3.19
ARB	28/39	14/25	6/12	17/20	23/33	4/16	1/4	19/21	48/76	47.37	0.15	0.00
PBF	33/39	5/25	0/12	9/20	20/33	5/16	1/4	15/21	38/76	-	-	-
Greedy	17/39	12/25	7/12	8/20	23/33	4/16	1/4	15/21	36/76	0.01	-	1.33

From Table 4.1 and Table 1 of the Online Supplement we can see that SFD outperforms the other three formulations on almost all test instances solved for the MpMP, and has the lowest SGM time for the solved instances. It is the formulation that solved the most instances to optimality (72 out of 76) within the time limit and for each instance type taken separately. Five instances could only be solved by the SFD and every instance solved by the other formulations was also solved by the SFD. Although the %LP gap for the implementation of the SFD is not the tightest, it seems that SFD exhibits the best trade-off between %LP gap, the number of explored nodes and memory consumption. As we can see, the PFB and the ARB have a better %LP gap than the SFD but are inefficient due to high memory consumption or CPU time. They are in most cases slower than FBF and SFD. On the other hand, we noted that the FBF has a low memory usage but has the worst LP gap on all instances and therefore explores more nodes. However, it is the second formulation that solved the most instances. Regarding the %gap of the greedy solution for the MpMP, we see that the tightness of the worst-case bound is attained on the first three instances, while the remaining solutions have a deviation never exceeding 2%.

Figure 4.1 shows the total number of solved instances of the MpMP with respect to a continuous variation on the time, going from 0 to 1,000 seconds. After this time limit very few instances are solved with any of the formulations. We observe that all formulations reach a peak after a few seconds and then the concave curves start flattening. However, the SFD clearly outperforms the other three models, solving more than 60 instances within less than 50 seconds.

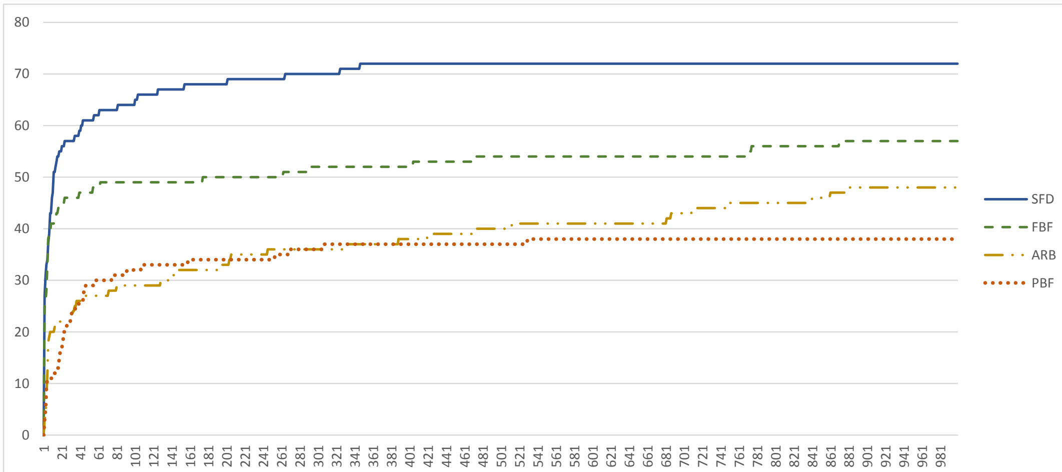


Figure 4.1: Comparison of the formulations in terms of the MpMP instances solved

Table 4.2 summarizes the results for the MUpLP. We have added one extra column corresponding to the instances of the MUFLP. We have also removed from this part of the experiments the first three instances which were only available for the MpMP case and are used to show the tight bound obtained with the greedy heuristic. In Table 4.4 of the Online Supplement section we provided the detailed information for all instances.

Table 4.2: Summary of the Computational Results for the MUpLP.

	$k=2$	$k=3$	$k=4$	$\|I\|=500$	$\|I\|$=1,000	$\|I\|$=1,500	$\|I\|$=2000	cap	MUFLP	Total	SGM sec	SGM nodes	Avg. %gap
SFML	27/36	15/25	9/12	15/20	27/33	8/16	1/4	18/21	10/20	51/73	54.44	425.79	4.44
FBF	17/36	13/25	12/12	15/20	25/33	2/16	0/4	21/21	13/20	42/73	66.03	67.41	7.63
ARB	25/36	16/25	6/12	17/20	25/33	4/16	1/4	21/21	20/20	47/73	86.01	0.36	0.00
PBF	30/36	4/25	0/12	9/20	19/33	5/16	1/4	15/21	14/20	34/73	-	-	-
Greedy	1/36	0/25	0/12	1/20	0/33	0/16	0/4	0/21	0/20	1/73	0.01	-	5.98

The results of Table 4.2 show that, even though the SFML solved more instances in total, none of the formulations clearly dominates the others. For instance, on the *cap* instances the FBF and ARB formulations performed better than the other two, and for MUFLP instances, the ARB was the best one. The FBF model is also more efficient in the memory consumption, which is particularly useful for instances

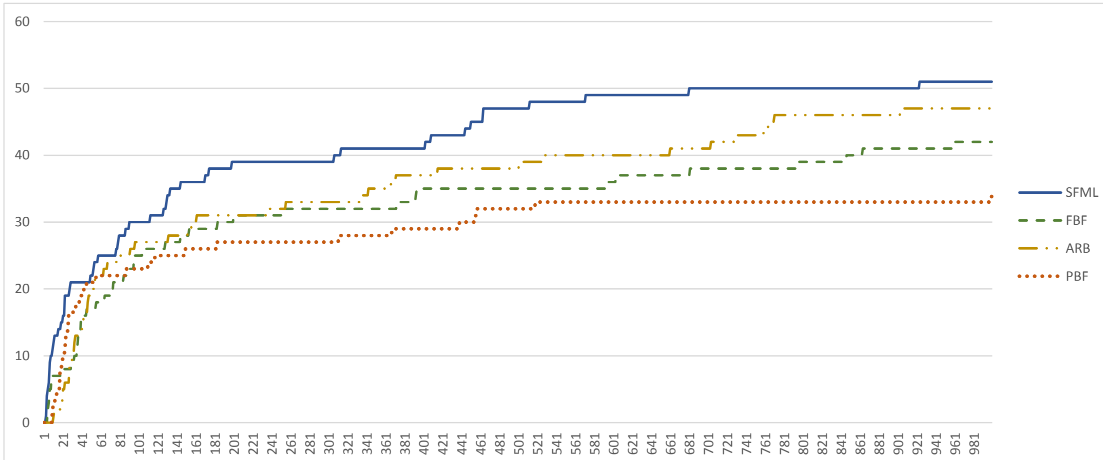


Figure 4.2: Comparison of the formulations in terms of the MUpLP instances solved

with more levels. However, when we increased the number of customers or potential facilities the performance of FBF deteriorates drastically, even when $k = 2$, where the other formulations are faster. On the other hand, the SFML is more competitive when the values of p_r are not redundant.

Figure 4.2 shows the total number of solved instances of the MUpLP. We note that for this more general problem, the gap between the curves is not as important as it was for the MpMP. Moreover, for the first 60 seconds all models solve almost the same number of instances, but the SFML maintains its dominance after 1,000 seconds. As was expected for the greedy solutions, the deviation from the optimal value is much larger in comparison with the MpMP. However, these solutions may be used as starting points in more elaborated heuristic procedures.

4.7 Conclusions

We have studied a general class of hierarchical facility location problems, called multi-level uncapacitated p-location problems. These problems were modeled as the maximization of a submodular set function, subject to a set of linear constraints. This representation was used to obtain worst-case performance results of some greedy heuristics for the general case called MUpLP and for the particular cases of the MpMP in which sharper bounds were obtained. In particular, we obtained a $(1 - 1/e)$-approximation algorithm for the case in which profits are non-negative and additive. We have also introduced an integer linear programming formulation for the MUpLP based on the submodular property. The results of our computational experiments confirm the efficiency of the submodular formulation over previous formulations modified for the MpMP. Instances with up to 2,000 customers, 200 potential facilities, and four levels of hierarchy were solved to optimality. Our results also show that for the more general case of the MUpLP, none of the considered MILP formulations clearly dominates the others.

Online Supplement

Greedy Heuristic

Since we only consider paths instead of the subsets $A_q(R)$, we refer to elements in Q instead of N.

Algorithm 2. Greedy Heuristic for the MpMP.

Let $S^0 = \emptyset$, $Q^0 = Q$ and set $t = 1$.
while $t < p_1 + 1$ **do**
  Select $q(t) \in Q^{t-1}$ for which $\rho_{q(t)}(S^{t-1}) = \max_{q \in Q^{t-1}} \rho_q(S^{t-1})$ with ties settled arbitrarily. Set $\rho_{t-1} = \rho_{q(t)}(S^{t-1})$.
  if $\rho_{t-1} \leq 0$ **then**
    Stop with S^{t-1} as the greedy solution
  else
    $S^t \leftarrow S^{t-1} \cup q(t)$ and $Q^t \leftarrow Q^{t-1} \setminus q(t)$
  end if
  for r such that $|N_r(S^t)| = p_r$ **do**
    Set $Q^t \leftarrow Q^t \setminus \{q = (j_1, \cdots, j_k) : \exists j_r \in V_r \setminus R_r^t \text{ and } j_r \in q\}$
  end for
  $t \leftarrow t + 1$
end while

Proofs of the Propositions

Proof of Proposition 4.2

Proof. Suppose that every optimal solution does not satisfy the single assignment property between lower to upper level facilities. Then, let (S^*, R^*) be an optimal solution in which there exist (at least) two clients i_1 and i_2 that are assigned to the sequences of opened facilities $q^1 = j_1^1, \cdots, j_s, j_{s+1}^1, \cdots, j_k^1$ and $q^2 = j_1^2, \cdots, j_s, j_{s+1}^2, \cdots, j_k^2$ respectively, where j_s is a facility of level s which is connected to two different facilities j_{s+1}^1 and j_{s+1}^2 of level $s+1$, i.e. $j_{s+1}^1 \neq j_{s+1}^2$.

Without loss of generality, assume that the profits associated with q^1 and q^2 from level s to level k are such $c_{j_s j_{s+1}^1} + \cdots + c_{j_{k-1}^1 j_k^1} \geq c_{j_s j_{s+1}^2} + \cdots + c_{j_{k-1}^2 j_k^2}$. Then, we can construct a new feasible solution $(\hat{S}, \hat{R})$ where client i_2 is reassigned to path $\hat{q}^2 = j_1^2, \cdots, j_s, j_{s+1}^1, \cdots, j_k^1$, such that

$$z(S^*, R^*) \leq z(\hat{S}, \hat{R}),$$

which contradicts the assumption and the result follows. □

Proof of Proposition 4.3

Proof. Suppose that there exists an optimal solution to the MUpLP containing more than p_1 paths, i.e. $S^* \subseteq Q$ such that $|S^*| = p_1 + 1$. Then, since S^* is feasible, at most p_1 facilities of level 1 should be open and be part of a path of S^*. This implies that there exists a vertex of V_1 that is in $N_1(S^*)$ contained in at least two different paths of S^*. This contradicts Proposition 2 and the result follows. □

Proof of Proposition 4.12

Proof. Suppose $(\eta, x^T, y_1^T, \cdots, y_k^T) \in X$, and let x_q^T denote the q^{th} component of x^T. Then

$$\eta \leq h(T) + \sum_{q \in Q \setminus T} \rho_q(T) x_q^T = h(T),$$

since $x_q^T = 0$ whenever $q \notin T$. Conversely, now suppose that $\eta \leq h(T)$. Since h is submodular and non-decreasing on Q (Proposition 1), then by Proposition 6

$$h(T) \leq h(S) + \sum_{q \in T \setminus S} \rho_q(S) \ \ \forall S \subseteq Q,$$

thus, for this case $h(T) \leq h(S) + \sum_{q \in Q \setminus S} \rho_q(S) x_q^T \ \ \forall S \subseteq Q$. Then, by hypothesis the result follows. □

Proof of Proposition 4.13

Proof. Suppose $T^* \subseteq Q$ is an optimal solution to Problem (3), then $\eta^* = h(T^*) \geq h(S)$ for all feasible $S \subseteq Q$. In particular, $\eta^* \leq h(T^*)$. Then $(\eta^*, x^{T^*}, y^{T^*}) \in X$ and since T^* satisfies the cardinality constraints, the solution $(\eta^*, x^{T^*}, y^{T^*})$ is feasible for the submodular formulation. Moreover, for any feasible set $S \subseteq Q$ of (3), the solution $(\eta_S = h(S), x^S, y^S)$ is feasible for SF. Then, $\eta^* \geq \eta_S$. The converse proof is similar and the result follows. □

Proof of Proposition 4.15

Proof. We recall that the elements of Q are ordered by non-decreasing values of their coefficients c_{iq}. We denote the t^{th} element according to that ordering as q_t (c_{iq_t}). For $i \in I$, consider the function associated with the right-hand side of (13), $F_i(D) = D + \sum_{q \in Q} (c_{iq} - D)^+ \bar{x}_q$, where D is one of the c_{iq} values, i.e. $D \in [\min_{q \in Q} c_{iq}, \max_{q \in Q} c_{iq}] = [c_{iq_0}, c_{iq_{|Q|-1}}]$. The separation problem for $i \in I$ is thus solved by $\min_D F_i(D)$. Let

d_{t_i} be the corresponding right-hand side of the t^{th} submodular constraint for $i \in I$. Then,

$$\begin{aligned} d_{t_i} &= c_{iq_t} + \sum_{q \in Q} (c_{iq} - c_{iq_t})^+ \bar{x}_q \\ &= c_{iq_t} + \sum_{s=t+1}^{|Q|} (c_{iq_s} - c_{iq_t}) \bar{x}_{q_s} = c_{iq_t} \left(1 - \sum_{s=t+1}^{|Q|} \bar{x}_{q_s} \right) + \sum_{s=t+1}^{|Q|} c_{iq_s} \bar{x}_{q_s}. \end{aligned}$$

For $i \in I$, consider $t^*(i) = \max\{t : \sum_{s=t+1}^{|Q|} \bar{x}_{q_s} \geq 1\}$. For simplicity, we only refer to this value as t^*, having in mind that there could be one different t^* for each $i \in I$. We now show that for $i \in I$, $d_{t^*} \leq d_t$ for all t. If $0 \leq t \leq t^*$, we have

$$\begin{aligned} d_{t^*} &= c_{iq_{t^*}} \left(1 - \sum_{s=t^*+1}^{|Q|} \bar{x}_{q_s} \right) + \sum_{s=t^*+1}^{|Q|} c_{iq_s} \bar{x}_{q_s} \leq c_{iq_t} \left(1 - \sum_{s=t^*+1}^{|Q|} \bar{x}_{q_s} \right) + \sum_{s=t^*+1}^{|Q|} c_{iq_s} \bar{x}_{q_s} \\ &= c_{iq_t} \left(1 - \sum_{s=t+1}^{|Q|} \bar{x}_{q_s} + \sum_{s=t+1}^{t^*} \bar{x}_{q_s} \right) + \sum_{s=t+1}^{|Q|} c_{iq_s} \bar{x}_{q_s} - \sum_{s=t+1}^{t^*} c_{iq_s} \bar{x}_{q_s} \\ &\leq c_{iq_t} \left(1 - \sum_{s=t+1}^{|Q|} \bar{x}_{q_s} \right) + \sum_{s=t+1}^{|Q|} c_{iq_s} \bar{x}_{q_s} = d_t, \end{aligned}$$

where the last inequality follows from $\sum_{s=t+1}^{t^*} (c_{iq_t} - c_{iq_s}) \bar{x}_{q_s} \leq 0$. Similarly, it can be shown that $d_{t^*} \leq d_t$ if $t^* < t$. Therefore, the minimum of $F_i(D)$ occurs when $D = c_{iq_{t^*}}$ with $t^* = \max\{t : \sum_{s=t+1}^{|Q|} \bar{x}_{q_s} \geq 1\}$. Given that there are at most $|Q|$ possible values of c_{iq} for each $i \in I$, the result follows. □

MILP Formulations for the MUpLP

A Path-based Formulation

The path-based formulation has been widely studied in the past for the MUFLP. Approximation algorithms have been developed based on this formulation [4, 10, 53] and their performance have been computationally tested on relatively small and medium sized instances [see e.g., 55, 97].

We define the following binary variables. The variable $x_{iq} = 1$ if $q = j_1, \cdots, j_k \in Q$ is assigned to $i \in I$ and 0 otherwise. Also, $y_{j_r} = 1$ if facility j_r of level r is open. The MUpLP can be modeled as

$$
\begin{aligned}
\text{(PBF) maximize} \quad & \sum_{i \in I} \sum_{q \in Q} c_{iq} x_{iq} - \sum_{r=1}^{k} \sum_{j_r \in V_r} f_{j_r} y_{j_r} & & \\
\text{subject to} \quad & \sum_{q \in Q} x_{iq} = 1 & \forall \;\; i \in I & \qquad (4.15)\\
& \sum_{q \in Q : j_r \in q} x_{iq} \le y_{j_r} & \forall \;\; i \in I, \;\; j_r \in V_r, \;\; r = 1, \cdots, k & \qquad (4.16)\\
& \sum_{j_r \in V_r} y_{j_r} \le p_r & r = 1, \cdots, k & \qquad (4.17)\\
& x_{iq} \ge 0 & \forall \;\; i \in I, \;\; q \in Q & \qquad (4.18)\\
& y_{j_r} \in \{0, 1\} & \forall \;\; j_r \in V_r, \;\; r = 1, \cdots, k. & \qquad (4.19)
\end{aligned}
$$

Constraints (4.15) ensure that exactly one path is assigned to every customer, while constraints (4.16) are the linking constraints which ensure that if a path is assigned to a customer, then all the facilities in such path must be open. Constraints (4.17) are the cardinality restrictions. Finally, note that the variables x_{iq} can be relaxed from binary to continuous variables, as for the UFLP. Note that the number of variables is $|I||Q| + |V|$ and the number of constraints is $|I|(1 + |V|) + k$.

An Arc-based Formulation

The arc-based formulation was studied in [66] as a generalization of the one presented by [3]. The authors define the binary variables $x_{ij_1} = 1$ if $j_1 \in V_1$ is assigned to customer $i \in I$ and 0 otherwise, y_{j_r} as in the PBF is one if facility j_r is open and $z_{iab} = 1$ if customer $i \in I$ uses the arc $(a,b) \in V_r \times V_{r+1}$ and 0 otherwise.

$$\text{(ARB) maximize} \quad \sum_{i \in I} \sum_{j_1 \in V_1} c_{ij_1} x_{ij_1} + \sum_{i \in I} \sum_{r=1}^{k-1} \sum_{(a,b) \in V_r \times V_{r+1}} c_{ab} z_{iab} - \sum_{r=1}^{k} \sum_{j_r \in V_r} f_{j_r} y_{j_r}$$

$$\text{subject to} \quad \sum_{j_1 \in V_1} x_{ij_1} = 1 \qquad \forall \ i \in I \tag{4.20}$$

$$\sum_{b \in V_2} z_{iab} = x_{ia} \qquad \forall \ i \in I, \quad a \in V_1 \tag{4.21}$$

$$\sum_{b \in V_{r+1}} z_{iab} = \sum_{b' \in V_{r-1}} z_{ib'a} \quad \forall \ i \in I, \ a \in V_r, \ r = 2, \cdots, k-1 \tag{4.22}$$

$$x_{ij_1} \leq y_{j_1} \qquad \forall \ i \in I, \quad j_1 \in V_1 \tag{4.23}$$

$$\sum_{a \in V_{r-1}} z_{iab} \leq y_b \qquad \forall \ i \in I, \ b \in V_r \ r = 2, \cdots, k \tag{4.24}$$

$$\sum_{j_r \in V_r} y_{j_r} \leq p_r \qquad r = 1, \cdots, k \tag{4.25}$$

$$x_{ij_1} \geq 0 \ , z_{iab} \geq 0 \qquad \forall \ i \in I, \quad j_1 \in V_1, \ (a,b) \in V_r \times V_{r+1} \tag{4.26}$$

$$y_{j_r} \in \{0,1\} \qquad \forall \ j_r \in V_r, \quad r = 1, \cdots, k. \tag{4.27}$$

Constraints (4.20) ensure that every customer is assigned to a first level facility. The sets of equalities (4.21) and (4.22) ensure the creation and assignation of sequences of facilities for each customer. Constraints (4.23) and (4.24) are the linking constraints, and they model the fact that if an arc or a facility of the first level are assigned to a customer, the corresponding facilities must be open. In this case, the variables x and z can be considered continuous without affecting the integer optimal solution. Also, the number of variables is $|I|(|V_1| + |V| + \sum_{r=1}^{k-1} |V_r||V_{r+1}|)$ and the number of constraints is $|I|(1 + \sum_{r=1}^{k-1} |V_r| + |V|) + k$.

A Flow Based Formulation

The flow-based formulation was recently presented by Kratica et al. [97]. The variables y_{j_r} are the same as before and z is defined as follows: z_{abr} = quantity of goods that facility a in level $r+1$ receives through facility b at level r. In this case, we consider that level 0 corresponds to I.

$$\text{(FBF) maximize} \sum_{r=1}^{k} \sum_{a \in V_{r+1}} \sum_{b \in V_r} c_{ab} z_{abr} - \sum_{r=1}^{k} \sum_{j_r \in V_r} f_{j_r} y_{j_r}$$

$$\text{subject to} \sum_{b \in V_1} z_{ab0} = 1 \qquad \forall \;\; a \in I \tag{4.28}$$

$$\sum_{b \in V_{r-1}} z_{abr-1} = \sum_{b \in V_{r+1}} z_{bar} \qquad \forall \; a \in V_r, \quad r = 1, \cdots, k-1 \tag{4.29}$$

$$z_{abr} \leq m y_b \qquad \forall \;\; a \in V_{r+1}, \; b \in V_r \; r = 1, \cdots, k-1 \tag{4.30}$$

$$\sum_{j_r \in V_r} y_{j_r} \leq p_r \qquad r = 1, \cdots, k \tag{4.31}$$

$$z_{ijr} \geq 0 \qquad \forall \;\; i \in V_{r+1}, j \in V_r, \; r = 0, \cdots, k-1 \tag{4.32}$$

$$y_{j_r} \in \{0, 1\} \qquad \forall \;\; j_r \in V_r, \quad r = 1, \cdots, k. \tag{4.33}$$

Similarly, constraints (4.28) and (4.29) define and assign the sequence of facilities to every customer while inequalities (4.30) are the linking constraints. Also, the number of variables is $|V| + \sum_{r=1}^{k-1} |V_r||V_{r+1}| + |I||V_1|$ and the number of constraints is $|I| + \sum_{r=1}^{k-1} |V_r| + \sum_{r=1}^{k} |V_r||V_{r+1}| + k$.

Computational Results

We present in Table 4.3 the detailed computational results for the MpMP. The first column describes the type of instance through its five subcolumns. The next 12 columns provide the CPU time in seconds needed to solve the instance, the percent duality gap relative to the LP relaxation bound and the number of nodes in the branch-and-cut tree for all four models. Finally, the last two columns provide the percent deviation of the greedy bound with respect to the optimal solution value and the optimal value obtained. Whenever CPLEX is not able to solve an instance within 1000 seconds, we write TIME in the corresponding entry of the table. If the computer runs out of memory we write MEM. The instances for which the optimal value is left blank are those where none of the formulations was able to solve it within 10 times the time limit (i.e. 10,000 seconds).

Table 4.3: Comparison of MILP formulations for the MpMP

Instance					SFD			FBF			ARB			PBF			Greedy	OPT.
Type	Levels	Customers	Pot. Facil.	P	Sec	BB nodes	LP %gap	Sec	BB nodes	LP %gap	Sec	BB nodes	LP %gap	Sec	BB nodes	LP %gap	% gap	
TightC2	2	2	3-1	2-1	0.01	0	0.00	0.01	0	0.00	0.01	0	0.00	0.02	0	0.00	25.00	4.00
TightC3	2	6	5-1	3-1	0.01	0	0.00	0.01	0	0.00	0.01	0	0.00	0.02	0	0.00	29.62	54.00
TightC4	2	12	7-1	4-1	0.01	0	0.00	0.01	0	0.00	0.01	0	0.00	0.02	0	0.00	31.64	768.00
capa	2	1000	70-30	2-1	0.53	0	0.45	4.76	2	6.79	70.47	0	0.00	43.07	0	0.00	0.53	55291.46
capa	2	1000	70-30	3-2	0.70	13	0.08	3.70	3	3.43	202.40	0	0.00	17.07	0	0.00	0.02	57282.10
capa	2	1000	70-30	70-30	0.02	0	0.00	0.16	0	0.00	3.85	0	0.00	2.57	0	0.00	0.00	59318.10
capa	2	1000	50-50	2-1	0.95	2	0.54	3.87	2	6.60	137.62	0	0.00	56.32	0	0.00	0.34	55291.46
capa	2	1000	50-50	3-2	2.25	36	0.12	2.90	0	3.21	30.03	0	0.00	21.47	0	0.00	0.00	57297.69
capa	2	1000	50-50	50-50	0.02	0	0.00	0.11	0	0.00	4.40	0	0.00	3.06	0	0.00	0.00	59197.50
capb	2	1000	70-30	2-1	0.61	0	0.10	5.74	0	8.78	35.33	0	0.00	38.52	0	0.00	0.00	54148.14
capb	2	1000	70-30	5-2	1.00	52	0.11	4.77	3	1.70	204.69	0	0.00	16.10	0	0.00	0.18	58347.59
capb	2	1000	70-30	70-30	0.02	0	0.00	0.16	0	0.00	3.85	0	0.00	2.53	0	0.00	0.00	59358.21
capc	2	1000	70-30	2-1	0.70	0	0.49	7.47	2	6.75	79.61	0	0.00	45.48	0	0.00	0.54	53964.64
capc	2	1000	70-30	3-2	1.81	19	0.07	4.71	0	4.53	328.99	0	0.00	20.98	0	0.00	0.00	55244.52
capc	2	1000	70-30	70-30	0.02	0	0.00	0.14	0	0.00	3.88	0	0.00	2.56	0	0.00	0.00	57868.14
capa	3	1000	55-30-15	2-1-1	4.28	11	0.60	4.76	0	6.53	419.71	0	0.00	MEM	MEM	MEM	0.00	56391.98
capa	3	1000	55-30-15	3-2-1	6.58	18	0.23	3.88	7	3.31	TIME	TIME	0.00	MEM	MEM	MEM	0.00	58338.01
capa	3	1000	55-30-15	55-30-15	0.09	0	0.00	0.23	0	0.00	4.79	0	0.00	34.48	0	0.00	0.00	60333.88
capb	3	1000	60-30-10	2-1-1	4.65	5	0.00	2.87	0	8.51	147.45	0	0.00	MEM	MEM	MEM	0.00	55411.46
capb	3	1000	60-30-10	5-2-2	3.96	29	0.05	3.09	0	1.62	468.61	0	0.00	MEM	MEM	MEM	0.00	59580.61
capb	3	1000	60-30-10	60-30-10	0.08	0	0.00	0.23	0	0.00	4.62	0	0.00	24.98	0	0.00	0.00	60564.36
capc	3	1000	60-30-10	2-1-1	5.69	28	0.50	7.19	2	5.86	244.81	0	0.00	MEM	MEM	MEM	0.63	55070.27
capc	3	1000	60-30-10	5-2-2	9.19	94	0.10	4.06	0	2.02	TIME	TIME	0.00	MEM	MEM	MEM	0.00	57316.09
capc	3	1000	60-30-10	60-30-10	0.08	0	0.00	0.25	0	0.00	4.79	0	0.00	43.90	0	0.00	0.00	58496.09
RAND	4	500	40-30-20-10	2-2-1-1	22.52	33	1.87	22.29	27	4.24	685.52	0	0.00	MEM	MEM	MEM	0.43	254133.58
RAND	4	500	40-30-20-10	3-3-3-3	10.31	10	0.01	0.69	0	1.89	708.05	0	0.00	MEM	MEM	MEM	0.00	260372.95
RAND	4	500	40-30-20-10	10-5-3-2	54.43	247	1.22	16.22	382	1.44	839.71	0	0.00	MEM	MEM	MEM	0.08	261566.17
RAND	4	500	40-30-20-10	40-30-20-10	1.48	0	0.00	0.09	0	0.00	2.48	0	0.00	MEM	MEM	MEM	0.00	265388.41
RAND	3	500	50-30-20	4-2-1	11.00	78	3.14	869.89	1517	5.06	680.16	0	0.00	MEM	MEM	MEM	0.21	189997.53
RAND	3	500	50-30-20	5-5-2	8.70	112	1.26	291.70	1069	2.81	745.60	3	0.01	MEM	MEM	MEM	0.39	194501.88
RAND	3	500	50-30-20	11-9-6	4.96	214	0.27	11.44	327	0.64	876.80	3	0.01	MEM	MEM	MEM	0.05	198841.46
RAND	3	500	50-30-20	50-30-20	0.14	0	0.00	0.11	0	0.00	2.28	0	0.00	30.30	0	0.00	0.00	200117.56
RAND	2	500	70-30	3-1	6.93	165	5.34	TIME	TIME	9.77	127.77	3	0.11	90.31	3	0.11	0.00	121577.18
RAND	2	500	70-30	5-4	10.92	665	0.54	261.21	707	3.44	859.37	0	0.00	13.37	0	0.00	0.83	130102.25
RAND	2	500	70-30	10-2	40.27	684	4.86	TIME	TIME	6.01	387.76	2	0.01	21.11	2	0.01	0.37	126632.92
RAND	2	500	70-30	70-30	0.02	0	0.00	0.08	0	0.00	1.97	0	0.00	1.28	0	0.00	0.00	134735.92
RAND	2	500	50-50	2-1	6.30	125	4.31	403.75	899	10.87	192.22	0	0.00	74.61	0	0.00	0.00	120632.62
RAND	2	500	50-50	5-5	0.50	24	0.02	0.95	0	3.08	11.15	0	0.00	11.75	0	0.00	0.34	131165.45
RAND	2	500	50-50	11-9	5.59	847	0.32	773.62	17458	1.24	17.80	2	0.00	16.52	2	0.00	0.13	133656.44
RAND	2	500	50-50	50-50	0.01	0	0.00	0.05	0	0.00	2.25	0	0.00	1.56	0	0.00	0.00	135339.65
RAND	4	500	60-30-20-10	4-2-1-1	99.97	106	3.12	767.20	604	4.27	TIME	TIME	0.00	MEM	MEM	MEM	0.74	253757.25
RAND	4	500	60-30-20-10	8-2-2-1	200.08	192	3.51	TIME	TIME	4.04	TIME	TIME	0.00	MEM	MEM	MEM	0.91	254373.99
RAND	4	500	60-30-20-10	15-10-5-2	80.03	401	1.12	13.73	221	1.31	TIME	TIME	0.00	MEM	MEM	MEM	0.10	261602.65
RAND	4	500	60-30-20-10	60-30-20-10	2.34	0	0.00	0.13	0	0.00	3.17	0	0.00	MEM	MEM	MEM	0.00	265088.22
RAND	4	1000	40-30-20-10	2-2-1-1	19.97	10	0.45	15.68	34	2.62	TIME	TIME	0.00	MEM	MEM	MEM	0.00	515497.20
RAND	4	1000	40-30-20-10	3-3-3-3	12.29	7	0.00	1.54	0	1.22	TIME	TIME	0.00	MEM	MEM	MEM	0.00	522919.72
RAND	4	1000	40-30-20-10	10-5-3-2	38.48	74	0.47	468.54	1955	0.69	TIME	TIME	0.00	MEM	MEM	MEM	0.00	525698.59
RAND	4	1000	40-30-20-10	40-30-20-10	1.53	0	0.00	0.19	0	0.00	4.96	0	0.00	MEM	MEM	MEM	0.00	529364.59
RAND	3	1000	50-30-20	4-2-1	13.76	43	2.87	TIME	TIME	4.95	TIME	TIME	0.00	MEM	MEM	MEM	0.12	379605.89
RAND	3	1000	50-30-20	5-5-2	16.24	135	1.28	TIME	TIME	2.96	TIME	TIME	0.00	MEM	MEM	MEM	0.31	387546.69
RAND	3	1000	50-30-20	11-9-6	8.94	225	0.34	173.21	1152	0.87	TIME	TIME	0.00	MEM	MEM	MEM	0.05	395917.40
RAND	3	1000	50-30-20	50-30-20	0.11	0	0.00	0.22	0	0.00	4.63	0	0.00	157.36	0	0.00	0.00	399382.14
RAND	2	1000	70-30	3-1	10.14	63	5.69	TIME	TIME	9.89	511.03	0	0.00	269.15	0	0.00	0.00	243200.59
RAND	2	1000	70-30	6-6	0.72	21	0.00	3.62	0	2.27	TIME	TIME	0.00	29.13	0	0.00	0.00	263774.28
RAND	2	1000	70-30	10-2	102.99	655	4.80	TIME	TIME	5.93	TIME	TIME	0.00	106.85	0	0.00	0.76	253903.07
RAND	2	1000	70-30	70-30	0.02	0	0.00	0.16	0	0.00	3.95	0	0.00	2.62	0	0.00	0.00	269897.17
RAND	3	1500	50-30-20	4-2-1	42.54	62	3.42	TIME	TIME	5.25	TIME	TIME	0.07	MEM	MEM	MEM	0.70	568051.40
RAND	3	1500	50-30-20	5-5-2	60.64	180	1.95	TIME	TIME	3.28	TIME	TIME	0.01	MEM	MEM	MEM	0.19	579839.88
RAND	3	1500	50-30-20	11-9-6	8.38	151	0.29	38.87	192	0.57	TIME	TIME	0.01	MEM	MEM	MEM	0.05	596088.75
RAND	3	1500	50-30-20	50-30-20	0.11	0	0.00	0.34	0	0.00	6.96	0	0.00	MEM	MEM	MEM	0.00	599510.09
RAND	2	1500	70-30	3-1	33.22	75	5.47	TIME	TIME	10.19	TIME	TIME	0.00	528.73	0	0.00	1.69	363930.34
RAND	2	1500	70-30	5-4	124.08	1601	0.67	TIME	TIME	3.85	TIME	TIME	0.02	304.40	3	0.02	0.09	389635.00
RAND	2	1500	70-30	10-2	263.80	591	5.41	TIME	TIME	6.60	TIME	TIME	0.00	252.78	0	0.00	0.04	378467.47
RAND	2	1500	70-30	70-30	0.01	0	0.00	0.23	0	0.00	5.93	0	0.00	3.90	0	0.00	0.00	405229.72
RAND	3	1500	100-70-30	4-2-1	345.13	102	4.39	TIME	TIME	6.96	TIME	TIME	0.09	MEM	MEM	MEM	0.27	570422.86
RAND	3	1500	100-70-30	10-5-2	TIME	TIME	2.98	TIME	TIME	3.86	TIME	TIME	0.00	MEM	MEM	MEM	0.26	589420.14
RAND	3	1500	100-70-30	35-9-6	TIME	TIME	-	TIME	TIME	-	TIME	TIME	-	MEM	MEM	MEM	-	-
RAND	3	1500	100-70-30	100-70-30	1.03	0	0.00	0.91	0	0.00	33.82	0	0.00	MEM	MEM	MEM	0.00	613079.56
RAND	2	1500	120-80	3-1	153.60	162	5.82	TIME	TIME	11.26	TIME	TIME	0.20	MEM	MEM	MEM	0.57	365305.31
RAND	2	1500	120-80	6-6	6.33	91	0.01	61.33	0	3.30	TIME	TIME	0.00	MEM	MEM	MEM	0.10	398033.28
RAND	2	1500	120-80	7-2	TIME	TIME	4.74	TIME	TIME	7.44	TIME	TIME	0.00	MEM	MEM	MEM	0.00	381010.59
RAND	2	1500	120-80	120-80	0.05	0	0.00	0.47	0	0.00	31.09	0	0.00	19.89	0	0.00	0.00	411635.15
RAND	2	2000	120-80	4-1	323.24	168	6.72	TIME	TIME	10.91	TIME	TIME	0.09	MEM	MEM	MEM	0.14	488987.61
RAND	2	2000	120-80	7-2	TIME	TIME	-	TIME	TIME	-	TIME	TIME	-	MEM	MEM	MEM	-	-
RAND	2	2000	120-80	10-10	14.74	245	0.02	53.34	0	2.13	TIME	TIME	0.00	MEM	MEM	MEM	0.10	537210.26
RAND	2	2000	120-80	120-80	0.05	0	0.00	0.62	0	0.00	41.57	0	0.00	25.88	0	0.00	0.00	548888.20

Table 4.4: Comparison of MILP formulations for the MUpLP

Instance					SFML			FBF			ARB			PBF			Greedy	OPT.
Type	Levels	Customers	Pot. Facil.	P	Sec	BB nodes	LP %Dev	Sec	BB nodes	LP %Dev	Sec	BB nodes	LP %Dev	Sec	BB nodes	LP %Dev	%Dev.	
capa	2	1000	70-30	2-1	1.79	9	3.72	4.15	0	16.81	26.65	0	0.00	33.09	0.00	0.00	0.16	48680.89
capa	2	1000	70-30	3-2	5.83	89	6.80	103.72	17	16.43	21.82	0	0.00	18.91	0.00	0.00	0.12	48899.33
capa	2	1000	70-30	70-30	449.51	12413	15.91	72.59	19	16.43	19.97	0	0.00	19.06	0.00	0.00	0.12	48899.33
capa	2	1000	50-50	2-1	2.54	19	4.05	3.84	5	16.41	38.14	0	0.00	41.82	0.00	0.00	0.28	48740.63
capa	2	1000	50-50	3-2	5.84	106	6.94	38.77	17	15.94	26.54	0	0.00	22.79	0.00	0.00	0.36	49013.55
capa	2	1000	50-50	50-50	401.15	12673	15.78	38.42	19	15.94	26.43	0	0.00	22.20	0.00	0.00	0.36	49013.55
capb	2	1000	70-30	2-1	2.81	36	1.47	5.88	2	12.80	31.92	0	0.00	54.88	0.00	0.00	0.33	51440.80
capb	2	1000	70-30	5-2	6.69	154	4.93	31.50	7	9.11	31.48	0	0.00	16.11	0.00	0.00	0.20	53614.63
capb	2	1000	70-30	70-30	143.38	14614	8.72	28.72	7	9.11	32.96	0	0.00	16.71	0.00	0.00	0.20	53614.63
capc	2	1000	70-30	2-1	2.23	18	1.31	5.60	2	9.54	41.68	0	0.00	30.12	0.00	0.00	0.32	52016.07
capc	2	1000	70-30	3-2	4.74	141	2.35	18.00	11	8.68	130.65	0	0.00	25.97	0.00	0.00	0.08	52508.38
capc	2	1000	70-30	70-30	922.76	85913	7.48	39.00	7	7.85	151.96	0	0.00	16.22	0.00	0.00	0.17	52984.68
capa	3	1000	55-30-15	2-1-1	21.58	23	4.68	7.30	3	23.16	254.30	0	0.00	MEM	MEM	MEM	1.00	45170.63
capa	3	1000	55-30-15	3-2-1	312.86	602	9.65	83.23	11	22.53	48.45	0	0.00	MEM	MEM	MEM	1.33	45541.44
capa	3	1000	55-30-15	55-30-15	TIME	TIME	22.53	94.30	11	22.53	155.22	0	0.00	998.30	0.00	0.00	1.33	45541.44
capb	3	1000	60-30-10	2-1-1	17.55	10	1.39	3.92	0	14.91	62.57	0	0.00	MEM	MEM	MEM	0.86	50867.76
capb	3	1000	60-30-10	5-2-2	56.86	348	5.31	35.43	6	11.26	53.55	0	0.00	MEM	MEM	MEM	0.72	53047.51
capb	3	1000	60-30-10	60-30-10	TIME	TIME	11.18	34.46	5	11.26	53.63	0	0.00	310.57	0.00	0.00	0.72	53047.51
capc	3	1000	60-30-10	2-1-1	21.34	54	1.36	7.82	3	10.86	160.68	0	0.00	MEM	MEM	MEM	0.48	51534.88
capc	3	1000	60-30-10	5-2-2	570.04	4909	4.51	143.83	27	10.04	702.50	0	0.00	MEM	MEM	MEM	0.54	52006.77
capc	3	1000	60-30-10	60-30-10	TIME	TIME	9.89	152.49	30	10.04	758.16	0	0.00	454.98	0.00	0.00	0.54	52006.77
RAND	4	500	40-30-20-10	2-2-1-1	89.51	59	2.40	54.90	63	4.98	727.62	0	0.00	MEM	MEM	MEM	16.98	250301.90
RAND	4	500	40-30-20-10	3-3-3-3	85.96	46	1.33	44.46	178	3.47	659.65	0	0.00	MEM	MEM	MEM	18.23	254293.79
RAND	4	500	40-30-20-10	10-5-3-2	305.07	891	2.97	87.34	230	3.47	769.80	0	0.01	MEM	MEM	MEM	18.23	254279.21
RAND	4	500	40-30-20-10	40-30-20-10	679.12	2661	3.74	94.82	292	3.47	29.73	0	0.00	MEM	MEM	MEM	18.23	254293.79
RAND	3	500	50-30-20	4-2-1	48.19	408	3.71	602.25	953	5.99	763.09	5	0.06	MEM	MEM	MEM	1.00	187181.03
RAND	3	500	50-30-20	5-5-2	128.68	2070	3.02	TIME	TIME	4.90	902.23	3	0.05	MEM	MEM	MEM	2.13	189357.10
RAND	3	500	50-30-20	11-9-6	TIME	TIME	3.86	TIME	TIME	4.56	527.34	2	0.00	MEM	MEM	MEM	2.48	190033.12
RAND	3	500	50-30-20	50-30-20	TIME	TIME	4.63	862.65	2797	4.56	40.23	0	0.00	517.87	2.00	0.00	2.48	190033.12
RAND	2	500	70-30	3-1	10.08	137	5.77	TIME	TIME	10.56	95.89	3	0.07	44.60	3.00	0.07	0.26	120201.82
RAND	2	500	70-30	5-4	26.55	1163	2.53	593.80	3051	5.68	500.26	5	0.03	21.15	5.00	0.03	5.36	126760.18
RAND	2	500	70-30	10-2	173.21	1202	5.77	TIME	TIME	7.19	341.16	0	0.00	13.37	0.00	0.00	3.80	124730.94
RAND	2	500	70-30	70-30	77.66	7350	4.88	793.25	1574	5.07	19.00	3	0.00	9.42	3.00	0.00	5.95	127577.26
RAND	2	500	50-50	2-1	19.06	351	4.62	371.84	639	11.55	235.53	0	0.00	109.37	0.00	0.00	0.00	119374.93
RAND	2	500	50-50	5-5	5.74	257	2.16	392.11	2220	5.57	19.80	0	0.00	11.97	0.00	0.00	5.98	127436.17
RAND	2	500	50-50	11-9	111.84	6508	3.85	680.91	3696	5.03	10.44	0	0.00	8.47	0.00	0.00	6.51	128170.61
RAND	2	500	50-50	50-50	462.68	29586	4.96	TIME	TIME	5.03	9.44	0	0.00	8.75	0.00	0.00	6.51	128170.61
RAND	4	500	60-30-20-10	4-2-1-1	443.91	353	3.45	959.14	421	5.00	TIME	TIME	0.00	MEM	MEM	MEM	17.47	249988.31
RAND	4	500	60-30-20-10	8-2-2-1	TIME	TIME	4.10	182.94	71	4.95	TIME	TIME	0.00	MEM	MEM	MEM	17.51	250103.66
RAND	4	500	60-30-20-10	15-10-5-2	TIME	TIME	3.31	199.06	260	3.65	TIME	TIME	0.00	MEM	MEM	MEM	18.62	253526.11
RAND	4	500	60-30-20-10	60-30-20-10	TIME	TIME	3.95	391.08	904	3.65	45.74	0	0.00	MEM	MEM	MEM	18.62	253526.11
RAND	4	1000	40-30-20-10	2-2-1-1	125.99	17	0.76	63.48	23	3.03	TIME	TIME	0.00	MEM	MEM	MEM	5.72	511471.20
RAND	4	1000	40-30-20-10	3-3-3-3	132.15	28	0.65	35.62	35	2.04	TIME	TIME	0.00	MEM	MEM	MEM	6.60	516694.46
RAND	4	1000	40-30-20-10	10-5-3-2	78.34	210	1.41	72.17	59	1.75	TIME	TIME	0.00	MEM	MEM	MEM	6.84	518196.04
RAND	4	1000	40-30-20-10	40-30-20-10	197.70	988	1.84	127.14	379	1.75	66.28	0	0.00	MEM	MEM	MEM	6.84	518196.04
RAND	3	1000	50-30-20	4-2-1	52.20	199	3.01	TIME	TIME	5.26	TIME	TIME	0.00	MEM	MEM	MEM	0.79	377413.96
RAND	3	1000	50-30-20	5-5-2	169.65	1446	1.97	TIME	TIME	3.79	TIME	TIME	0.00	MEM	MEM	MEM	2.28	383252.89
RAND	3	1000	50-30-20	11-9-6	462.19	11965	2.57	TIME	TIME	3.20	TIME	TIME	0.00	MEM	MEM	MEM	2.88	385594.58
RAND	3	1000	50-30-20	50-30-20	TIME	TIME	3.20	TIME	TIME	3.20	90.14	2	0.00	TIME	TIME	0.00	2.88	385594.58
RAND	2	1000	70-30	3-1	75.46	337	5.71	TIME	TIME	10.16	365.54	0	0.00	116.61	0.00	0.00	1.56	242132.88
RAND	2	1000	70-30	6-6	3.49	94	1.50	TIME	TIME	3.97	TIME	TIME	0.00	37.72	0.00	0.00	7.84	258805.41
RAND	2	1000	70-30	10-2	TIME	TIME	5.39	TIME	TIME	6.65	TIME	TIME	0.00	86.42	0.00	0.00	5.19	251578.41
RAND	2	1000	70-30	70-30	TIME	TIME	3.55	TIME	TIME	3.64	45.33	2	0.01	24.65	2.00	0.01	8.15	259695.92
RAND	3	1500	50-30-20	4-2-1	407.87	375	3.65	TIME	TIME	5.54	TIME	TIME	0.02	MEM	MEM	MEM	13.11	565487.40
RAND	3	1500	50-30-20	5-5-2	129.16	695	2.43	TIME	TIME	3.82	TIME	TIME	0.03	MEM	MEM	MEM	14.67	575818.92
RAND	3	1500	50-30-20	11-9-6	14.56	167	1.55	845.54	1818	1.91	TIME	TIME	0.00	MEM	MEM	MEM	16.32	587209.46
RAND	3	1500	50-30-20	50-30-20	21.91	384	1.91	251.00	1650	1.91	76.64	0	0.00	MEM	MEM	MEM	16.32	587201.42
RAND	2	1500	70-30	3-1	511.74	689	5.75	TIME	TIME	10.54	TIME	TIME	0.02	455.15	3.00	0.02	3.05	362148.34
RAND	2	1500	70-30	5-4	51.59	746	1.25	TIME	TIME	4.57	TIME	TIME	0.00	148.19	0.00	0.00	8.69	386302.21
RAND	2	1500	70-30	10-2	TIME	TIME	6.09	TIME	TIME	7.39	TIME	TIME	0.01	366.35	3.00	0.01	5.89	374898.37
RAND	2	1500	70-30	70-30	8.41	374	3.02	TIME	TIME	3.11	46.80	0	0.00	25.40	0.00	0.00	10.05	392225.23
RAND	3	1500	100-70-30	4-2-1	TIME	TIME	-	TIME	TIME	-	TIME	TIME	-	MEM	MEM	MEM	-	-
RAND	3	1500	100-70-30	10-5-2	TIME	TIME	-	TIME	TIME	-	TIME	TIME	-	MEM	MEM	MEM	-	-
RAND	3	1500	100-70-30	35-9-6	TIME	TIME	-	TIME	TIME	-	TIME	TIME	-	MEM	MEM	MEM	-	-
RAND	3	1500	100-70-30	100-70-30	TIME	TIME	2.71	TIME	TIME	2.72	414.06	0	0.00	MEM	MEM	MEM	10.87	595451.10
RAND	2	1500	120-80	3-1	TIME	TIME	-	TIME	TIME	-	TIME	TIME	-	MEM	MEM	MEM	-	-
RAND	2	1500	120-80	6-6	10.00	87	0.93	TIME	TIME	4.31	TIME	TIME	0.00	MEM	MEM	MEM	11.94	393452.37
RAND	2	1500	120-80	7-2	TIME	TIME	-	TIME	TIME	-	TIME	TIME	-	MEM	MEM	MEM	-	-
RAND	2	1500	120-80	120-80	TIME	TIME	3.54	TIME	TIME	3.59	335.14	0	0.00	181.97	0.00	0.00	12.60	396423.32
RAND	2	2000	120-80	4-1	TIME	TIME	-	TIME	TIME	-	TIME	TIME	-	MEM	MEM	MEM	-	-
RAND	2	2000	120-80	7-2	TIME	TIME	-	TIME	TIME	-	TIME	TIME	-	MEM	MEM	MEM	-	-
RAND	2	2000	120-80	10-10	27.19	319	1.08	TIME	TIME	3.25	TIME	TIME	0.00	MEM	MEM	MEM	8.60	530624.08
RAND	2	2000	120-80	120-80	TIME	TIME	2.96	TIME	TIME	2.98	370.47	0	0.00	437.49	0.00	0.00	8.85	532073.27

Chapter 5

An exact algorithm for multi-level uncapacitated facility location

The content of this chapter was submitted for publication under the title "An Exact Algorithm for Multi-level Uncapacitated Facility Location", *Transportation Science*, January 2017, [137].

Abstract

We study a general class of multi-level uncapacitated p-location problems in which the selection of links between levels of facilities is part of the decision process. An exact algorithm based on a Benders reformulation is proposed to solve large-scale instances of the general problem and some well-known particular cases. We exploit the network flow structure of the reformulation to efficiently generate Pareto-optimal cuts. Extensive computational experiments are performed to assess the performance of several different variants of the Benders algorithm. Results obtained on benchmark instances with up to 3,000 customers, 250 potential facilities and four levels confirm its efficiency.

5.1 Introduction

Multi-level facility location problems (MLFLPs) lie at the heart of network design planning in transportation and telecommunications systems. Given a set of customers that have a service requirement and a set of potential facilities partitioned into k levels, MLFLPs consist of selecting a set facilities to open at each level so that every customer is assigned to a sequence of opened facilities, exactly one from each level, while optimizing an objective function. MLFLPs are a special case of the important class of *Hierarchical facility location problems* (HFLPs) where different hierarchies of facilities and their interactions are considered. Applications of HFLPs arise naturally in supply chain management [123] where the interactions between warehouses, distribution centers and retail stores play a major role, and in health care systems [143] which usually require serving users from different levels of clinics and hospitals. Other examples arise in hierarchical telecommunication networks [34, 77], freight transportation [68, 70], and solid waste management systems [22]. The two surveys of Şahin and Süral [47] and Zanjirani Farahani et al. [168] provide classifications as well as overviews of models, applications, and algorithms for HFLPs.

A fundamental problem of MLFLPs is the so-called warehouse and plant location problem introduced by Kaufman et al. [90], also denoted as the *two-level uncapacitated facility location problem* (TUFLP). A natural extension to more than two levels of facilities corresponds to the *multi-level uncapacitated facility location problem* (MUFLP) studied by Aardal et al. [4]. Ortiz-Astorquiza et al. [136] recently introduced a class of MLFLPs in which cardinality constraints are introduced at each level. This problem is denoted as the *multi-level uncapacitated p-location problem* (MUpLP).

In this paper we present an extension of the MUpLP in which the selection of links (or edges) between levels of facilities, with their associated set-up costs, is part of the decision process. The problem is defined as follows. Let $I = \{1, \cdots, m\}$ be the set of customers, and $V_1, \cdots, V_k$ be the sets of sites among which facilities of levels 1 to k can be selected (or opened), with $V = \cup_{r=1}^{k} V_r$. Also, consider $c_{ij_1 \cdots j_k}$ to be the profit associated with the allocation of customer i to the sequence of facilities $j_1, \cdots, j_k$, where $j_r \in V_r$. Now, let $p = (p_1, \cdots, p_k)$ be a vector of positive integers, f_{j_r} be the non-negative fixed cost associated with opening facility j_r at level r, and d^r_{ab} be the cost of opening the edge between facilities $a \in V_r$ and $b \in V_{r+1}$. The *multi-level uncapacitated p-location problem with edge set-up costs* (MUpLP-E) consists of selecting sets of facilities and edges to open, such that no more than p_r facilities are opened at level r and of assigning each customer to a set of open facilities, exactly one at each level, while maximizing the total profit, minus the total setup cost. An edge can be opened if both of the corresponding facilities are open. When a customer is assigned to a set of opened facilities all the edges in the corresponding sequence must be activated. We assume throughout the paper that the profits c are additive, that is, $c_{ij_1 \cdots j_k} = c_{ij_1} + \cdots + c_{j_{k-1} j_k}$. Finally, we work with the maximization version of these problems. Similar to the case of the MUFLP, the maximization and minimization versions of the MUpLP-E are equivalent from an optimization point of view [170].

MUpLPs-E belong to a broader class of optimization problems referred to as *general network design problems* (GNDPs) where both the facilty location and network design decisions are predominant and non-trivial [39]. Examples of GNDPs are *facility location-network design problems* [42, 121], *location-vehicle routing problems* [12], *hub location problems* [32, 38], and *hub arc location problems* [33, 40], among others. However, to the best of our knowledge, only a few papers have focused on MLFLPs that incorporate non-trivial network design decisions. For example, Barros and Labbé [21] consider the TUFLP with set-up costs on the edges and present mixed integer linear programing (MILP) formulations and a branch-and-bound (BB) algorithm based on Lagrangean relaxations to solve it. Gendron et al. [69] study a variant of the TUFLP with set-up costs on the edges, denoted as TUFLP-S, in which single assignment constraints are considered. These constraints force each first-level facility to be connected to at most one second-level facility. The authors present a multilayer variable neighborhood search metaheuristic to solve the problem. Gendron et al. [71] present six different MILP formulations for the TUFLP-S and computationally compare them using several sets of benchmark instances. Gendron et al. [70] describe an exact algorithm for the same TUFLP-S in which a Lagrangean relaxation is used as a bounding procedure in an enumeration tree. For other MLFLPs that do not include non-trivial network design decisions we refer to [136] and references therein.

The main contribution of this article is twofold. First, we introduce the MUpLP-E, a general class of MLFLPs in which the selection of links between levels of facilities is part of the decision process. Second, we present an exact branch-and-cut algorithm based on a Benders reformulation to solve large-scale instances of the MUpLP-E. This reformulation is obtained by projecting out a large set of binary variables from an extension of the arc-based formulation introduced in [3] for the TUFLP. Exact separation procedures are developed to efficiently generate feasibility and optimality cuts at fractional and integer points. We show that the well-known cut-set inequalities are sufficient to guarantee feasibility of the primal subproblem and thus, these can replace the standard feasibility cuts. In addition, we exploit the network flow structure of the reformulation to efficiently generate Pareto-optimal cuts using different strategies. In order to assess the performance of our algorithm, we have performed extensive computational experiments on several sets of benchmark instances for the general MUpLP-E and some special cases previously studied in the literature.

Our motivation for applying Benders decomposition [24] lies mainly in the problem structure which will be analyzed in the paper. The increased attention that this method has attracted in the last few years is noteworthy. It has been applied to several supply chain problems, which embed sophisticated versions of MLFLP [for example, 44, 124, 142]. Reference [72] is one of the earliest papers ever published in the field of supply chain management, and it also happens to be one of the first in which the applicability of Benders decomposition is demonstrated. Moreover, it was only very recently that Fischetti et al. [61, 62] computationally showed the importance of a proper Benders implementation for single-level FLPs.

The remainder of the paper is organized as follows. Section 5.2 provides a formal definition of the MUpLP-E and presents an MILP formulation. In Section 5.3 we describe the Benders reformulation, the branch-and-cut algorithm, and the separation of the Benders cuts. In Section 5.4 we present the enhancements of the algorithm and discuss the main characteristics of the problem that can be exploited in the implementation. Section 5.5 presents the results of extensive computational experiments performed to compare different versions of the Benders algorithm to each other, to that of the MILP formulation and to those of previously proposed methods for special cases of the problem. Conclusions follow in Section 5.6.

5.2 Problem Definition

Let $G = (V \cup I, E)$ be a graph with vertex set $V \cup I$ partitioned into $k+1$ levels, where I represents the set of customers, V is partitioned into $\{V_1, \cdots, V_k\}$, corresponding to the sets of potential facilities at levels 1 to k, and E is the set of edges. Also, for $r = 1, \cdots, k-1$, let $E_r = \{\{a, b\} \in E : a \in V_r \text{ and } b \in V_{r+1}\}$, and let $E_0 = \{\{i, b\} \in E : i \in I \text{ and } b \in V_1\}$. Let $c_{ij_1 \cdots j_k}$ be the profit associated with the allocation of customer i to the sequence of facilities $j_1, \cdots, j_k$ and $p = (p_1, p_2, \cdots, p_k)$ be a vector of positive integers. Now, for $r = 1, \cdots, k$, and $r = 1, \cdots, k-1$, let f_{j_r} and d^r_{ab} be the non-negative fixed cost associated with the opening of facility j_r at level r and those of opening the edge $\{a, b\} \in E_r$, respectively. The MUpLP-E consists of selecting sets of facilities and edges to open, so that no more than p_r facilities are opened at level r and of assigning each customer to a set of open facilities, exactly one at each level, while maximizing the total profit minus the total setup cost. An edge $\{a, b\} \in E_r$ can be opened if both of the corresponding facilities are open. Also, when a customer is assigned to a set of open facilities all the edges in the corresponding sequence must be activated.

We assume that the profit (or cost) c is additive with respect to the profits on the edges. Typically, this assumption is made for the minimization version of the problem but as explained in [136], we can properly transform the problem to move from one version to the other from an optimization point of view. Thus, the following assumption holds throughout the paper unless otherwise stated.

Assumption 5.1. *We assume that c is additive, that is, for all $i \in I$ and $j_r \in V_r$ for $r = 1, \cdots, k$ we have $c_{ij_1 \cdots j_k} = c_{ij_1} + c_{j_1 j_2} + \cdots + c_{j_{k-1} j_k}$*

When all the values of d^r_{ab} are set to zero, in conjunction with Assumption 5.1, there exists an optimal solution satisfying the single-assignment property from lower to upper level facilities. We now extend the arc-based formulation (ABF) [3, 66] to the MUpLP-E as follows. We define binary location variables y_{j_r} equal to 1 if and only if a facility is located at node j_r, and binary link activation variables w^r_{ab} equal to 1 if and only if edge $\{a, b\} \in E_r$ is selected. We also define binary assignment variables x_{ia} equal to 1 if and only if customer $i \in I$ is assigned to first-level facility $a \in V_1$. Finally, for $r = 1, \cdots, k-1$, we introduce binary arc variables z^r_{iab} equal to one if customer $i \in I$ uses the edge $\{a, b\} \in E_r$ and 0 otherwise. The MUpLP-E can be stated as follows:

$$\max \sum_{i\in I}\sum_{j_1\in V_1} c_{ij_1}x_{ij_1} + \sum_{i\in I}\sum_{r=1}^{k-1}\sum_{\{a,b\}\in E_r} c_{ab}z^r_{iab} - \sum_{r=1}^{k}\sum_{j_r\in V_r} f_{j_r}y_{j_r} - \sum_{r=1}^{k-1}\sum_{\{a,b\}\in E_r} d^r_{ab}w^r_{ab}$$

$$\text{s. t.} \quad \sum_{a\in V_1:\{i,a\}\in E_0} x_{ia} = 1 \qquad i \in I \tag{5.1}$$

$$\sum_{b\in V_2} z^1_{iab} = x_{ia} \qquad \{i,a\} \in E_0 \tag{5.2}$$

$$\sum_{b\in V_{r+1}:\{a,b\}\in E_r} z^r_{iab} = \sum_{b'\in V_{r-1}:\{b',a\}\in E_{r-1}} z^{r-1}_{ib'a} \qquad i\in I,\ a\in V_r,\ r=2,\cdots,k-1 \tag{5.3}$$

$$x_{ia} \le y_a \qquad \{i,a\}\in E_0 \tag{5.4}$$

$$\sum_{a\in V_{r-1}:\{a,b\}\in E_{r-1}} z^{r-1}_{iab} \le y_b \qquad i\in I,\ b\in V_r,\ r=2,\cdots,k \tag{5.5}$$

$$z^r_{iab} \le w^r_{ab} \qquad i\in I,\ \{a,b\}\in E_r,\ r=1,\cdots,k-1 \tag{5.6}$$

$$\sum_{j_r\in V_r} y_{j_r} \le p_r \qquad r=1,\cdots,k \tag{5.7}$$

$$x_{ia} \ge 0 \qquad \{i,a\}\in E_1 \tag{5.8}$$

$$z^r_{iab} \ge 0 \qquad i\in I,\quad \{a,b\}\in E_r,\quad r=1,\cdots k-1 \tag{5.9}$$

$$y_{j_r} \in \{0,1\} \qquad j_r\in V_r,\quad r=1,\cdots,k. \tag{5.10}$$

$$w^r_{ab} \in \{0,1\} \qquad \{a,b\}\in E_r,\quad r=1,\cdots,k-1. \tag{5.11}$$

Constraints (5.1) ensure that every customer is assigned to a first-level facility. The equalities (5.2) and (5.3) ensure the assignment of sequences of facilities for each customer. Constraints (5.4)–(5.6) are linking constraints between variables and (5.7) are the cardinality constraints. Constraints (5.8)–(5.11) are the standard nonegativity and integrality conditions on the decision variables. In this case, the variables x and z can be considered continuous without affecting the integer optimal solution. Moreover, in order to simplify the notation, whenever there is no ambiguity we will write $\sum_{b \in V_{r+1}}$ instead of $\sum_{b \in V_{r+1}:\{a,b\} \in E_r}$.

Some special cases of interest arise from the MUpLP-E. Clearly, when all fixed costs for the edges d^r_{ab} are zero, we obtain the MUpLP, which subsumes the *uncapacitated p-location problem* (UpLP) [45] when $k = 1$, which in turn, contains as special cases both the *uncapacitated facility location problem* (UFLP) [100] and the *p-median problem* (p-MP) [83]. Thus, multi-level extensions of the UFLP and the p-MP are also special cases of the MUpLP. Namely, the well-known *multi-level uncapacitated facility location problem* (MUFLP) [90] is obtained when all cardinality constraints are redundant, i.e. when $p_r = |V_r|$ for all r, and the *multi-level p-median problem* (MpMP) is obtained when all setup costs are set to zero, that is, $f_{j_r} = 0$. Other related formulation for the special case when $k = 2$, $p_r = |V_r|$ for all r, and $d \equiv 0$ was the one presented by [21] and later used by [70], denoted as TUFLP-C. Note that the following transformation of the ABF variables x_{ia} and z_{iab} into the TUFLP-C variables x_{bai} for $i \in I$, $a \in V_1$ and $b \in V_2$ (considering that the corresponding edges exists) is sufficient to prove the equivalence in the LP bounds of both formulations. That is, $x_{ia} = \sum_{b \in V_2} x_{bai}$ and $z_{iab} = x_{bai}$.

5.3 Benders Decomposition for MUpLP-E

Benders decomposition is a well-known partitioning method applicable to mixed integer programs [24]. It decomposes the original MILP problem into two simpler ones: an integer *master problem* and a linear *subproblem*. The main idea of the method is to reformulate the problem by projecting out a set of complicating variables to obtain a formulation with fewer variables but typically with a huge number of constraints.

These new constraints are usually referred to as Benders cuts and involve only the variables kept in the reduced problem, plus one additional continuous variable. Given that only a small subset of these constraints are usually active in an optimal solution, a natural relaxation is obtained by dropping most of them and generate them on the fly as needed.

The *standard* Benders decomposition algorithm is an iterative procedure in which at every iteration a relaxed integer master problem, containing only a small subset of Benders cuts, is optimally solved to obtain a valid dual bound. The linear subproblem is then solved to obtain a primal bound and to determine whether additional Benders cuts need to be incorporated to the master problem. With the addition of Benders cuts at each iteration, new tentative solutions are generated by the master problem, and new cuts are produced until the convergence of the bounds is attained, if an optimal solution exists.

Although the standard Benders decomposition has been successfully implemented to solve a variety of difficult optimization problems [see for instance, 41, 43, 144], for other problems the method is clearly outperformed by other decomposition strategies, such as Lagrangean relaxation and column generation. One of its major drawbacks is the need to solve an integer master problem at each iteration. To overcome this difficulty, *modern* implementations of Benders decomposition have considered the solution of the Benders reformulation with a standard branch-and-cut framework, in which Benders cuts are separated not only at integer solutions but also at fractional solutions at the nodes of a single enumeration tree [see, 8, 58, 61, 120, and references therein]. This results in the solution of a single integer program (the Benders reformulation) in order to obtain an optimal solution to the original problem. This is the approach we follow to develop an exact algorithm for MUpLP-E based on the following Benders reformulation of the ABF previously presented.

5.3.1 Benders Reformulation

Let Y denote the set of vectors (y, w) satisfying constraints (5.7), (5.10), and (5.11). For any fixed $(\bar{y}, \bar{w}) \in Y$, the *primal subproblem* (PS) in the space of the x and z variables is

$$\text{maximize} \sum_{i \in I} \sum_{j_1 \in V_1} c_{ij_1} x_{ij_1} + \sum_{i \in I} \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} c_{ab} z_{iab}^r$$

$$\text{subject to} \sum_{a \in V_1} x_{ia} = 1 \qquad i \in I \tag{5.12}$$

$$\sum_{b \in V_2} z_{iab}^1 = x_{ia} \qquad \{i, a\} \in E_0 \tag{5.13}$$

$$\sum_{b \in V_{r+1}} z_{iab}^r = \sum_{b' \in V_{r-1}} z_{ib'a}^{r-1} \qquad i \in I,\ a \in V_r,\ r = 2, \cdots, k-1 \tag{5.14}$$

$$x_{ia} \leq \bar{y}_a \qquad \{i, a\} \in E_0 \tag{5.15}$$

$$z_{iab}^r \leq \bar{w}_{ab}^r \qquad i \in I, \quad \{a, b\} \in E_r, \quad r = 1, \cdots, k-1 \tag{5.16}$$

$$\sum_{a \in V_{r-1}} z_{iab}^{r-1} \leq \bar{y}_b \qquad i \in I,\ b \in V_r,\ r = 2, \cdots, k \tag{5.17}$$

$$x_{ia} \geq 0 \qquad \{i, a\} \in E_0 \tag{5.18}$$

$$z_{iab}^r \geq 0 \qquad i \in I, \quad \{a, b\} \in E_r, \quad r = 1, \cdots k-1. \tag{5.19}$$

Note that PS can be decomposed into $|I|$ problems, one for each $i \in I$. Thus, we can construct the corresponding *dual subproblem* (DS_i) for each $i \in I$. Moreover, for every PS_i if we redefine the variables x_{ia} as z_{iia}^0, then constraints (5.13) can be seen as a special case of (5.14). Let λ, β_{ar}, α_{ia}^0, α_{ab}^r, and θ_{br} be the dual variables associated with constraints, (5.12), (5.14), (5.15), (5.17), and (5.16), respectively. For every $i \in I$, the DS_i can be stated as follows:

$$
\begin{aligned}
\text{minimize} \quad & \lambda + \sum_{a\in V_1} \bar{y}_a \alpha_{ia}^0 + \sum_{r=2}^{k} \sum_{b\in V_r} \bar{y}_b \theta_{br} + \sum_{r=1}^{k-1} \sum_{\{a,b\}\in E_r} \bar{w}_{ab}^r \alpha_{ab}^r \\
\text{subject to} \quad & \lambda - \beta_{a1} + \alpha_{ia}^0 \geq c_{ia} && \{i,a\} \in E_0 && (5.20)\\
& \beta_{ar} - \beta_{br+1} + \alpha_{ab}^r + \theta_{br+1} \geq c_{ab} && \{a,b\} \in E_r \quad r = 1, \cdots, k-2 && (5.21)\\
& \beta_{ak-1} + \alpha_{ab}^{k-1} + \theta_{bk} \geq c_{ab} && \{a,b\} \in E_{k-1} && (5.22)\\
& \alpha_{ab}^r \geq 0 && r = 0, \cdots, k-1, \ \{a,b\} \in E_r && (5.23)\\
& \theta_{br} \geq 0 && r = 2, \cdots, k, \quad b \in V_r. && (5.24)
\end{aligned}
$$

There exists at least one solution in the set of feasible solutions associated with DS_i. This is true because for an infeasible PS_i, the corresponding DS_i is either unbounded or infeasible. If the DS_i is infeasible, then the PS_i in the homogeneous form is unbounded, which is a contradiction since in this case the homogeneous form of the PS_i yields a finite value. Thus, the DS_i corresponding to an infeasible PS_i is unbounded. This is actually a special case of a classical result of network flow problems [78]. We will discuss the connection with network flow problems in the following sections. Therefore, we use the representation of each polyhedron associated with DS_i in terms of its extreme points and extreme rays to determine whether PS is infeasible or feasible and bounded, we denote them EP_i and ED_i, respectively.

If, for a given $(y, w) \in Y$, there exists at least one $i \in I$ and one extreme ray $(\lambda, \alpha, \theta) \in ER_i$ for which

$$0 > \lambda_i + \sum_{a\in V_1} y_a \alpha_{ia}^0 + \sum_{r=2}^{k} \sum_{b\in V_r} y_b \theta_{br} + \sum_{r=1}^{k-1} \sum_{\{a,b\}\in E_r} w_{ab}^r \alpha_{ab}^r,$$

then DS_i is unbounded and PS_i is infeasible. However, if

$$0 \leq \lambda_i + \sum_{a \in V_1} y_a \alpha^0_{ia} + \sum_{r=2}^{k} \sum_{b \in V_r} y_b \theta_{br} + \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} w^r_{ab} \alpha^r_{ab},$$

for each $i \in I$ and each extreme ray $(\lambda, \alpha, \theta) \in ER_i$, then all DS_i are bounded and the PS is feasible. The optimal value of each DS_i is then equal to

$$\min_{(\lambda,\alpha,\theta) \in EP_i} \lambda_i + \sum_{a \in V_1} y_a \alpha^0_{ia} + \sum_{r=2}^{k} \sum_{b \in V_r} y_b \theta_{br} + \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} w^r_{ab} \alpha^r_{ab}.$$

Introducing extra continuous variables η_i for the overall profit of each customer $i \in I$, the *Benders reformulation* (BR) associated with ABF is

$$\text{maximize} \quad \sum_{i \in I} \eta_i - \sum_{r=1}^{k} \sum_{j_r \in V_r} f_{j_r} y_{j_r} - \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} d^r_{ab} w^r_{ab}$$

subject to

$$\eta_i \leq \lambda_i + \sum_{a \in V_1} y_a \alpha^0_{ia} + \sum_{r=2}^{k} \sum_{b \in V_r} y_b \theta_{br} + \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} w^r_{ab} \alpha^r_{ab} \qquad i \in I, (\lambda, \alpha, \theta) \in EP_i \tag{5.25}$$

$$0 \leq \lambda_i + \sum_{a \in V_1} y_a \alpha^0_{ia} + \sum_{r=2}^{k} \sum_{b \in V_r} y_b \theta_{br} + \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} w^r_{ab} \alpha^r_{ab} \qquad i \in I, (\lambda, \alpha, \theta) \in ER_i \tag{5.26}$$

$$(y, w) \in Y.$$

We note that BR contains only the binary design variables (y, w) and $|I|$ additional continuous variables. Constraints (5.25) and (5.26) are the so-called Benders *optimality* and *feasibility* cuts, respectively.

5.4 An Exact Algorithm for MUpLP-E

In this section we present an exact branch-and-cut (B&C) algorithm based on BR to solve MUpLP-E. We use several strategies to enhance the algorithm and speed up its convergence. In particular, *(i)* we exploit the structure of the subproblem to identify optimality cuts by efficiently solving several network flow problems, *(ii)* we generate Pareto-optimal cuts using a variable core-point selection strategy, and *(iii)* we introduce valid inequalities that help reduce the number of cuts added and improve the overall performance of the algorithm.

5.4.1 Network Flow Structure

One important characteristic of well-known MILP formulations of single-level FLPs is that the subproblems of a typical Benders decomposition algorithm possess a network flow structure [111, 112]. This property conveniently extends to ABF and thus, can also be exploited to efficiently solve the subproblems.

First note that for every (undirected) edge $\{a, b\} \in E_r$, with $a \in V_r$ and $b \in V_{r+1}$, we can work instead with its associated (directed) arc (a, b), as if there was a flow to be sent from each customer to the k-th level of facilities, without affecting the optimal solution. In what follows, we refer to arcs $(a, b) \in V_r \times V_{r+1}$ and edges $\{a, b\} \in E_r$ indistinctly. For every $i \in I$, the corresponding constraints of PS_i are thus similar to those of a minimum cost flow problem. Equations (5.12)–(5.14) are the well-known flow conservation constraints, while (5.15) and (5.16) enforce the arc capacity constraints. In this case, inequalities (5.17) impose the capacity limits of the vertices in the network. Finally, we add a dummy vertex D to the network such that for every vertex $a \in V_k$ the arcs (a, D) exist and they have arc capacities equal to one. Also, we impose profits c and costs d equal to zero for these edges and a flow demand of -1 for vertex D. Therefore, we can consider each PS_i as a *minimum cost flow problem* with negative costs c_{ab} on the arcs (a, b) in which we require a unit of flow to be sent from customer i and received by the dummy vertex. That is, we include the extra redundant constraint $\sum_{a \in V_k} z^k_{iaD} = -1$ in PS$_i$ with the corresponding variables. Moreover, it is well known that vertex capacity constraints can be viewed as arc capacity constraints after a simple transformation on the graph [see Chapter 2 of reference 11]. Thus, we replace every vertex $a \in V$ that has a capacity, with two copies a' and a'', and we link them by an arc (a', a'') which has the corresponding vertex capacity and a profit $c_{a'a''}$ equal to zero. Vertex a' receives all the inflow associated with a, and a'' sends all the vertex outflow.

In the transformed graph $G^T = (V^T \cup I, E^T)$ we have $|V^T| = 2|V| - |V_1| + 1$ and $|E^T| = |E| + |V \backslash V_1| + |V_k|$. In what follows, we refer to the PS$_i$ with these modifications as PS$_i$-N. As before, for each $i \in I$ we can obtain the dual of PS$_i$-N. In this case, we have one dual variable for every vertex pair in the transformed graph which we denote as β^1_{ar} and β^2_{ar} for each $a \in V_r$, and $r = 2, \cdots, k$, and β^1_{a1} for $a \in V_1$, since the vertices from the first level are not duplicated. The dual variables corresponding to the customer and the dummy vertex are denoted by λ and λ_D, respectively. We also have one dual variable for every arc, that is, α^r_{ab} for those arcs in the original network, and θ_{br} for those between the duplicated vertices. Then, for every $i \in I$ the corresponding DS$_i$-N is

$$\begin{aligned}
\text{minimize} \quad & \lambda - \lambda_D + \sum_{a \in V_1} \bar{y}_{a_1} \alpha^0_{ia} + \sum_{r=2}^{k} \sum_{b \in V_r} \bar{y}_b \theta_{br} + \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} \bar{w}^r_{ab} \alpha^r_{ab} + \sum_{a \in V_k} \bar{y}_a \alpha^k_{aD} \\
\text{subject to} \quad & \lambda - \beta^1_{a1} + \alpha^0_{ia} \geq c_{ia} \qquad \{i, a\} \in E_0 \qquad (5.27) \\
& \beta^2_{ar} - \beta^1_{br+1} + \alpha^r_{ab} \geq c_{ab} \qquad \{a, b\} \in E_r, \quad r = 1, \cdots, k-1 \qquad (5.28) \\
& \beta^1_{br} - \beta^2_{br} + \theta_{br} \geq 0 \qquad b \in V_r,\ r = 2, \cdots, k \qquad (5.29) \\
& \beta^2_{ak} - \lambda_D + \alpha^k_{aD} \geq 0 \qquad a \in V_k \qquad (5.30) \\
& \alpha^r_{ab} \geq 0 \qquad \{a, b\} \in E_r, \quad r = 0, \cdots, k-1 \qquad (5.31) \\
& \theta_{br} \geq 0 \qquad b \in V_r, \quad r = 2, \cdots, k. \qquad (5.32)
\end{aligned}$$

As before, feasibility and optimality cuts can be generated using the set of extreme points and extreme directions of the set of feasible solutions of DS$_i$-N. However, as with single-level FLPs, the subproblems are solved in the primal space using a specialized algorithm for network flow problems such as the network simplex algorithm. These are already well implemented in general purpose solvers which also provide the values of the dual variables associated with the vertices of the network G^T (i.e. $\lambda, \lambda_D, \beta$). In order to obtain the rest of the dual variable values α and θ, for each edge $\{a, b\} \in E^T$, we consider the formula $\max\{0, c_{ab} - \beta_b + \beta_a\}$ [see Chapter 9, 11], where β_b and β_a are the dual variables associated with the vertices a and b in the transformed network.

5.4.2 Pareto-optimal Cuts

It is well known that when Benders decomposition is applied to network design and facility location problems, the primal subproblem is typically degenerate. That is, there are several possibilities for the selection of a Benders cut given a solution to the master problem. Magnanti and Wong [109] proposed a procedure for obtaining

Pareto-optimal cuts, that is, cuts that are not dominated by any other cut. In this section we refer to the primal and dual subproblems in the network flow problem form (PS-N and DS-N), although the same procedure can be also applied to the original PS and DS.

Let $(\hat{y}, \hat{w})$ be a *core point* of the set $Y = \{(y, w) : (5.7), (5.10), \text{and } (5.11)\}$, that is, a point in the relative interior of its convex hull. For a given $(\bar{y}, \bar{w})$, let U_i denote the optimal value of DS$_i$-N. To identify a Pareto-optimal cut that separates the point $(\bar{y}, \bar{w})$ using the procedure described by Magnanti and Wong [109], we must solve the following *Pareto-optimal* subproblem (PO$_i$-N):

$$\text{minimize} \quad \lambda - \lambda_D + \sum_{a \in V_1} \hat{y}_a \alpha^0_{ia} + \sum_{r=2}^{k} \sum_{b \in V_r} \hat{y}_b \theta_{br} + \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} \hat{w}^r_{ab} \alpha^r_{ab} + \sum_{a \in V_k} \hat{y}_a \alpha^k_{aD}$$

$$\text{subject to} \quad (5.27) - (5.32)$$

$$\lambda - \lambda_D + \sum_{a \in V_1} \bar{y}_{a_1} \alpha^0_{ia} + \sum_{r=2}^{k} \sum_{b \in V_r} \bar{y}_b \theta_{br} + \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} \bar{w}^r_{ab} \alpha^r_{ab} + \sum_{a \in V_k} \bar{y}_a \alpha^k_{aD} = U_i. \tag{5.33}$$

We note that constraints (5.33) guarantee that the optimal solution to PO$_i$-N belongs to the set of optimal solutions to the original DS-N. That is, the separation problem of the Benders optimality cuts is optimally solved to identify either a Pareto-optimal inequality that is most violated by the point $(\bar{y}, \bar{w})$ or conclude that none exist.

The dual of PO$_i$-N, denoted as DPO$_i$-N, can be obtained by using the corresponding dual variable x_0 of the additional constraint (5.33) as follows:

$$\text{maximize} \quad \sum_{j_1 \in V_1} c_{ij_1} x_{ij_1} + \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} c_{ab} \; z^r_{iab} + U_i x_0$$

$$\text{subject to} \quad \sum_{a \in V_1} x_{ia} = 1 + x_0 \tag{5.34}$$

$$\sum_{b \in V_2} z^1_{iab} = x_{ia} \qquad \{i,a\} \in E_0 \tag{5.35}$$

$$\sum_{b \in V_{r+1}} z^r_{iab} = \sum_{b' \in V_{r-1}} z^{r-1}_{ib'a} \qquad a \in V_r, \; r = 2, \cdots, k-1 \tag{5.36}$$

$$\sum_{a \in V_k} z^k_{iaD} = -(1 + x_0) \tag{5.37}$$

$$x_{ia} \leq x_0 \bar{y}_a + \hat{y}_a \qquad \{i,a\} \in E_0 \tag{5.38}$$

$$\sum_{a \in V_{r-1}} z^{r-1}_{iab} \leq x_0 \bar{y}_b + \hat{y}_b \qquad b \in V_r, \; r = 2, \cdots, k \tag{5.39}$$

$$z^r_{iab} \leq x_0 \bar{w}^r_{ab} + \hat{w}^r_{ab} \qquad \{a,b\} \in E_r, \quad r = 1, \cdots, k-1 \tag{5.40}$$

$$x_{ia} \geq 0 \qquad \{i,a\} \in E_0 \tag{5.41}$$

$$z^r_{iab} \geq 0 \qquad \{a,b\} \in E_r, \quad r = 1, \cdots k-1 \tag{5.42}$$

$$z^k_{iaD} \geq 0 \qquad a \in V_k \tag{5.43}$$

$$x_0 \geq 0. \tag{5.44}$$

Even though x_0 is unrestricted in sign, we note that the non-negativity conditions (5.44) can be added given that the equality sign of (5.33) can be modified to a less than or equal sign without affecting the optimal solution. This modification is particularly useful to mitigate numerical stability issues that arise when solving PO$_i$-N.

Given that x_0 affects the right-hand side of flow conservation and capacity constraints, this problem can be seen as a *parametric minimum cost flow problem.* However, Magnanti et al. [112] show how this type of problems can be solved by only one minimum cost flow problem for each customer when setting $x_0 \geq \sum_{r=1}^{k-1} \sum_{\{a,b\} \in E_r} \hat{w}^r_{ab} +$

$\sum_{r=1}^{k} \sum_{j_r \in V_r} \hat{y}_{j_r}$. Therefore, the only differences with respect to the original PS$_i$-N are the amount of flow to be sent and received $(1 + x_0)$, as well as the arc and node capacities, which now depend on the core point and on x_0. An additional benefit of a priori fixing the value of x_0 is that there is no need to actually solve PS$_i$-N or DS$_i$-N to obtain U_i.

A slightly different procedure for generating Pareto-optimal cuts is given in [139]. It relies on two interesting observations about the original procedure proposed in [109]. First, Papadakos [139] shows that there is no need to add the extra constraint (5.33) to the Pareto-optimal subproblem to generate a Pareto-optimal cut and second, the requirement on $(\hat{y}, \hat{w})$ of being a core point may also be relaxed. With respect to the first point, the author argues that when the PS (and the DPO) is chosen instead of the DS (and the PO) to obtain the corresponding coefficients of the Benders cuts, typically because they can be handled more efficiently, and a ϵ-optimal solution to the PS is obtained, the DPO then becomes numerically unbounded due to the additional constraint (5.33). As for the second point, the author points out the practical difficulty of finding valid core points for some problems and suggests the use of a different class of points, referred to as Magnanti-Wong (MW) points, in order to obtain Pareto-optimal cuts. For special cases, Papadakos [139] provide sufficient conditions for a point to be an MW point. Moreover, he shows that any strict convex combination of an MW point and a point in Y is also an MW point. That is, if $(\hat{y}, \hat{w})$ is an MW point and $(\bar{y}, \bar{w}) \in Y$, for any $\phi \in \mathbb{R}$ such that $0 < \phi < 1$,

$$(\hat{y}, \hat{w})' = \phi(\hat{y}, \hat{w}) + (1 - \phi)(\bar{y}, \bar{w}), \tag{5.45}$$

is also an MW point. This expression can be used to generate a sequence of MW points to construct different Pareto-optimal cuts by simply using as $(\bar{y}, \bar{w})$ the current solution of the restricted master problem and as $(\hat{y}, \hat{w})$ the MW point considered in the previous iteration of the Benders algorithm. The modified procedure presented in [139] relies on the solution of a simplified version of the Pareto-optimal subproblem PO_i-N that does not include constraint (5.33) and the use of an MW point updated using (5.45) as explained above.

On one hand, this procedure has the potential benefit of avoiding the solution of the original DP to obtain the value U_i used in constraint (5.33) before solving the Pareto-optimal subproblem. On the other hand, given that constraint (5.33) is relaxed from the model, it does not guarantee that the generated Pareto-optimal cut will be even violated by the current solution to the master problem. The convergence is thus guaranteed by limiting the number of possible updates to the MW point in which at the last iteration, a classical Benders cut is generated.

In Section 5.5, we compare the impact of using Magnanti-Wong and Papadakos procedures to generate Pareto-optimal cuts for MUpLP-E. In both procedures, we use a dynamic core point (or MW point) selection strategy based on (5.45). We also analyze the benefit of using such a strategy versus a static one when using the Magnanti-Wong procedure.

5.4.3 Valid Inequalities

We recall that the set Y consist of the cardinality constraints on the number of facilities at each level as well as of the integrality conditions on the y, w variables. However, note that there are no explicit linking constraints that ensure that open links are associated with open facilities and vise versa. When only few Benders cuts have been added to the master problem, there may not be enough information yet to cause the selection of a set of facilities and links such that these define a connected network for each customer. Therefore, the PS will tend to be infeasible and several feasibility cuts will have to be added before generating any optimality cut. To overcome this difficulty, we introduce the following structural constraints that provide additional information to the BR:

$$\sum_{b \in V_{r+1}:\{a,b\} \in E_r} w^r_{ab} \geq y_a \qquad a \in V_r,\ r = 1, \cdots, k-1 \tag{5.46}$$

$$w^r_{ab} \leq y_a \qquad \{a,b\} \in E_r, \quad r = 1, \cdots, k-1 \tag{5.47}$$

$$w^r_{ab} \leq y_b \qquad \{a,b\} \in E_r, \quad r = 1, \cdots, k-1. \tag{5.48}$$

Constraints (5.46) ensure that if a facility is opened at node a in level r, then there must be at least one edge incident to a. Constraints (5.47)–(5.48) allow the activation of only those edges that are incident to open facilities. These constraints, although redundant for the BR, have shown to be useful in reducing the number of feasibility cuts as well as the overall CPU time consumed by the test instances. Thus, in our B&C algorithm we consider a slightly strengthened BF with $Y = \{(y, w) : (5.7), (5.10), (5.11) \text{ and } (5.46) - (5.48) \text{ are satisfied}\}$.

Given that in any feasible solution of the ABF, for every customer there is at least one path of opened edges with its corresponding facilities going from customer $i \in I$ to an opened facility to level k, then the following constraints are valid for ABF.

For each $i \in I$ and every subset $S \subseteq V \cup \{i\}$ with $i \in S$,

$$\sum_{\{a,b\} \in \delta(S)} w^r_{ab} \geq 1,$$

where $\delta(S) = \{\{a,b\} \in E : a \in S \text{ and } b \notin S\}$. These constraints are the so-called *cut-set* inequalities and provide alternative feasibility conditions for the BR. That is, when PS_i is infeasible (i.e. a feasibility Benders cut needs to be added) we can find the cut-set of minimum value and include it to the BR in order to cut off that solution. Moreover, note that these inequalities can be extended for the case in which we solve each PS_i as a network flow problem (PS_i-N). Simply, using the transformed graph G^T, we can think of the variables y_a as $w^r_{a'a''}$ where a' and a'' are the duplicated vertices for each $a \in V \backslash V_1$. Thus, we now have to select a set S from $V^T \cup \{i\}$ instead of $V \cup \{i\}$, and then for every $i \in I$ and $\delta(S) \subseteq E^T$, the cut-set inequalities can be restated as

$$\sum_{\{a,b\} \in \delta(S)} (w^r_{ab} + y_a) \geq 1, \tag{5.49}$$

where $\delta(S) = \{\{a, b\} \in E^T : a \in S \text{ and } b \notin S\}$. These inequalities guarantee that there is at least one path from each customer to the dummy vertex D, where a unit of flow can be sent. Therefore, constraints (5.49) define sufficient conditions for the feasibility of PS. Interestingly, every cut-set inequality can actually be mapped to an extreme ray of the DS-N (or DS) as shown in following proposition.

Proposition 5.1. *Every cut-set inequality of the form* (5.49) *corresponds to a Benders feasibility cut for DS-N.*

Proof. For each $i \in I$ and $S \subseteq V^T \cup \{i\}$, where $S \neq \emptyset$ and $i \in S$, consider the following values for the dual variables. We set to one the variables associated with a vertex in the graph G^T, that is, λ, λ_D and β, only if the corresponding vertex is not in the set S, and to zero otherwise. In this case, we always have $\lambda = 0$. For the dual variables associated with edges $\{a, b\} \in E^T$ (i.e. α and θ), we set them to one only if the edge $\{a, b\} \in \delta(S)$, or equivalently if $a \in S$ and $b \notin S$. Recall that the edges defined between duplicated vertices $\{a', a''\}$ are associated with the variables θ. Then, it is not difficult to verify that for every subset S, using the above definition of the dual variables, constraints (5.27)–(5.32) are satisfied in the homogeneous form. This means that such a definition of the dual variables corresponds to an extreme ray of the associated polyhedral set and thus defines a Benders feasibility cut for DS-N.

□

An important benefit of this alternative set of cut-set inequalities that guarantees the feasibility of PS is that they can be efficiently separated. To this end, we have implemented the Edmonds-Karp algorithm [54] to compute the maximum flow between customer $i \in I$ and the dummy vertex. If the maximum flow does not satisfy the requirement, then we must obtain the values for a feasibility cut which can be found with the minimum $i-D$ cut in the network using the same algorithm. Another important observation is that when working with k-partite complete graphs, the networks for each $i \in I$ are identical. This means that the minimum $i-D$ cut will be the same for every i. Moreover, the extreme directions will be the same for each DS_i-N since the only change from one DS to another is the value of c_{ia} in the first set of constraints. Therefore, when the graph is k-partitive complete and we need to add feasibility cuts for a given solution, we only have to find one cut instead of $|I|$.

In Section 5.5 we compare the computational performance of using standard feasibility cuts and cut-set inequalities. These experiments provide some insights into another interesting non-trivial open question which is to determine whether every feasibility cut actually corresponds to a cut-set inequality.

5.5 Computational Experiments

We have conducted an extensive computational study in order to assess the empirical performance of the different variants of Benders decomposition described in Sections 5.3 and 5.4. All versions of the algorithm were coded in C and run on an Intel Xeon E5 2687W V3 processor at 3.10 GHz under Linux environment. The algorithms were implemented using the callable library of CPLEX 12.6.3 with its default settings using only one thread. As mentioned, all variants of the Benders algorithm have been embedded within a B&C framework. For their implementation we use the *lazycutcallback* and *usercutcallback* functions of CPLEX. That is, we considered integer and fractional solutions to generate cuts. From now on, we refer to the above functions as LAZYC and USERC to simplify the notation. Also, for presentation purposes, we only present summarized results of the experiments. The interested reader is referred to the Online Appendix for the detailed results.

5.5.1 Benchmark Instances

We have performed our experiments on 265 instances. Ortiz-Astorquiza et al. [136] previously presented 73 instances for the MUpLP ($d = 0$). For this study, we include 12 more which were generated following the same structure but with larger sets of customers and potential facilities and we include set-up costs for the edges (i.e. $d > 0$) in all 85 instances. We refer to this set of instances as CAP/RAND. We have also tested the algorithms on two sets of 90 instances each, which were used in [70] for the case where $k = 2$ and $d = 0$. Finally, we have tested the case in which we include set-up costs on the edges for these instances. These sets of instances are referred to as GAP and LGAP (Large GAP), respectively.

Some of the CAP/RAND instances were transformed from the *capa*, *capb* and *capc* instances from the OR-Library [23] for the UFLP to their multi-level versions. Out of the 85 instances, 21 correspond to the type CAP where each one has 1,000 customers and 100 potential facilities. The setup costs for opening facilities were modified from the single level version in order to be dependent on its level and also scaled down by a factor of 1,000. The CAP instances have variants with $k = 2$ and $k = 3$ and for each multi-level version instance, three values of $p = (p_1, \cdots, p_k)$ were selected: a small one, a medium one and one with the same values of the potential facilities configuration, that is, making the cardinality constraints redundant (as in the MUFLP). The remaining 64 RAND instances were randomly generated and range between 500 and 3,000 customers, between 100 and 250 potential facilities, and between two and four levels. The $c_{j_r j_{r+1}}$ values go from 25 to 125. For these instances we considered four different values of p. Moreover, the original CAP/RAND instances were modified in order to have set-up costs on the edges d^r_{ab}. For every edge $\{a,b\} \in E_r$, we take $[(f_{a_r} + f_{b_{r+1}})/2] \times 0.2 + \mu$, where μ is a pseudo-random number between 10 and 100.

The procedure used to generate GAP instances for the two-level version from hard instances for the UFLP [93] was initially presented by Landete and Marín [103] who converted instances with 100 customers and 100 facilities into two-level $50 \times 50 \times 50$ instances. The same idea was applied to the set LGAP with 150×150 to $75 \times 75 \times 75$. In the original sets GAP and LGAP the set-up cost for opening edges is zero. However, it is important to mention that the underlying graphs are sparse, that is, not all the edges are available in order to select the sequence of facilities for each customer. This applies to the edges between first level and customers (E_0). In this case we used the same formula as for CAP/RAND instances for adding the set-up costs on the edges of E_1 for these sets of instances and we set $p_r = |V_r|$ for $r = 1, 2$.

5.5.2 Setting up Parameters

In this section we describe some of the features that we consider to play an important role in the tuning of different versions of the Benders decomposition algorithm. Based on preliminary experiments we found that the algorithm is relatively unstable in the sense that variations of certain parameter values give rise to a different pool of cuts, which affects the performance of the algorithm. Thus, we empirically defined tolerance values and we assessed the impact on the changes of other parameters described in this section.

First, as mentioned in Section 5.4, due to numerical errors, checking the feasibility of $(\bar{y}, \bar{w})$ in PS$_i$-N is not always sufficient since DPO$_i$-N may yield an infeasible solution. To overcome this problem we first tried changing the optimality tolerance of the solver, and if the problem persists, we either reinitialized the core point or found the corresponding values of the dual variables using the standard Benders optimality cut for that particular solution.

Moreover, in terms of tuning parameters we considered the following values in order to filter some of the numerical errors and to avoid adding unnecessary cuts. We defined the values *TolMaxFlow*, *TolLAZYC* and *TolUSERC*. The *TolMaxFlow* value was set to 1E-6 and represents the tolerance on the maximum flow for the Edmonds-Karp algorithm implemented in order to determine whether a solution is primal feasible or not. The other two parameters correspond to the tolerances on the violation of Benders optimality cuts by the current solution. In particular, we set these values to 0.0001 and 0.1, respectively. Moreover, we noticed that in many cases it is preferable to branch in the enumeration tree instead of keep adding USERC cuts that are most of the time very similar to each other and do not improve the bound efficiently. Therefore, we have included a maximum number of USERC cuts at the root node and at every node, which we denote as $MaxUSERC_{root}$ and *MaxUSERC*, respectively. Similarly, we have set a maximum number of iterations per node in the B&C tree, except for the root node, that we call *MaxIterNode*. Finally, we also included a parameter denoted *Depth* which defines the frequency at which we solved the separation problem in the enumeration tree and therefore the frequency of adding Benders cuts. As its name indicates, this parameter is dependent on the depth of the B&C tree. We considered the following values for tuning the parameters:

- $MaxIterNode = \{2, 5\}$,
- $\{MaxUSERC_{root}, MaxUSERC\} = \{\{8|I|, 2|I|\}, \{12|I|, 4|I|\}, \{80|I|, 20|I|\}\}$,
- $Depth_{CAP/RAND} = \{5, 20\}$,
- $Depth_{GAP/LGAP} = \{25, 50\}$,
- $\phi = \{0.5, 0.7\}$.

Recall that ϕ is a real number between 0 and 1 in the convex combination equation (5.45) to update the core point, and note that the values of *Depth* depend on the type of instance. The reason for this is that the optimality gap for GAP/LGAP instances when exiting the root node is typically large (around 12 to 18%), while for each of the CAP/RAND instances it is less than 2%. Therefore, the number of nodes in the branching tree for a GAP/LGAP instance would be much larger than for the CAP/RAND instances.

It is well-known that considering the trade-off between solving the separation problem and branching in the enumeration tree is of critical importance. Adding Benders cuts too frequently could be rather counterproductive for those instances that have a large optimality gap after the root node. It is not difficult to extend this notion in a more general instance-independent implementation of the algorithm.

Testing all non-trivial configurations of parameters together with the many different variants of the Benders algorithm presented in Section 5.4 in the full set of instances would require an unreasonable computational effort. Thus, in order to identify the most promising version of the algorithm we have selected a subset of instances in which we tested the performance of the different versions. Accordingly, we randomly chose 16 instances out of the 174 instances (85 CAP/RAND and 89 feasible LGAP) of main interest. The remaining 74 feasible GAP instances were solved within five seconds for the ABF so we only chose one GAP instance for the preliminary tests. The obtained results showed that on average, for most variants of the algorithm the best configuration of parameters is $MaxIterNode = 2$, $\{MaxUSERC_{root}, MaxUSERC\} = \{80|I|, 20|I|\}$, $Depth_{CAP/RAND} = 20$, $Depth_{GAP/LGAP} = 50$ and $\phi = 0.7$. Additional details for the preliminary comparison of the different configuration of parameters on the 17 instances are presented in the Online Appendix.

5.5.3 Analysis of Algorithmic Enhancements

We now present computational results obtained from a subset of chosen instances to assess the impact of the proposed strategies to enhance the B&C algorithm. As mentioned before, we randomly selected nine instances from CAP/RAND, seven from LGAP and one from GAP. For these preliminary tests we imposed a time limit of 7,200 seconds (2 hours) and we used the parameters configuration described in Section 5.5.2. In all the tables the average values are computed on the instances solved by the solution algorithms in comparison. We use the notation n/m to indicate that n instances were solved out of m. We omit this information when all variants in comparison solve all instances to optimality. Also, it is important to mention that for the GAP/LGAP instances which are not originally defined on a k-partite complete graph, it seems that solving the BR using only the edges that are in the network outperforms an algorithm for which the instance is redefined to be k-partite complete. Moreover, when defining the instances on sparse graphs, 17 out 180 (16 GAP and one LGAP) are infeasible, which is consistent with the results of Gendron et al. [70].

We first evaluate the benefit of exploiting the network flow structure of the problem together with the two described procedures to generate Pareto-optimal cuts. In particular, we test the performance of our algorithm when solving the subproblems as network flow problem in the primal space or as a standard LP in the dual space. Also, we compare the Magnanti-Wong and Papadakos procedures to generate Pareto-optimal cuts. In Table 5.1 we summarize these comparisons. For all variants compared in this experiments, we consider the generation of standard Benders feasibility cuts and a dynamic core point selection strategy in which all components of the initial *approximate* core point are set to one, i.e., $y_{j_r} = 1$ and $w^r_{ab} = 1$. Although strictly speaking this point is not a core point of Y, it is a very simple starting point whenever the core point is updated using equation (5.45), which in turn approximates a valid core point of Y depending on the current solution $\bar{y}, \bar{w}$.

Table 5.1: Comparison for Pareto-optimal cuts generation and subproblems solution

	CAP/RAND				GAP/LGAP			
PO procedure	Papadakos		MW		Papadakos		MW	
Solver	Network	LP	Network	LP	Network	LP	Network	LP
Inst. Solved	4/9	9/9	9/9	9/9	8/8	8/8	8/8	8/8
Av. CPU time	-	283.9	7.7	133.7	97.9	58.4	22.5	39.7
Av. BB nodes	-	7.2	1.6	2.9	3328.0	2439.6	2600.0	3102.3
Av Feas. cuts	-	40.7	9.4	12.2	2309.1	1836.5	1528.1	2423.5
Av Opt. cuts	-	26509.7	9958.9	10251.6	1419.6	805.9	182.6	355.8
Av. Subp. time	-	277.7	4.0	130.1	1.0	6.1	0.2	2.8
Av. Feas. Time	-	0.2	0.1	0.1	0.0	0.0	0.0	0.0

In Table 5.1, we left blank those values for which the corresponding variant did not solve all instances tested within the time limit. Note that for both sets of instances, in almost all criteria used in the comparison (i.e CPU time, BB nodes, etc), applying the MW procedure when solving the subproblems as network flow problems appear to be the best variant. However, the Papadakos procedure works better when the usual LP solver of CPLEX is used instead. Moreover, this version provides the best average number of BB nodes for the GAP/LGAP instances among all algorithms compared.

Table 5.2 compares three options for the core point selection strategy. In particular, we tested two dynamic variants with a variable approximate core point and one static version with a fixed one during the entire procedure. For this comparison we apply the MW procedure to generate optimality cuts and we use the standard Benders feasibility cuts. For this case, we only work with the CAP/RAND subset because the fixed core point that we found is only applicable to instances in which the graph is k-partite complete. Thus, we consider two initial values for the (approximate) core point. One is where we set all variables to one as in the previous experiment. The second (actual) core point that we take is defined when the graph is k-partite complete. More precisely, we set $y_{j_r} = 1/|V_r|$ and $w^r_{ab} = 0.5/|V_r|$ when $p_r = 1$, and when $p_r > 1$ we take $y_{j_r} = 1.5/|V_r|$ and $w^r_{ab} = 1/|V_r|$ for all the respective values of r. It is not difficult to prove that this point is indeed a core point of Y.

Table 5.2: Core Point comparison on nine CAP/RAND instances

	Dynamic		Static
	all ones	core point	core point
Av. CPU time	7.7	10.1	14.1
Av. BB nodes	1.6	0.6	13.0
Av Feas. cuts	9.4	13.3	15.8
Av Opt. cuts	9958.9	9982.2	15227.8
Av. Subp. time	4.0	4.4	7.5
Av. Feas. Time	0.1	0.1	0.1

The results of Table 5.2 show that there is a reduction of almost 30% in the average CPU time when passing from a static core point strategy to a dynamic one where it is updated iteratively. Moreover, the selection of the point with all its values set to one seems to be the best choice for this set of instances.

We next assess the advantages of including in BR the valid inequalities (5.46)–(5.48). Table 5.3 compares two variants of the algorithm obtained by including constraints (5.46)–(5.48) in the Benders reformulation or not. The first variant, denoted by DB, corresponds to a standard BR which considers Benders feasibility cuts, solving the subproblems with the LP solver in the dual space and no procedure for generating Pareto-optimal cuts. The second variant, denoted by MW, uses the Magnanti-Wong procedure for generating Pareto-optimal cuts with a dynamic core point strategy, and Benders feasibility cuts.

Table 5.3: Comparison of variants including and excluding valid inequalities

	CAP/RAND				GAP/LGAP			
	Including (5.46)–(5.48)		Excluding (5.46)–(5.48)		Including (5.46)–(5.48)		Excluding (5.46)–(5.48)	
	DB	MW	DB	MW	DB	MW	DB	MW
Inst. Solved	9/9	9/9	1/9	9/9	8/8	8/8	2/8	3/8
Av. CPU time	369.8	7.7	-	125.8	82.9	22.5	-	-
Av. BB nodes	94.3	1.6	-	153.1	4099.6	2600.0	-	-
Av Feas. cuts	97.7	9.4	-	2484.0	2377.4	1528.1	-	-
Av Opt. cuts	49051.2	9958.9	-	62321.8	741.6	182.6	-	-
Av. Subp. time	188.5	4.0	-	18.8	1.8	0.2	-	-
Av. Feas. time	0.8	0.1	-	82.0	0.0	0.0	-	-

Note that inequalities (5.46)–(5.48) have a determinant impact on the overall performance of the algorithm, even for the straightforward implementation of Benders within a B&C. Thus, in all the following comparisons we consider (5.46)–(5.48) as part of the BR. Similar to the results obtained in the previous experiment, from Table 5.3 we observe once more that the combination of generating Pareto-optimal cuts and solving the subproblems as network flows makes an important difference when solving the test instances in comparison with a standard implementation of Benders in a B&C algorithm.

Another important variant that we have computationally tested is where cut-set inequalities are used instead of the standard Benders feasibility cuts. Table 5.4 compares the performance of the two considered classes of cuts. For this experiments, we use the Benders algorithm with the MW procedure and initialize all the values of the approximate core point to one at the first iteration.

Table 5.4: Feasibility cuts comparison on subsets of instances

	CAP/RAND		GAP/LGAP	
	feasibility cuts	cut-set inequalities	feasibility cuts	cut-set inequalities
Av. CPU time	7.7	20.3	22.5	18.2
Av. BB nodes	1.6	3.2	2600.0	2159.8
Av Feas. cuts	9.4	100.6	1528.1	366.0
Av Opt. cuts	9958.9	10966.4	182.6	223.6
Av. Subp. time	4.0	4.7	0.2	0.2
Av. Feas. time	0.1	0.0	0.0	0.0

All nine CAP/RAND and eight GAP/LGAP instances were solved to optimality within the time limit for the two variants. It seems that the algorithm with standard Benders feasibility cuts outperforms its counterpart with cut-set for the CAP/RAND instances, and the opposite happens for the GAP/LGAP instances.

5.5.4 Particular Cases and Comparison with Other Solution Procedures

In this section we present the results obtained when executing the most promising versions of the algorithm on the complete sets of instances and we compare their performance with those of the ABF and a direct implementation of Benders in a B&C algorithm. Moreover, we provide results for some special cases of the MUpLP-

E when fixing values of the input and we compare the best variant of the Benders algorithm that we introduce with other solution algorithms proposed before for these special cases.

From the previous section, we know that using the Magnanti-Wong procedure for generating Pareto-optimal cuts with a dynamic approximate core point together with solving the subproblems as network flow problems seems to be the most efficient variant of our B&C algorithm. Following the notation of the previous section we denote this variant as MW. For the CAP/RAND instances we only consider standard Benders feasibility cuts while for GAP/LGAP we also compare the performance of using the cut-set inequalities (5.49). Tables 5.5 and 5.6 summarize this comparison. For all variants we use the configuration of parameters as described in Section 5.5.2. We have included the corresponding results of the ABF and those of the ABF with the extra constraints (5.46)–(5.48) denoted ABF+E. In particular, we present the number of instances solved according to their type. For example, in Table 5.5, the row $k = 2$ provides the number of two-level instances solved to optimality (out of 40) for each implementation, whereas row $p_r = |V_r|$ provides the same information for those instances for which the cardinality constraints are redundant. Moreover, we include in the last six rows the average values of the total CPU time, number of BB nodes, number of total feasibility cuts added, number of total optimality cuts added, CPU time spent in the PS/DS and CPU time spent in the separation of feasibility cuts. We computed these values considering only the 55 solved instances within the time limit by all four methodologies compared. In this case we imposed a time limit of 86,400 seconds (one day). We note that the number of cuts added may not correspond to the final number of cuts in the model due to the automatic purges executed by CPLEX.

Benders decomposition appears to be an appropriate solution methodology for the MUpLP-E. In particular, we see that a tuned up straight-forward BR (DB) vari-

Table 5.5: Summary of results on 85 CAP/RAND instances

	ABF	ABF+E	DB	MW
$k = 2$	30/40	29/40	33/40	40/40
$k = 3$	19/29	17/29	28/29	29/29
$k = 4$	13/16	12/16	16/16	16/16
$\lvert I \rvert = 500$	20/20	20/20	20/20	20/20
$\lvert I \rvert = 1000$	31/33	28/33	33/33	33/33
$\lvert I \rvert = 1500$	7/16	6/16	15/16	16/16
$\lvert I \rvert = 2000$	2/8	2/8	5/8	8/8
$\lvert I \rvert = 3000$	2/8	2/8	4/8	8/8
CAP	21/21	21/21	21/21	21/21
$p_r = \lvert V_r \rvert$	23/23	23/23	20/23	23/23
total	62/85	58/85	77/85	85/85
Av. CPU time	13275.31	15432.36	288.27	4.75
Av. BB nodes	0.16	0.05	34.67	1.02
Av Feas. cuts	-	-	842.75	9.69
Av Opt. cuts	-	-	42404.25	13061.06
Av. Subp time	-	-	465.95	8.35
Av. Feas. time	-	-	17.82	0.16

ant already reduces the average CPU time from more than 13,000 to less than 300 seconds. Moreover, the algorithm with the enhancements described in Section 5.4 clearly outperforms the formulations and the DB approach. In comparison with the AFB we observe a reduction of the average CPU time of more than three orders of magnitude without a significant increase in the average number of BB nodes. Also, the MW implementation solved all 85 CAP/RAND instances within the time limit, whereas the ABF implementation only solved 62. Interestingly, in the MW version the maximum CPU time for solving an instance was 351 seconds. That is, every instance is solved within less than 400 seconds. On the other hand, in the ABF case, only 10 instances were solved within that time limit. It is worth highlighting that, although the redundant constraints (5.46)–(5.48) seem to be counterproductive in terms of total time for the ABF in the CAP/RAND instances, they are highly beneficial in reducing the CPU time when used in the BR for both sets of instances. The complete results are presented in the tables of the Online Appendix.

We note that for the best version of the algorithm, we also assessed the algorithmic performance using as an initial lower bound the one obtained with a modified greedy heuristic presented in [136] for the MUpLP-E. The modification on the heuristic simply consists on subtracting from the objective value the values for the set-up costs of the edges associated with the solution obtained with the greedy heuristic for the MUpLP. However, the results of these experiments are omitted from the tables because the improvement in CPU time and BB nodes is not considerable (less than 5%). Also, adding an initial set of Benders cuts obtained from a feasible solution generated with the same heuristic did not improve the performance of the best versions presented here.

Table 5.6 presents the summary of results for the modified GAP/LGAP instances. In this case we omit the number of instances solved per type because all methods were able to solve the 163 (feasible) instances within the time limit. Nevertheless, we observe a reduction of approximately 90% in the average CPU time, which seems to translate into a 25% increase in the average number of BB nodes when comparing the ABF with the MW algorithm with the cut-set inequalities approach. Also note that in the case of the GAP/LGAP instances in which the original graph is sparse, the cut-set inequalities seem to perform better. There is an important reduction in the number of optimality cuts, and correspondingly in the CPU time, when these inequalities are used to ensure feasibility.

Table 5.6: Summary of results on 163 GAP/LGAP instances

		ABF	ABF+E	DB		MW	
				standard	cut-set	standard	cut-set
GAP	Av. CPU time	4.1	2.3	2.1	1.3	1.6	1.0
	Av. BB nodes	228.8	83.8	308.4	199.3	283.1	188.9
GAPL	Av. CPU time	574.1	215.3	315.9	169.8	103.3	47.2
	Av. BB nodes	2822.9	1153.6	6081.4	5054.3	6042.9	4021.6
TOTAL	Av. CPU time	315.3	118.6	173.5	93.3	57.1	26.2
	Av. BB nodes	1645.2	667.9	3460.5	2850.2	3428.0	2281.6
	Av Feas. cuts	-	-	2397.3	413.7	1991.8	385.4
	Av Opt. cuts	-	-	681.0	635.5	209.4	183.4
	Av. Subp. time	-	-	1.5	1.6	0.2	0.2
	Av. Feas. Time	-	-	0.0	0.0	0.0	0.0

Tables 5.7 and 5.8 provide the results of the experiments when assessing the performance of the Benders algorithm described above (MW), for previously studied

cases of the MUpLP-E. First, in Table 5.7 we show the results for the GAP/LGAP instances when there are no set-up costs on the edges. Recall that for these instances we have $p_r = |V_r|$ for $r = 1, 2$ and $k = 2$.

Table 5.7: Summary of results on 163 GAP/LGAP instances with $d = 0$

		ABF	DB	MW	
				standard	cut-set
GAP	Av. CPU time	1.5	2.5	1.7	1.3
	Av. BB nodes	225.4	356.1	318.2	257.9
GAPL	Av. CPU time	231.8	257.2	49.0	43.5
	Av. BB nodes	3838.2	5967.0	5105.7	4743.3
TOTAL	Av. CPU time	127.2	141.5	27.2	24.6
	Av. BB nodes	2198.0	3419.7	2887.8	2823.3
	Av Feas. cuts	-	1871.6	284.6	422.4
	Av Opt. cuts	-	1061.6	1396.1	277.9
	Av. Subp. time	-	2.8	0.2	0.2
	Av. Feas. Time	-	0.0	0.0	0.0

In this case, we observe a reduction of almost 80% in the average CPU time between ABF and the MW variant of the algorithm. Moreover, the solution method proposed by [70] for these sets of instances shows an average CPU time over 1,100 seconds for LGAP instances, although the average number of BB nodes is less than 820. In comparison, the Benders algorithm yields an average CPU time of less than 44 seconds for the LGAP instances but almost 5,000 BB nodes on average. We also note that the authors compared the performance of the algorithm proposed with a slightly older version of CPLEX (12.6.1). However, for the GAP/LGAP instances, they present the results of CPLEX yielding a smaller average CPU time than the proposed algorithm.

Finally, Table 5.8 summarizes the results for three special cases of the MUpLP-E namely, the MUpLP, the MpMP and the MUFLP (see Section 5.2). The first row for each special case corresponds to the number of CAP/RAND instances solved within two hours of CPU time. The averages of time and BB nodes are computed based on the number of instances solved by all four methods in comparison. For this comparison we have included a new column named SF. In this column we provide the results obtained when using the Submodular Formulation (SF) introduced by for the

MUpLP. This formulation is embedded in a B&C framework and it was shown to outperform other well-known MILP for the case of the MpMP and to be competitive for the general MUpLP. It is important to mention that we have executed the experiments following the configuration parameters of the authors, which showed to be most beneficial for the SF. Thus, only for the SF, we turned off the CPLEX heuristics and changed the default purge parameter of the USERC to filter. We also included the bound obtained with the greedy heuristic.

Table 5.8: Special cases of MUpLP-E on 85 CAP/RAND instances

	ABF	DB	MW	SF
MUpLP	62/85	72/85	83/85	58/85
Av. CPU time	960.52	77.54	5.62	649.60
Av. BB nodes	0.37	17.65	7.80	6867.61
MpMP	62/85	77/85	85/85	76/85
Av. CPU time	1099.48	54.62	4.22	35.81
Av. BB nodes	0.05	1.62	10.39	93.18
MUFLP	23/23	20/23	23/23	12/23
Av. CPU time	314.65	551.18	9.34	-
Av. BB nodes	0.00	45.05	2.65	-

For Table 5.8 the number of solved instances by all methodologies is 51 and 61 for the MUpLP and the MpMP, respectively. For the MUFLP case we consider 20 solved instances although the SF was only able to solve 12 within the time limit of two hours. We note that the Benders algorithm proposed outperforms the other methods in comparison for all the special cases of the MUpLP-E on the CAP/RAND set of instances.

5.6 Conclusions

We studied a general class of multi-level uncapacitated p-location problems in which link activation decision between levels of facilities are considered. A sophisticated exact algorithm based on a Benders reformulation was presented to solve large-scale instances of the problem. The results of extensive computational experiments confirm the efficiency of our Benders decomposition algorithm. In particular, we note that for the MUpLP-E, the Magnanti-Wong procedure using a dynamic core selection strategy combined with a network solver approach for the subproblems appears to be the most efficient variant. Instances with up to 3,000 customers, 250 potential facilities, and four levels of hierarchy were solved to optimality. The best version of the Benders algorithm was able to solve each CAP/RAND instance within less than 360 seconds, reducing by more than three orders of magnitude the average CPU time from that obtained using the ABF. Also, some hard benchmark instances with large LP gaps were tested in the experiments. A 90% reduction in the average CPU time was obtained compared with the ABF solved by CPLEX. Moreover, our Benders algorithm outperformed state-of-the-art solution algorithms when solving special cases of interests of the MUpLP-E.

Appendix

Computational Experiments

We introduce notation with which we refer to the different variants of the Benders algorithm. We consider four main criteria to change between variants, namely: (*i*) Use of a method for guaranteeing Pareto-optimal cuts, (*ii*) solution of the subproblems, (*iii*) selection of feasibility cuts and (*iv*) updating of the core point. We note that for all the variants we include the valid inequalities (5.46)–(5.48) and the implementation of the algorithm is done via a B&C framework. For the first criterion we can select among a Direct Benders (DB) or Magnanti-Wong (MW) approaches, or we can choose to implement the Papadakos [139] (P) version in which we do not require an additional constraint in the PO but also leads to pareto-optimal cuts. Thus, there are three options in the first criterion namely, DB, MW and P. For the second criterion we can decide to solve them with an standard LP solver or with a more specialized network flow algorithm. We denote these two possibilities by Ls and Ns, respectively. In particular, we use a general purpose solver (CPLEX) which has already implemented these two options. Also, for the third criterion we can select between using cut-set inequalities and standard Benders feasibility cuts, which we denote by Cf and Bf, respectively. Finally, for the MW versions we can choose to keep a fixed core point throughout all iterations or we can update it as in equation (5.45). We call these two options fixed (F) and variable (V). Finally for the core point criterion, we write (o) when the initial approximate core point is set to all ones and (n) otherwise. Thus, an implementation using a Pareto-optimal cuts with standard Benders feasibility cuts in which we solve the subproblems with a network solver and that consider a variable core point for every subproblem, is denoted as MW-Bf-Ns-V. Note that for the case of DB we do not have core points and for the P version we always update the core point accordingly. In those cases where the criterion is obvious or does not exist we omit it from the variant name (i.e. DB-Bf-N).

Also, we tested different values for the configuration of parameters as discussed in Section 5.5.2. In Table 5.9 we present the list of configuration tests used to tune the different variants of the Benders algorithm. Note that in the row of Depth, we have only used 5 or 20 which corresponds to the values used for the CAP/RAND instances only. For the GAP/LGAP instances we used 25 and 50, respectively.

Table 5.9: Configuration tests

Parameter \ Test	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15	T16	T17	T18	T19	T20
ϕ	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.7	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5	0.5
MaxIterNode	2	2	2	2	5	5	5	5	5	5	2	2	2	2	5	5	5	5	5	5
MaxUSERC	8,2	8,2	80,20	80,20	8,2	8,2	80,20	80,20	12,4	12,4	8,2	8,2	80,20	80,20	8,2	8,2	80,20	80,20	12,4	12,4
Depth	5	20	5	20	5	20	5	20	5	20	5	20	5	20	5	20	5	20	5	20

We preliminary tested the variants of the algorithm along with the configurations of parameters in subsets of instances. We were able to determine that two of the most beneficial configuration for most of the variants were test 4 and 8. Figure 5.1 shows the average CPU time per instance for these tests. The dominance in terms of CPU time of the MW method with the network solver approach is evident in comparison with the other versions. However, note that the MW method is actually the only one that benefits from solving the subproblems with the network solver. For the other two cases it seems that the Benders cuts obtained by solving the DS with the LP solver yield a better performance. Thus, for the MW version the computational advantage of generating Pareto-optimal cuts increases when the subproblems are solved more efficiently.

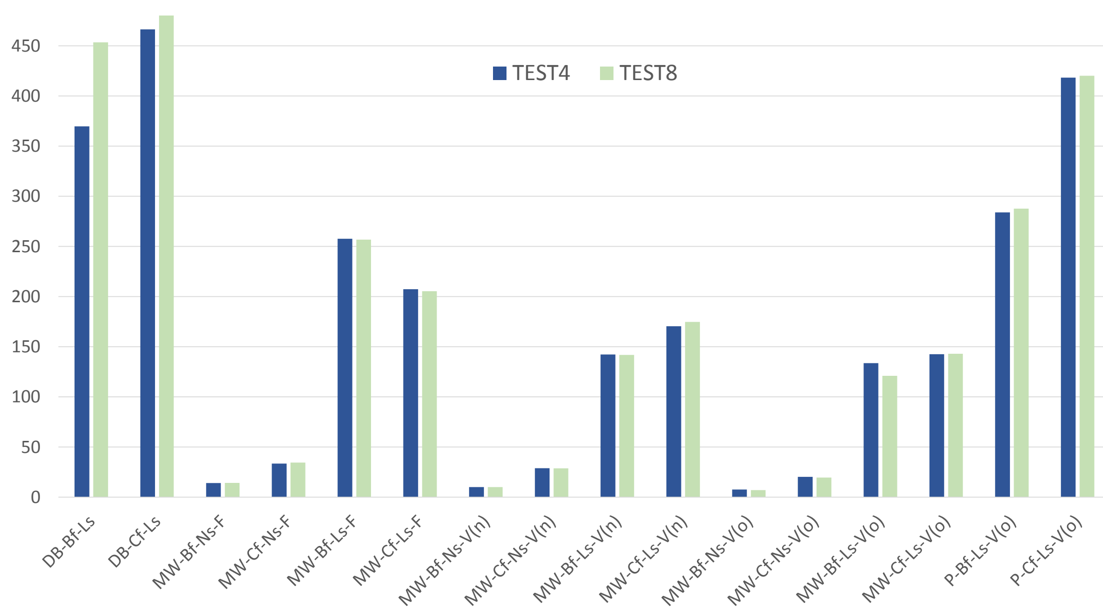


Figure 5.1: Average CPU times on nine CAP/RAND instances

Similarly, in Figure 5.2 we have summarized the results for a subset of the original GAP/LGAP instances, that is, with $d = 0$. In this case the dominance of one variant with a selection of one or two configuration tests is not as clear as before. Thus, we have included in the figure one more configuration that yields good results for certain variants. Although the best CPU times are again obtained for the MW versions with a network solver, now the cut-set inequalities seem to make a greater difference in comparison with the usual Benders feasibility cuts. This is also true for the P versions which are in this case more competitive than they were with the CAP/RAND instances.

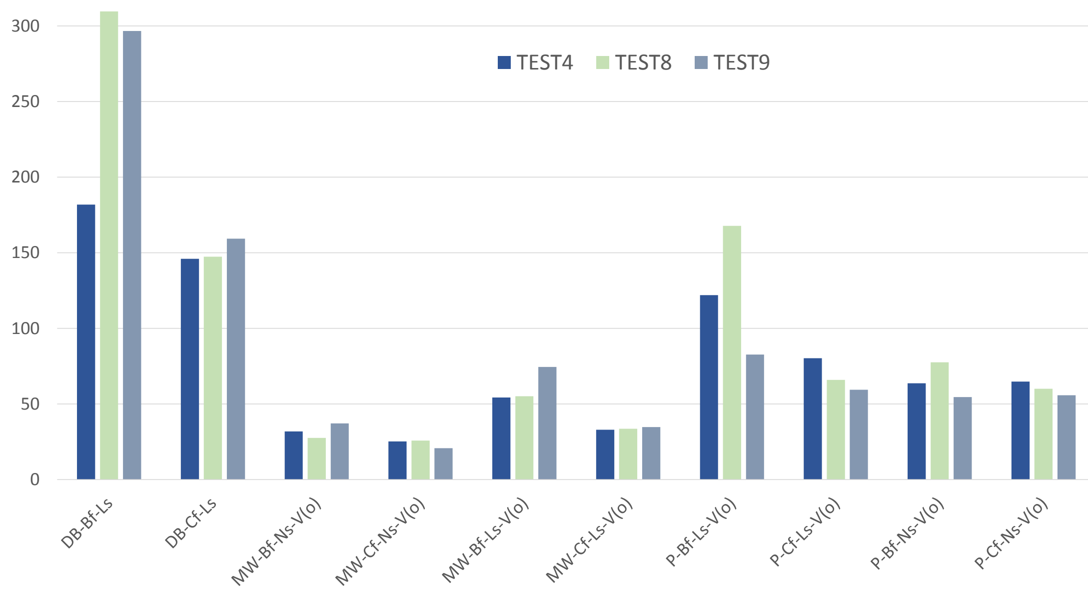


Figure 5.2: Average CPU times on eight GAP/LGAP instances with $d = 0$

The detailed computational results for the MUpLP-E are presented in Tables 5.10–5.14. Table 5.10 provides the results for the CAP/RAND instances where the first column describes the type of instance through its five subcolumns. The next 10 columns provide the CPU time in seconds needed to solve the instance and the number of nodes in the BB tree for all five methods. Whenever it is not possible to solve an instance within 86,400 seconds, we write TIME in the corresponding entry of the table. For the GAP/LGAP instances recall that for all of them $k = 2$, $p_r = |V_r|$ for $r = 1, 2$ and $|V_1| = |V_2|$. For the GAP instances $|V_1| = 50$ while for the LGAP case $|V_1| = 75$.

Table 5.10: Computational comparison on CAP/RAND instances

Instance					ABF		ABF+E		DB-Bf-Ls		MW-Bf-Ns-V(n)		MW-Bf-Ns-V(o)	
Type	Levels	$\|I\|$	Pot. Facilities	p	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes
capa	2	1000	70-30	2-1	4841.91	0	1437.97	0	92.23	4	3.02	0	1.10	0
capa	2	1000	70-30	3-2	4050.23	0	1314.10	0	123.46	7	4.08	3	2.03	2
capa	2	1000	70-30	70-30	4538.76	0	1448.20	0	56.85	0	4.33	3	1.32	0
capa	2	1000	50-50	2-1	897.54	0	672.63	0	263.82	9	4.61	0	2.93	0
capa	2	1000	50-50	3-2	772.30	0	779.33	0	107.75	14	3.47	0	3.03	0
capa	2	1000	50-50	50-50	909.35	0	1055.36	0	137.13	0	3.28	0	2.22	0
capb	2	1000	70-30	2-1	17122.61	0	8622.50	0	174.45	26	3.99	0	2.32	4
capb	2	1000	70-30	5-2	9456.22	0	3252.35	0	40.07	0	3.82	2	1.86	0
capb	2	1000	70-30	70-30	9439.91	0	3573.28	0	31.80	3	3.83	3	1.60	0
capc	2	1000	70-30	2-1	3850.60	0	7267.36	0	105.07	9	2.87	0	2.09	0
capc	2	1000	70-30	3-2	18754.39	0	24642.77	0	136.34	17	4.16	0	4.21	0
capc	2	1000	70-30	70-30	10328.28	0	6973.83	0	52.29	0	2.83	0	2.49	0
capa	3	1000	55-30-15	2-1-1	35332.49	0	25747.68	0	1096.40	28	3.76	0	1.69	0
capa	3	1000	55-30-15	3-2-1	25039.98	0	29976.38	0	573.85	7	2.95	0	2.97	0
capa	3	1000	55-30-15	55-30-15	33631.99	0	24204.30	0	337.96	67	4.12	0	3.00	4
capb	3	1000	60-30-10	2-1-1	21404.65	0	28086.94	0	1244.40	5	3.97	0	2.82	9
capb	3	1000	60-30-10	5-2-2	4648.88	0	7015.60	0	205.20	8	3.80	0	1.85	0
capb	3	1000	60-30-10	60-30-10	2472.56	0	5026.74	0	107.98	0	4.13	0	1.79	0
capc	3	1000	60-30-10	2-1-1	23897.40	0	13168.87	0	519.25	39	3.57	0	2.28	0
capc	3	1000	60-30-10	5-2-2	28942.66	0	12311.40	0	1718.33	70	6.13	0	3.64	6
capc	3	1000	60-30-10	60-30-10	20349.74	0	15726.67	0	602.35	332	6.70	0	4.08	8
RAND	4	500	40-30-20-10	2-2-1-1	7879.22	3	5326.62	0	143.61	57	11.00	5	4.29	2
RAND	4	500	40-30-20-10	3-3-3-3	9293.44	0	20665.58	0	29.82	0	2.78	0	2.49	0
RAND	4	500	40-30-20-10	10-5-3-2	12401.64	0	15997.58	0	26.13	0	2.31	0	1.97	0
RAND	4	500	40-30-20-10	40-30-20-10	343.87	0	601.11	0	23.54	0	1.35	0	2.86	0
RAND	3	500	50-30-20	4-2-1	42495.76	3	45443.68	0	84.05	64	3.87	0	3.34	0
RAND	3	500	50-30-20	5-5-2	26355.03	0	43806.22	0	56.22	0	3.18	0	3.34	0
RAND	3	500	50-30-20	11-9-6	8664.97	0	21689.68	0	23.44	0	1.88	0	1.63	0
RAND	3	500	50-30-20	50-30-20	228.20	0	355.76	0	21.55	0	1.97	0	2.04	0
RAND	2	500	70-30	3-1	720.53	0	785.58	0	17.97	0	2.62	0	1.53	0
RAND	2	500	70-30	5-4	7560.48	0	19114.86	0	30.55	6	2.28	0	1.89	0
RAND	2	500	70-30	10-2	11296.17	0	18085.45	0	37.89	0	1.64	0	1.34	0
RAND	2	500	70-30	70-30	79.13	0	79.35	0	19.42	73	2.29	4	1.93	5
RAND	2	500	50-50	2-1	2368.81	0	2917.60	0	33.15	0	2.56	0	2.77	0
RAND	2	500	50-50	5-5	153.24	0	143.54	0	17.56	0	1.95	0	2.19	0
RAND	2	500	50-50	11-9	63.62	0	84.15	0	10.51	3	1.38	0	1.52	0
RAND	2	500	50-50	50-50	63.33	0	78.67	0	10.94	0	1.52	0	0.96	0
RAND	4	500	60-30-20-10	4-2-1-1	56682.36	0	70814.31	0	164.45	0	8.91	0	4.39	0
RAND	4	500	60-30-20-10	8-2-2-1	45776.90	0	75466.96	0	91.90	0	5.04	0	5.20	0
RAND	4	500	60-30-20-10	15-10-5-2	16845.47	0	29530.55	0	48.14	0	2.82	0	2.45	0
RAND	4	500	60-30-20-10	60-30-20-10	452.01	0	986.65	0	66.57	0	2.13	0	2.46	0
RAND	4	1000	40-30-20-10	2-2-1-1	8099.98	0	12372.26	0	132.09	0	9.63	0	5.31	0
RAND	4	1000	40-30-20-10	3-3-3-3	34559.25	0	TIME	TIME	83.84	8	5.49	2	3.30	0
RAND	4	1000	40-30-20-10	10-5-3-2	30432.15	0	52271.96	0	48.81	0	3.77	0	2.88	0
RAND	4	1000	40-30-20-10	40-30-20-10	644.24	0	862.38	0	42.08	2	1.70	0	2.55	0
RAND	3	1000	50-30-20	4-2-1	TIME	TIME	TIME	TIME	77.64	3	6.85	0	5.35	0
RAND	3	1000	50-30-20	5-5-2	TIME	TIME	TIME	TIME	112.57	0	6.37	0	5.54	0
RAND	3	1000	50-30-20	11-9-6	49009.40	0	TIME	TIME	79.34	49	4.85	2	4.29	2
RAND	3	1000	50-30-20	50-30-20	339.28	0	450.09	0	62.83	13	5.03	3	4.26	3
RAND	2	1000	70-30	3-1	1949.38	0	2427.11	0	64.68	10	6.08	0	2.43	0
RAND	2	1000	70-30	6-6	20898.47	0	TIME	TIME	20.89	7	3.16	0	3.36	0
RAND	2	1000	70-30	10-2	44272.87	0	73042.52	0	84.69	12	4.98	0	2.45	0
RAND	2	1000	70-30	70-30	104.04	0	94.87	0	19.64	12	1.70	0	1.69	0
RAND	3	1500	50-30-20	4-2-1	TIME	TIME	TIME	TIME	148.87	2	13.89	0	8.52	0
RAND	3	1500	50-30-20	5-5-2	TIME	TIME	TIME	TIME	231.30	95	15.14	3	19.51	6
RAND	3	1500	50-30-20	11-9-6	67288.70	0	TIME	TIME	39.38	0	6.67	2	3.96	0
RAND	3	1500	50-30-20	50-30-20	351.08	0	458.99	0	33.34	4	2.29	0	3.14	2
RAND	2	1500	70-30	3-1	13683.47	0	15735.37	0	110.10	2	16.70	2	6.68	5
RAND	2	1500	70-30	5-4	TIME	TIME	TIME	TIME	94.15	0	10.67	0	6.02	0
RAND	2	1500	70-30	10-2	TIME	TIME	TIME	TIME	206.76	22	8.88	0	5.29	0
RAND	2	1500	70-30	70-30	169.87	0	172.13	0	24.08	8	2.24	0	2.46	0
RAND	3	1500	100-70-30	4-2-1	TIME	TIME	TIME	TIME	1810.70	65	158.73	34	62.00	11
RAND	3	1500	100-70-30	10-5-2	TIME	TIME	TIME	TIME	1027.68	48	53.98	0	36.62	0
RAND	3	1500	100-70-30	35-9-6	TIME	TIME	TIME	TIME	388.11	28	25.46	0	24.58	0
RAND	3	1500	100-70-30	100-70-30	2282.82	0	2401.48	0	376.17	8	14.96	0	14.77	0
RAND	2	1500	120-80	3-1	77333.64	3	68823.59	3	2768.97	224	57.23	9	48.70	6
RAND	2	1500	120-80	6-6	TIME	TIME	TIME	TIME	2221.19	394	29.15	0	29.08	0
RAND	2	1500	120-80	7-2	TIME	TIME	TIME	TIME	6047.57	789	42.34	0	33.61	0
RAND	2	1500	120-80	120-80	2152.11	0	2818.64	3	TIME	TIME	23.97	3	22.17	7
RAND	2	2000	120-80	4-1	TIME	TIME	TIME	TIME	6977.17	800	169.67	355	72.30	20
RAND	2	2000	120-80	7-2	TIME	TIME	TIME	TIME	TIME	TIME	57.63	0	51.33	0
RAND	2	2000	120-80	10-10	TIME	TIME	TIME	TIME	TIME	TIME	41.91	0	29.51	0
RAND	2	2000	120-80	120-80	1487.73	0	1399.18	0	TIME	TIME	14.45	0	20.60	2
RAND	4	2000	100-50-35-15	2-2-1-1	TIME	TIME	TIME	TIME	1965.08	15	115.68	0	58.19	0
RAND	4	2000	100-50-35-15	6-5-3-1	TIME	TIME	TIME	TIME	903.78	7	67.09	5	42.46	0
RAND	4	2000	100-50-35-15	12-7-5-3	TIME	TIME	TIME	TIME	471.23	5	41.90	0	30.18	0
RAND	4	2000	100-50-35-15	100-50-35-15	13596.62	0	17901.84	0	381.37	6	16.38	0	20.93	0
RAND	2	3000	150-100	5-2	TIME	TIME	TIME	TIME	TIME	TIME	361.30	0	351.43	5
RAND	2	3000	150-100	15-5	TIME	TIME	TIME	TIME	5711.92	561	173.54	2	151.98	0
RAND	2	3000	150-100	65-12	TIME	TIME	TIME	TIME	TIME	TIME	65.22	0	55.53	0
RAND	2	3000	150-100	150-100	4178.26	0	4071.47	0	TIME	TIME	41.71	0	29.65	0
RAND	3	3000	120-80-50	5-3-2	TIME	TIME	TIME	TIME	TIME	TIME	307.05	4	250.57	3
RAND	3	3000	120-80-50	15-8-3	TIME	TIME	TIME	TIME	12064.88	1261	458.95	282	169.44	17
RAND	3	3000	120-80-50	45-27-6	TIME	TIME	TIME	TIME	3232.42	139	102.60	0	65.91	2
RAND	3	3000	120-80-50	120-80-50	6475.84	0	7486.84	0	3021.45	758	35.12	0	41.02	0

Table 5.11: Computational comparison GAP instances

	ABF		ABF+E		DB-Bf-Ls		DB-Cf-Ls		MW-Bf-Ns-V(o)		MW-Cf-Ns-V(o)	
Instance	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes
333GapC	2.78	116	1.26	25	1.58	132	0.98	80	1.27	83	0.64	29
431GapB	0.82	33	0.65	19	1.14	134	0.90	116	0.67	48	0.66	70
432GapA	1.63	60	1.72	42	2.34	442	1.17	173	1.40	345	1.10	225
433GapC	19.45	589	9.16	264	4.35	826	3.68	583	5.54	1446	2.32	382
531GapB	2.04	122	1.49	69	1.53	185	0.92	67	1.01	120	0.66	132
532GapA	4.85	237	2.60	84	2.61	376	2.51	620	1.73	471	1.57	443
533GapC	4.66	184	2.79	78	2.43	318	1.32	180	2.12	330	1.07	170
632GapA	0.65	39	0.64	27	0.93	60	0.81	91	0.72	68	0.53	61
633GapC	3.15	287	1.26	64	1.43	125	1.48	267	0.95	106	0.75	101
732GapA	3.35	228	1.83	95	1.51	245	1.20	173	1.33	209	0.89	162
733GapC	17.29	1267	6.52	366	7.26	1837	2.30	657	4.18	1513	1.72	763
832GapA	5.37	450	2.22	155	2.27	512	1.57	251	1.46	439	1.01	308
833GapC	4.39	253	2.56	109	2.08	193	1.55	222	1.45	154	0.98	174
931GapB	2.23	258	1.41	117	1.41	112	0.82	119	0.91	70	0.70	124
932GapA	4.15	294	1.28	29	1.60	198	1.08	127	1.26	142	0.77	124
933GapC	2.07	160	1.72	119	1.08	194	1.19	203	0.81	204	0.75	181
1031GapB	3.65	275	1.22	67	1.49	147	1.37	237	0.98	91	0.73	122
1032GapA	0.68	19	1.03	18	0.65	19	0.62	21	0.63	19	0.39	14
1033GapC	1.24	48	0.59	15	0.95	41	0.82	39	0.57	11	0.54	31
1132GapA	1.94	118	1.61	61	2.07	194	0.93	58	1.39	135	0.91	155
1133GapC	5.78	319	3.48	130	3.38	675	1.44	170	2.09	331	1.25	275
1231GapB	1.51	54	1.13	23	2.21	197	0.88	42	1.17	111	0.74	56
1232GapA	1.88	120	0.95	37	1.53	202	0.91	101	1.08	133	0.61	46
1233GapC	3.19	85	3.16	83	3.33	743	2.22	296	2.20	506	1.22	140
1331GapB	4.72	535	2.39	233	2.29	456	1.27	246	1.40	275	0.95	288
1332GapA	2.92	173	2.33	73	2.28	209	1.50	143	1.65	273	0.86	111
1333GapC	10.63	717	4.79	255	7.17	1557	2.03	491	4.06	1229	1.43	395
1431GapB	3.00	312	1.67	113	1.51	333	1.52	511	1.34	442	0.95	273
1432GapA	1.97	89	1.39	45	1.94	270	1.14	199	1.40	229	0.81	211
1433GapC	13.31	512	7.25	178	6.12	942	3.11	567	4.35	1194	2.54	636
1532GapA	5.42	304	3.73	121	1.69	270	1.17	165	1.10	127	0.82	68
1533GapC	4.70	235	3.07	109	1.83	172	1.48	168	1.22	147	1.10	178
1632GapA	5.67	243	3.72	132	2.66	443	1.55	225	1.75	390	0.89	129
1633GapC	5.20	175	3.32	97	2.26	299	1.69	293	2.01	347	1.21	255
1731GapB	3.23	250	1.10	41	1.17	162	1.09	121	1.12	205	0.67	134
1733GapC	15.22	683	5.19	176	4.22	1031	2.18	438	2.47	530	1.60	317
1832GapA	2.70	231	1.39	61	0.92	120	0.96	116	0.73	117	0.74	107
1833GapC	5.11	296	2.96	116	2.13	321	1.22	287	1.50	455	0.84	227
1932GapA	1.14	57	0.81	27	0.82	48	0.65	21	0.70	68	0.47	18
1933GapC	1.31	91	0.70	30	1.29	112	1.02	130	1.12	219	0.85	143
2031GapB	1.49	124	0.93	43	1.24	111	0.70	72	1.07	107	0.62	76
2032GapA	1.82	60	1.14	9	1.07	55	1.08	56	0.69	20	0.74	62
2033GapC	7.16	454	4.30	182	2.57	515	1.64	372	1.88	527	1.23	290
2132GapA	3.69	200	2.50	97	1.73	237	2.10	311	1.50	219	1.31	295
2133GapC	2.65	231	1.78	86	1.68	130	1.04	120	0.94	88	0.66	92
2232GapA	4.92	349	3.32	147	2.32	308	2.06	351	1.55	308	1.49	382
2233GapC	2.26	62	2.07	55	1.79	153	1.53	139	1.45	170	1.10	151
2331GapB	0.78	38	0.74	24	0.73	36	0.69	34	0.57	52	0.48	33
2332GapA	2.53	135	2.26	99	1.71	198	1.21	196	1.22	233	1.00	238
2333GapC	9.25	345	4.25	118	2.70	357	1.96	357	2.15	359	1.31	215
2431GapB	2.25	204	1.30	77	1.45	183	1.07	190	0.94	156	0.74	177
2432GapA	3.28	200	1.55	25	2.33	224	1.34	131	1.35	112	1.10	201
2433GapC	4.72	185	3.39	109	2.69	290	1.47	255	1.92	335	1.10	191
2532GapA	2.93	184	2.33	71	1.94	228	1.33	173	1.58	267	1.10	162
2533GapC	2.83	114	1.22	13	1.81	150	1.27	110	1.77	167	1.18	236
2632GapA	2.88	77	1.88	33	2.12	322	1.43	137	1.35	162	1.50	311
2633GapC	4.92	600	1.82	125	1.72	303	1.23	306	1.32	279	0.89	265
2731GapB	0.67	11	0.69	17	0.97	53	0.62	42	0.50	19	0.63	93
2732GapA	2.64	109	2.24	53	1.70	245	1.43	158	1.41	189	0.97	204
2733GapC	4.43	146	2.66	81	2.05	355	1.22	164	2.03	419	0.95	160
2831GapB	1.29	107	0.88	25	0.95	39	1.07	93	0.88	55	0.56	52
2832GapA	2.03	60	1.68	23	1.93	182	0.89	40	1.33	52	0.80	62
2833GapC	8.45	379	3.77	130	2.48	384	1.62	261	1.79	266	1.64	371
2931GapB	1.67	85	1.04	25	1.37	85	1.22	124	1.10	66	0.79	63
2932GapA	1.97	98	1.22	32	1.99	172	1.01	43	1.39	124	0.74	56
2933GapC	2.82	133	2.00	61	1.42	101	1.05	94	1.36	150	0.97	139
3032GapA	3.17	186	1.87	74	2.35	249	1.10	136	1.73	198	1.01	211
3033GapC	3.64	136	2.70	45	2.86	462	1.36	258	3.08	804	1.47	363
3131GapB	3.44	242	1.97	94	1.53	149	1.33	173	1.13	158	0.96	163
3132GapA	2.41	127	1.23	46	1.67	195	1.16	107	0.97	87	0.86	90
3133GapC	8.31	283	6.19	127	5.74	816	2.22	326	3.78	702	1.96	301
3231GapB	0.83	49	0.43	21	1.17	65	0.89	65	0.94	66	0.77	102
3232GapA	6.38	561	1.93	87	2.06	342	1.39	284	1.54	338	1.08	296
3233GapC	3.56	142	2.68	45	2.06	274	0.85	57	2.00	365	0.88	99

Table 5.12: Computational comparison LGAP instances

	ABF		ABF+E		DB-Bf-Ls		DB-Cf-Ls		MW-Bf-Ns-V(o)		MW-Cf-Ns-V(o)	
Instance	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes
GapA0	683.10	4777	155.50	1273	74.57	4264	59.88	2750	29.70	4860	17.06	2592
GapA1	1016.04	6381	407.84	2793	217.89	9839	369.06	15285	76.27	7248	86.03	11433
GapA2	619.12	3383	129.00	834	54.69	2955	42.26	1698	30.61	5025	11.87	1404
GapA3	650.22	3454	179.94	1048	81.02	3485	176.44	13782	35.85	4363	43.21	5196
GapA4	105.62	745	64.96	564	21.09	968	10.19	497	5.53	509	5.58	544
GapA5	1319.43	7453	254.95	1514	373.32	13500	117.15	6944	61.40	6219	40.81	4584
GapA6	149.71	830	62.35	301	26.02	1502	20.94	1173	11.89	941	8.05	936
GapA7	1590.21	8782	674.19	4141	150.90	10229	141.59	6588	128.37	11418	39.82	5942
GapA8	966.51	6457	214.09	1571	171.31	5906	447.65	16429	38.15	4566	46.02	6585
GapA9	249.51	1773	131.63	851	47.29	3278	30.11	1788	28.73	4322	15.91	1918
GapA10	202.75	1481	99.26	663	49.56	1793	58.03	2830	11.27	1592	11.61	1902
GapA11	453.16	2948	202.59	1469	83.06	4139	52.55	4003	34.82	3950	29.78	3638
GapA12	172.48	711	75.72	259	28.26	1267	40.48	1582	23.32	2422	19.71	2263
GapA13	1259.98	7921	375.08	2785	225.48	5835	178.08	8121	74.78	9273	51.46	5533
GapA14	457.59	2564	105.53	686	76.08	2561	39.02	1362	13.27	1598	9.64	1269
GapA15	64.59	400	37.21	137	15.13	629	14.35	883	5.20	444	7.99	958
GapA17	1149.59	7485	632.90	4732	257.53	9915	318.19	12818	83.14	9718	71.76	8791
GapA18	44.16	210	25.33	90	8.58	333	15.84	1144	4.50	296	6.39	562
GapA19	103.30	707	60.21	371	31.62	1390	42.66	1938	10.91	1457	11.40	1371
GapB0	329.36	2726	156.45	1198	57.01	3019	40.10	2469	34.14	6563	24.71	3991
GapB1	153.94	515	107.92	364	85.13	4538	37.28	2028	26.52	2097	17.17	1481
GapB2	135.98	707	100.78	525	39.20	1452	19.84	1322	22.19	1654	10.21	1072
GapB3	443.73	2506	218.48	1464	241.46	6130	232.01	6369	182.33	21898	39.86	4653
GapB4	142.37	479	53.84	177	103.81	2922	51.90	2359	38.33	3834	24.43	1935
GapB5	135.17	663	73.69	293	37.71	1341	12.77	950	10.43	1072	9.06	987
GapB6	1453.05	5235	235.88	921	266.59	6174	137.24	3552	97.30	8142	65.43	5819
GapB7	502.27	1456	239.37	762	1511.32	13695	1576.83	29039	1232.78	35507	111.95	5838
GapB8	258.14	1118	108.63	488	78.34	2762	115.11	3841	23.28	2165	16.50	2046
GapB9	2481.82	12266	1169.28	7148	357.35	10643	426.77	9606	160.11	11381	137.97	11583
GapB10	430.82	1920	221.42	983	67.25	2991	87.53	3535	104.56	10363	24.04	2639
GapB11	289.88	1821	136.31	850	66.89	3408	47.62	1704	18.33	1559	16.79	1791
GapB12	279.63	2099	119.13	847	90.81	5025	43.34	2378	23.88	4139	13.49	1876
GapB13	150.53	655	98.04	352	51.77	1706	35.60	1391	21.15	1380	17.55	1200
GapB14	1426.84	7247	227.59	1670	378.43	11435	229.87	6587	49.51	4913	39.69	4923
GapB15	182.72	688	147.17	588	47.82	2652	46.56	2327	24.15	1637	34.31	2591
GapB16	338.46	1844	164.93	856	82.99	3216	124.10	7384	50.05	6335	35.71	3778
GapB17	172.90	592	98.50	313	52.18	1872	40.90	3819	23.59	1608	34.24	2760
GapB18	362.67	1511	145.57	629	86.24	2842	47.56	2253	31.09	2836	30.92	2470
GapB19	504.56	2067	262.97	1381	124.46	6792	86.89	3713	35.00	3322	31.20	3605
GapC0	1734.03	9701	771.22	5302	703.31	18537	202.28	10444	128.36	9097	62.03	8241
GapC1	119.87	350	51.74	125	31.01	1023	17.38	582	14.11	934	13.91	1083
GapC2	636.14	2892	268.06	1346	111.47	3880	106.62	6550	47.60	4391	26.84	3005
GapC3	255.65	688	166.20	379	122.20	3197	244.30	7190	60.78	3935	61.81	4427
GapC4	1113.72	3915	307.51	1379	323.91	8901	197.46	7123	324.81	22313	106.81	8616
GapC5	350.35	1652	163.98	744	185.31	5739	80.70	2370	62.08	5320	32.05	2703
GapC6	521.32	2860	254.84	1184	39.19	3533	76.40	2259	33.10	3807	30.56	2594
GapC7	2449.30	5834	635.10	1823	2810.47	25130	3229.30	35754	649.36	18702	676.58	24414
GapC8	270.68	701	292.18	935	170.67	4410	70.31	2465	98.40	6441	31.58	1919
GapC9	228.42	865	124.02	494	40.04	1387	52.23	1732	20.08	1358	18.65	1648
GapC10	214.66	825	114.02	464	143.54	4816	49.42	5222	27.13	1826	17.73	1678
GapC11	345.59	1296	158.06	662	108.62	5331	89.40	3323	54.34	6075	40.39	3727
GapC12	267.11	850	205.70	668	235.43	5161	65.28	4407	48.19	3406	37.76	4023
GapC13	78.66	189	83.48	222	15.94	432	19.04	779	10.49	702	16.40	1461
GapC14	299.87	988	138.77	537	69.77	1851	89.22	4283	26.02	1942	53.73	5896
GapC15	272.76	1285	98.48	485	42.90	1905	28.82	1463	11.47	1429	14.68	1926
GapC16	349.74	994	145.90	398	119.78	2272	64.26	2571	61.35	4496	27.58	2589
GapC17	1180.20	4703	290.82	1319	845.65	10389	156.33	5939	127.09	10503	77.22	6909
GapC18	402.56	1684	187.89	784	103.66	3162	122.25	3775	41.31	3553	22.89	2702
GapC19	472.60	2872	278.34	1743	295.92	17222	88.41	3862	51.37	6503	35.18	3966
GapA0U	1182.34	7063	171.13	1154	119.38	3883	53.51	2467	28.93	4073	20.68	2931
GapA1U	100.19	346	76.09	332	60.19	3572	132.68	6828	23.86	2300	11.79	1017
GapA2U	222.59	1424	109.37	662	28.20	1146	33.17	1803	13.58	1667	9.88	1269
GapA3U	938.46	5905	287.90	2064	363.71	19454	100.23	4087	56.64	5778	40.24	5706
GapA4U	138.61	1121	114.96	925	86.35	2852	50.51	1666	14.24	1758	10.46	2220
GapA5U	1455.73	10594	716.30	5008	194.72	6952	88.12	4190	73.45	8597	41.90	5418
GapA6U	140.73	998	87.08	460	52.41	2075	40.00	2159	14.29	1813	11.68	1299
GapA7U	315.96	2926	192.55	1777	80.82	3645	41.96	2910	19.71	3206	15.23	2450
GapA8U	272.98	1831	139.93	937	46.37	2173	31.89	1927	26.10	3652	19.08	3120
GapA9U	327.48	1999	77.82	496	52.54	2395	31.50	1677	12.17	1102	8.04	878
GapB0U	175.89	897	95.71	417	44.97	1977	51.73	2141	15.59	1644	13.68	1439
GapB1U	500.97	2518	205.88	928	126.45	4809	62.35	3161	89.06	7841	37.36	3276
GapB2U	1181.54	6560	282.48	1691	100.14	4358	108.05	5819	49.06	6742	37.77	3906
GapB3U	54.37	238	34.52	133	13.54	705	14.07	838	7.10	648	6.19	745
GapB4U	1424.40	9423	330.52	2346	224.46	10075	359.04	11934	56.66	6141	80.69	10102
GapB5U	127.30	495	135.10	576	212.00	4789	86.33	2201	182.20	12074	41.33	4373
GapB6U	1369.27	5026	432.64	1947	683.71	11483	254.05	7907	201.57	11922	97.36	8489
GapB7U	1492.97	5846	373.69	1627	1588.20	16288	322.93	9197	1753.78	45418	335.27	18986
GapB8U	1102.22	4711	271.36	1102	314.44	7906	84.13	3542	53.76	5390	55.42	4987
GapB9U	271.02	1480	171.68	1010	545.11	14427	143.56	9059	164.82	11993	44.51	5947
GapC0U	380.96	1325	170.14	573	373.29	17613	99.11	3424	43.90	4371	62.43	5294
GapC1U	506.71	2179	266.37	1155	200.38	5105	165.87	7468	123.57	7346	51.78	3301
GapC2U	92.69	531	80.23	357	21.20	862	11.44	493	7.41	670	8.03	824
GapC3U	603.83	2506	211.74	842	770.05	15095	79.74	3586	46.26	4640	38.20	4104
GapC4U	224.99	839	103.07	373	37.63	925	53.08	2803	17.44	1473	21.37	1646
GapC5U	2617.12	7482	680.56	2430	8002.82	47113	1653.59	28597	1153.13	35160	318.29	19091
GapC6U	79.19	319	95.31	350	20.62	1131	16.00	774	12.98	993	8.22	620
GapC7U	118.14	498	85.10	287	19.39	986	32.18	1061	17.84	1404	12.24	925
GapC8U	420.52	1635	302.20	1374	1311.41	17966	138.54	2902	59.97	5422	41.12	3914
GapC9U	257.51	800	146.78	456	83.83	2800	105.19	2876	37.24	3223	25.93	1657

Table 5.13: Computational comparison GAP instances with $d = 0$

	ABF		DB-Bf-Ls		DB-Cf-Ls		MW-Bf-Ns-V(o)		MW-Cf-Ns-V(o)	
Instance	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes
333GapC	0.71	47	1.94	249	0.96	51	1.58	160	0.97	67
431GapB	0.52	69	1.53	118	1.34	129	0.99	122	0.79	65
432GapA	0.59	52	1.99	255	1.79	316	1.02	164	1.04	109
433GapC	4.93	384	5.02	594	2.77	366	3.81	759	2.11	289
531GapB	1.03	175	1.56	206	1.29	206	1.00	133	0.94	190
532GapA	2.07	261	2.67	364	1.86	398	2.12	345	1.27	208
533GapC	3.85	581	3.45	489	2.12	382	1.86	303	1.37	272
632GapA	0.36	33	1.43	97	1.21	99	0.69	39	0.59	49
633GapC	0.91	150	2.68	526	1.23	130	2.11	457	0.94	108
732GapA	1.38	305	2.03	335	1.85	557	1.07	211	1.10	350
733GapC	4.28	792	3.88	716	3.28	781	2.41	745	2.40	885
832GapA	1.69	394	2.08	258	2.01	251	1.68	289	1.26	296
833GapC	1.71	304	2.54	335	2.28	443	1.49	314	1.60	338
931GapB	0.80	205	1.95	331	1.68	271	1.17	283	1.01	228
932GapA	0.96	126	2.76	438	1.19	90	1.54	202	0.90	82
933GapC	0.75	172	2.06	347	1.28	211	1.11	175	0.72	198
1031GapB	1.46	295	1.65	140	1.31	167	1.16	178	0.97	163
1032GapA	0.25	22	1.14	23	1.14	63	0.80	37	0.86	55
1033GapC	0.29	23	1.40	81	0.67	17	0.71	26	0.72	57
1132GapA	1.61	315	2.33	225	1.43	270	1.73	274	0.96	145
1133GapC	5.54	1099	3.25	825	2.33	643	2.51	807	2.15	784
1231GapB	0.37	22	1.62	84	0.81	18	1.51	141	0.75	67
1232GapA	0.72	102	2.45	359	2.02	276	1.54	215	1.20	174
1233GapC	0.83	46	1.79	192	1.73	195	1.40	199	1.57	289
1331GapB	1.02	272	3.83	612	1.83	364	1.24	327	1.23	337
1332GapA	0.99	167	1.94	259	2.20	276	1.41	140	1.24	200
1333GapC	4.41	820	4.54	937	3.21	991	3.29	886	2.45	846
1431GapB	0.75	183	2.69	606	3.86	415	2.04	552	2.90	520
1432GapA	0.68	91	2.32	286	1.43	133	1.25	157	1.05	247
1433GapC	4.41	562	3.51	580	3.76	964	3.50	839	2.28	564
1532GapA	0.86	61	1.64	189	1.08	74	1.06	91	1.21	285
1533GapC	1.83	269	2.63	335	1.88	307	1.66	295	1.31	252
1632GapA	1.91	212	2.28	318	2.14	292	1.85	287	1.69	428
1633GapC	2.32	352	4.51	931	2.23	351	2.97	719	1.74	417
1731GapB	0.86	164	1.33	174	1.12	137	0.96	83	1.03	212
1733GapC	3.28	416	5.17	943	3.50	666	2.86	639	2.23	470
1832GapA	0.84	132	1.52	137	1.46	127	1.16	160	1.06	102
1833GapC	3.70	633	3.39	647	2.95	806	2.02	707	1.81	512
1932GapA	0.52	93	1.02	55	1.29	69	0.66	49	0.60	29
1933GapC	0.48	90	1.72	341	2.20	310	1.75	264	1.42	432
2031GapB	0.38	66	1.38	95	1.51	80	1.35	135	1.07	121
2032GapA	0.78	71	1.83	158	1.34	116	1.61	207	0.93	99
2033GapC	4.52	1122	6.03	1604	2.71	870	3.47	1624	1.83	655
2132GapA	1.82	257	2.59	246	2.68	429	1.67	327	1.25	187
2133GapC	1.11	234	1.95	294	1.83	208	1.23	215	1.16	234
2232GapA	1.27	156	2.65	311	3.16	497	1.72	270	1.31	213
2233GapC	1.16	103	2.19	199	1.82	206	1.76	181	1.46	207
2331GapB	0.29	33	0.78	24	0.67	45	0.63	33	0.54	37
2332GapA	1.22	202	2.57	500	2.47	508	1.56	382	1.37	368
2333GapC	2.67	263	4.25	514	2.85	473	2.88	587	1.98	369
2431GapB	0.90	161	2.06	294	1.62	183	1.21	208	0.91	155
2432GapA	1.77	221	2.24	222	2.74	291	1.64	233	1.27	113
2433GapC	2.18	221	2.21	220	1.83	301	1.76	193	1.42	370
2532GapA	1.25	193	2.21	288	1.34	163	1.69	324	1.12	223
2533GapC	0.77	56	1.90	128	1.50	134	1.42	90	1.19	95
2632GapA	1.24	120	2.64	375	2.10	264	1.70	182	1.39	203
2633GapC	0.84	257	2.54	451	1.75	497	1.42	346	1.06	278
2731GapB	0.65	71	1.69	122	1.33	131	1.06	104	0.83	64
2732GapA	0.81	48	1.75	173	1.89	160	1.75	267	1.21	116
2733GapC	2.72	334	3.93	956	3.18	776	2.09	446	1.57	426
2831GapB	0.64	103	1.71	152	1.81	205	1.34	260	1.12	160
2832GapA	0.63	49	2.20	176	1.65	112	1.55	166	1.05	70
2833GapC	2.60	334	2.42	367	3.33	646	2.83	823	1.72	446
2931GapB	0.47	47	1.59	92	1.52	117	1.67	206	1.20	141
2932GapA	0.73	104	2.04	157	1.09	77	1.28	161	0.77	48
2933GapC	0.93	89	2.07	145	1.92	266	1.27	97	1.02	80
3032GapA	0.71	77	1.84	130	1.52	79	1.37	117	0.90	66
3033GapC	1.17	121	2.28	243	2.53	417	2.25	399	1.71	405
3131GapB	0.91	108	2.31	258	1.14	174	1.24	188	1.15	181
3132GapA	1.30	164	2.22	298	1.97	364	1.53	295	1.41	471
3133GapC	2.84	284	4.43	1023	2.41	377	3.62	809	1.96	404
3231GapB	0.27	12	1.66	83	0.73	42	0.82	34	0.73	76
3232GapA	1.94	456	3.08	636	2.60	524	1.77	482	1.23	300
3233GapC	0.76	49	3.12	484	2.57	243	1.94	382	2.01	383

Table 5.14: Computational comparison LGAP instances with $d = 0$

	ABF		DB-Bf-Ls		DB-Cf-Ls		MW-Bf-Ns-V(o)		MW-Cf-Ns-V(o)	
Instance	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes	time	BB nodes
GapA0	180.51	3598	206.89	6200	332.06	10782	69.91	10075	51.36	7051
GapA1	254.65	4298	518.42	11738	198.74	5376	52.11	5404	58.77	7912
GapA2	104.92	1632	51.50	1839	58.18	4050	15.17	1825	18.22	2406
GapA3	279.24	5786	116.14	5816	150.57	6215	39.96	4444	33.15	4240
GapA4	90.49	1983	27.62	1128	27.53	1140	6.12	800	6.93	813
GapA5	351.44	7282	1474.13	18487	290.18	11024	152.10	17926	110.05	16040
GapA6	25.83	497	36.67	1593	45.72	1542	9.34	1249	8.46	632
GapA7	673.16	14230	985.08	26893	570.78	11875	249.32	27138	87.90	11171
GapA8	233.24	4277	253.95	5653	113.17	4547	28.68	2819	31.58	3474
GapA9	192.42	3940	130.95	3502	79.98	3475	30.48	3857	16.94	2670
GapA10	80.28	1964	57.52	4706	69.41	2640	18.62	2411	17.64	2525
GapA11	290.59	5013	482.73	13463	192.75	6080	91.10	9537	72.61	13601
GapA12	294.77	5226	135.73	3213	126.72	4114	36.54	4902	25.97	2645
GapA13	753.77	18682	1005.50	27698	1439.72	41766	85.74	11575	137.77	16627
GapA14	62.78	1155	44.27	1628	40.81	1814	14.31	1893	21.58	2669
GapA15	53.17	1224	83.95	3846	83.90	3722	24.17	2820	30.37	4000
GapA17	347.50	6153	806.19	16242	256.85	6077	146.04	13576	53.64	5853
GapA18	29.46	563	19.63	1087	18.44	795	9.65	1089	12.06	1039
GapA19	31.71	619	112.02	3060	38.32	2037	11.77	1343	14.40	1639
GapB0	64.20	1514	108.57	3656	43.45	1778	12.89	1905	29.55	3681
GapB1	219.35	2386	62.00	2078	129.86	3310	24.92	3057	37.26	3392
GapB2	32.94	554	50.56	4020	67.67	2808	22.15	3684	21.67	2865
GapB3	144.92	2747	156.81	4558	114.52	4794	51.03	4657	46.40	4653
GapB4	29.76	326	73.52	2037	45.36	1422	14.58	1443	22.37	1868
GapB5	41.63	737	46.30	1173	18.23	978	15.00	1053	14.06	1300
GapB6	101.37	1234	96.10	3033	217.08	4956	26.88	2533	23.95	2234
GapB7	418.31	3537	337.37	6956	367.81	7293	85.06	6290	69.98	5603
GapB8	45.06	593	47.70	1762	94.51	3763	21.41	2179	26.21	2303
GapB9	784.66	10102	736.68	16135	412.11	14372	175.41	17731	88.72	10749
GapB10	115.05	1715	334.53	6094	92.50	3858	40.62	4347	29.57	4046
GapB11	73.87	1107	137.56	2839	133.89	3490	25.90	2809	27.07	2598
GapB12	380.74	7994	272.51	7181	316.04	6685	48.51	5926	65.02	6963
GapB13	56.75	729	66.99	1655	65.59	2772	18.94	1745	16.85	1680
GapB14	323.54	5379	276.74	10055	371.52	7900	63.17	9717	54.76	6432
GapB15	189.52	2937	144.69	3837	164.51	4748	60.71	5569	51.47	4162
GapB16	110.06	1872	120.79	3126	79.16	3020	30.82	3375	40.28	3814
GapB17	67.51	859	39.30	1359	55.72	1932	14.71	1101	17.96	1401
GapB18	112.37	1571	157.23	3555	217.06	5843	30.68	2845	73.54	7594
GapB19	183.18	2865	435.00	5194	229.34	7927	24.08	2238	34.46	3282
GapC0	656.67	11952	730.01	12613	1718.11	17191	110.22	12303	96.45	8452
GapC1	60.72	763	40.04	1378	65.59	1991	14.76	1132	12.55	1639
GapC2	190.81	2840	192.85	5690	118.91	4092	32.27	3614	32.25	3008
GapC3	80.59	841	114.49	2509	245.10	5585	39.18	3685	36.32	3539
GapC4	151.80	2002	325.22	5364	225.97	7310	46.06	4029	79.55	7462
GapC5	572.23	8238	340.60	8900	324.16	13383	86.84	8470	94.37	7523
GapC6	532.97	9362	474.99	8367	450.02	11545	54.17	7683	54.15	6170
GapC7	656.82	5840	654.17	7895	183.46	4384	85.65	4874	53.24	3666
GapC8	109.28	1030	128.60	2899	103.81	2209	75.98	7439	30.06	2032
GapC9	457.91	7741	576.43	9738	367.96	8002	45.30	4495	54.99	5846
GapC10	50.75	737	64.63	2056	56.33	1877	14.82	1420	25.66	3122
GapC11	129.17	1954	111.45	4956	118.78	2684	32.96	3577	36.88	3798
GapC12	120.30	1459	161.72	4017	162.13	4442	64.59	6322	53.04	5075
GapC13	62.97	620	81.43	1513	55.51	1428	15.81	1322	18.65	1447
GapC14	64.61	659	69.06	1689	210.63	3486	20.71	1958	29.07	2990
GapC15	446.99	8640	387.06	10909	464.01	8600	90.86	9768	72.41	9071
GapC16	311.47	3284	153.62	5820	190.83	4928	56.38	4800	33.65	3165
GapC17	199.93	2821	495.42	10481	325.08	15013	42.90	4767	76.42	8025
GapC18	175.99	2741	137.80	4290	387.79	7931	34.20	3168	39.36	4342
GapC19	359.34	7724	970.34	17147	621.00	12150	104.93	12240	58.47	6675
GapA0U	247.04	4182	144.39	4415	116.72	3087	33.54	3050	29.33	4273
GapA1U	40.64	482	29.13	1271	47.96	1497	14.54	1261	8.28	810
GapA2U	75.21	1669	51.26	1973	35.35	2268	13.34	1870	11.20	1647
GapA3U	635.70	14218	611.33	18192	584.43	15910	103.30	11463	120.07	18379
GapA4U	104.83	2840	84.15	4291	92.69	4129	21.53	4029	21.91	3506
GapA5U	401.88	7766	299.85	8351	188.75	6213	75.38	8068	42.06	4380
GapA6U	174.01	2950	52.34	2137	66.74	2864	15.22	1393	15.78	2037
GapA7U	218.87	4972	81.37	3738	87.05	4458	18.71	2140	18.91	2612
GapA8U	101.81	1973	116.71	4312	57.26	2797	31.77	3832	16.58	2321
GapA9U	195.76	3544	127.10	6034	126.51	4723	19.23	2102	19.88	2928
GapB0U	345.24	6844	193.40	3881	199.91	5199	31.74	3058	39.69	5111
GapB1U	128.95	1686	86.93	3573	90.87	2944	36.59	3381	29.41	2554
GapB2U	381.39	6203	119.02	5079	132.99	4917	33.88	4531	40.98	3984
GapB3U	29.71	383	46.29	1393	16.22	665	9.53	675	8.61	718
GapB4U	424.15	8107	269.06	5788	216.69	5753	48.34	5849	58.63	6238
GapB5U	369.94	5452	263.31	7345	218.22	6704	65.27	5527	66.20	7181
GapB6U	591.74	6829	491.47	6359	535.65	9656	75.64	6695	66.49	6034
GapB7U	232.56	3006	787.31	10413	453.05	9059	98.17	10343	53.99	4651
GapB8U	750.76	12510	858.13	14930	753.22	12720	138.27	12956	91.82	11758
GapB9U	84.58	1453	101.42	3537	122.17	5454	66.31	7790	28.49	3216
GapC0U	225.39	2956	180.79	4379	140.61	4195	32.52	3264	43.53	4074
GapC1U	187.04	2869	200.67	4871	473.38	6752	73.56	4344	99.99	6868
GapC2U	30.15	506	26.11	1155	22.82	1105	10.49	854	9.82	1018
GapC3U	158.19	2305	196.15	7898	87.91	4136	31.49	3535	44.07	3544
GapC4U	35.31	422	95.76	2045	48.04	1803	11.87	974	13.25	1340
GapC5U	626.71	7558	530.75	8288	1866.56	25748	79.03	7956	133.30	12640
GapC6U	71.04	1167	97.76	2326	60.90	2399	29.61	3022	20.78	2330
GapC7U	35.58	408	21.59	833	23.81	1075	16.54	1149	11.50	728
GapC8U	120.83	1579	120.81	3274	182.03	3983	123.92	10206	75.96	7266
GapC9U	387.38	4632	143.72	2655	250.91	4890	33.47	3141	39.75	3331

Chapter 6

Conclusions

This thesis addressed an important class of discrete facility location problems (FLPs) called *multi-level facility location problems* (MLFLPs). We concentrated our efforts in a general class of MLFLPs, denoted multi-level uncapacitated p-location problems (MUpLPs). In a sense, these problems can be considered fundamental given the fact that they generalize well-known discrete FLPs while maintaining the mathematical structure which enables the development of efficient solution methods. Moreover, MLFLPs appear as subproblems in various application-driven research areas. Motivated by this increasing interest in the field, and by identifying that most general versions of these problems lacked efficient exact algorithms to solve them, we embraced the challenge of this venture.

In Chapter 2 we reviewed the main references related to the field of MLFLPs, including uncapacitated and capacitated variants. There, we clearly defined the class of MLFLPs by identifying their main characteristics. We also pointed out the differences and similarities with well-known related areas. Based on the types of decisions involved, we determined three main categories within the area and proposed a classification scheme. Moreover, for each category of MLFLPs we presented overviews of formulations and solution algorithms. We noted that with the exception of one, all papers concerned with the development of exact algorithms (or polyhedral studies) referred to the special case of two levels. Thus, all the work to solve the most general versions of these problems comes from the approximation algorithms context. In particular, an important number of papers have been published relating to the development of approximation algorithms with performance guarantees for the multi-level uncapacitated facility location problem (MUFLP).

In Chapter 3 we introduced an alternative combinatorial representation for the MUFLP, which is one of the most representative MLFLPs. This new characterization considers paths as elements of the ground set to be selected or opened, in contrast with the classical one which uses the set of vertices. An important observation of this new representation of the problem is that the corresponding objective function satisfies the submodularity property. This clarified a previous conclusion from the literature where the problem was proved not to possess this property referring to the classical representation. Hence, we illustrated that the submodularity property is not intrinsic to an optimization problem but rather to a set function of its mathematical representation. Then, in Chapter 4 we were able to use this result pertaining to the submodular property in the general case of the MUpLP. We proposed a new MILP formulation which we denoted as *submodular formulation* and a greedy heuristic with worst-case performance results. Although the area of approximation algorithms with performance guarantees has been a rich field of research for MLFLPs, we were able to improve the best-known bound for the maximization version of the problem and develop sharper bounds for the multi-level p-median problem (MpMP). In particular, we obtained a $(1-1/e)$-approximation algorithm for the case in which profits are non-negative and additive. We also presented the computational results of our experiments where we assessed the performance of the submodular formulation and that of the greedy heuristic. The results confirm the efficiency of the submodular formulation over previous formulations modified for the MpMP. Our results also show that for the more general case of the MUpLP, none of the considered MILP formulations clearly dominates the others, although, on average, the submodular formulation performs best.

Finally, in Chapter 5 we studied an extension of the MUpLP in which link activation decisions between levels of facilities are considered. We denoted this version of the problem as MUpLP-E. We developed a sophisticated exact algorithm based on a Benders reformulation to solve large-scale instances of the problem. We included various techniques in order to enhance a straightforward implementation. Among others, we considered diverse methods to identify Pareto-optimal cuts, we exploited the network flow structure of the subproblems, considered updating the core point at every iteration, and added valid inequalities to the model. The results of extensive computational experiments confirm the efficiency of our Benders decomposition algorithm. Instances of the MUpLP-E with up to 3,000 customers, 250 potential facilities, and four levels of hierarchy were solved to optimality. We used two types of instances, one randomly generated, where the best version of the Benders algorithm was able to reduce by more than three orders of magnitude the average CPU time from that obtained using an arc-based formulation in a state-of-the-art general purpose solver. For the second, some hard benchmark instances with large LP gaps, a 90% reduction in the average CPU time was obtained compared with the previous formulation.

Although this is an area of research that has received considerable attention, we believe that there are several topics that are yet to be studied in this context. Exploring more general and challenging versions of the problems in order to assess the limits of these results is an important research avenue. Also, numerous variants of the fundamental FLPs in one level have been studied from an applications and theoretical point of view. Most of these versions could extend to the multi-level environment. For instance, some recent papers in MLFLPs have considered uncertain parameters, dynamic facility location where facilities can be opened and closed at each time period, or MLFLPs with service penalties where customers can be chose not to be served. Another important step towards a more systematic growth of the field is the incorporation of a common set of instances for MLFLPs that allows a more fair comparison of the techniques implemented.

Bibliography

[1] K. Aardal. *On the solution of one and two-level capacitated facility location problems by the cutting plane approach*. PhD thesis, Université Catholique de Louvain, Louvain-la-Neuve, Belgium, 1992.

[2] K. Aardal. Reformulation of capacitated facility location problems: How redundant information can help. *Annals of Operations Research*, 82:289–308, 1998.

[3] K. Aardal, M. Labbé, J. Leung, and M. Queyranne. On the two-level uncapacitated facility location problem. *INFORMS Journal on Computing*, 8(3): 289–301, 1996.

[4] K. Aardal, F. A. Chudak, and D. B. Shmoys. A 3-approximation algorithm for the k-level uncapacitated facility location problem. *Information Processing Letters*, 72:161–167, 1999.

[5] K. Aardal, P. L. Van den Berg, D. Gijswijt, and S. Li. Approximation algorithms for hard k-facility location problems. *European Journal of Operational Research*, 242:358–368, 2015.

[6] B. Addis, G. Carello, and A. Ceselli. Exactly solving a two-level location problem with modular node capacities. *Networks*, 59(1):161–180, 2012.

[7] B. Addis, G. Carello, and A. Ceselli. Combining very large scale and ILP

based neighborhoods for a two-level location problem. *European Journal of Operational Research*, 231(3):535–546, 2013.

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

[9] A. Ageev. Improved approximation algorithms for multilevel facility location problems. *Operations Research Letters*, 30(5):327–332, 2002.

[10] A. Ageev, Y. Ye, and J. Zhang. Improved combinatorial approximation algorithms for the k-level facility location problem. *SIAM Journal on Discrete Mathematics*, 18:207–217, 2004.

[11] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[12] M. Albareda-Sambola. Location-routing and location-arc routing. In G. Laporte, S. Nickel, and F. Saldanha da Gama, editors, *Location Science*, pages 399–418. Springer International Publishing, Cham, Switzerland, 2015.

[13] M. Alminyana, F. Borrás, and J. T. Pastor. A new directed branching heuristic for the pq-median problem. *Location Science*, 6:1–23, 1998.

[14] S. A. Alumur, H. Yaman, and B. Y. Kara. Hierarchical multimodal hub location problem with time-definite deliveries. *Transportation Research Part E: Logistics and Transportation Review*, 48(6):1107–1120, 2012.

[15] M. Asadi, A. Niknafs, and M. Ghodsi. An Approximation Algorithm for the k-level Uncapacitated Facility Location Problem with Penalties. *13th International Computer Society of Iran Computer Conference on Advances in Computer Science and Engineering, CSICC 2008*, 6:41–49, 2008.

[16] N. Asgari, E. Nikbakhsh, A. Hill, and R. Zanjirani Farahani. Supply chain management 1982-2015: a review. *IMA Journal of Mathematics in Management*, 27(3):353–379, 2016.

[17] D. Babayev. Comments on the note of Frieze. *Mathematical Programming*, 7: 249–252, 1974.

[18] M. Baïou and F. Barahona. A polyhedral study of a two level facility location model. *RAIRO - Operations Research*, 48(2):153–165, 2014.

[19] A. Balakrishnan, T. L. Magnanti, and P. B. Mirchandani. A dual-based algorithm for multi-level network design. *Management Science*, 40(5):567–581, 1994.

[20] A. I. Barros and M. Labbé. The multi-level uncapacitated facility location problem is not submodular. *European Journal of Operational Research*, 72: 607–609, 1994.

[21] A. I. Barros and M. Labbé. A general model for the uncapacitated facility and depot location problem. *Location Science*, 2(3):173–191, 1994.

[22] A. I. Barros, R. Dekker, and V. Scholten. A two-level network for recycling sand : A case study. *European Journal of Operational Research*, 110:199–214, 1998.

[23] J. E. Beasley. OR-Library: Distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41:1069–1072, 1990. URL `http://people.brunel.ac.uk/ mastjjb/jeb/info.html`.

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

[25] J. M. Bloemhof-Ruwaard, M. Salomon, and L. N. Van Wassenhove. On coordination of product and by-product flows in two-level distribution networks: Model formulations and solution procedures. *European Journal of Operational Research*, 79(2):325–339, 1994.

[26] J. M. Bloemhof-Ruwaard, M. Salomon, and L. N. Van Wassenhove. The capacitated distribution and waste disposal problem. *European Journal of Operational Research*, 88(3):490–503, 1996.

[27] A. Bumb. An approximation algorithm for the maximization version of the two level uncapacitated facility location problem. *Operations Research Letters*, 29 (4):155–161, 2001.

[28] A. Bumb and W. Kern. A simple dual ascent algorithm for the multilevel facility location problem. In M. Goemans, K. Jansen, J. Rolim, and L. Trevisan, editors, *Approximation, Randomization, and Combinatorial Optimization: Algorithms and Techniques. Lecture Notes in Computer Science*, volume 2129, pages 55–63. Springer, Berlin Heidelberg, 2001.

[29] J. Byrka, S. Li, and B. Rybicki. Improved approximation algorithm for k-level uncapacitated facility location problem (with penalties). *Theory of Computing Systems*, 58:19–44, 2016.

[30] G. Calinescu, C. Chekuri, P. Martin, and J. Vondrák. Maximizing a monotone submodular function subject to a matroid constraint. *SIAM Journal on Computing*, 40(6):1740–1766, 2011.

[31] A. B. Calvo and D. H. Marks. Location of health care facilities: An analytical approach. *Socio-Economic Planning Sciences*, 7(5):407–422, 1973.

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

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

[34] P. Chardaire, J.-L. Lutton, and A. Sutter. Upper and lower bounds for the two-level simple plant location problem. *Annals of Operations Research*, 86: 117–140, 1999.

[35] Y. Y. Chen and H. F. Wang. Applying a revised VAM to a multi-level capacitated facility location problem. *IEEM 2007: 2007 IEEE International Conference on Industrial Engineering and Engineering Management*, pages 337–341, 2007.

[36] Y. Çinar and H. Yaman. The vendor location problem. *Computers & Operations Research*, 38(12):1678–1695, 2011.

[37] M. Conforti and G. Cornuéjols. Submodular set functions, matroids and the greedy algorithm: tight worst-case bounds and some generalizations of the Rado-Edmonds theorem. *Discrete Applied Mathematics*, 7:251–274, 1984.

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

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

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

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

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

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

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

[45] G. Cornuéjols, M. L. Fisher, and G. L. Nemhauser. Location of bank accounts to optimize float: an analytic study of exact and approximate algorithms. *Management Science*, 23(8):789–810, 1977.

[46] G. Cornuéjols, G. L. Nemhauser, and L. A. Wolsey. The uncapacitated facility location problem. In P. B. Mirchandani and R. L. Francis, editors, *Discrete Location Theory*, pages 119–172. Wiley, New York, 1990.

[47] G. Şahin and H. Süral. A review of hierarchical facility location models. *Computers & Operations Research*, 34(8):2310–2331, Aug. 2007.

[48] G. Şahin, H. Süral, and S. Meral. Locational analysis for regionalization of Turkish red crescent blood services. *Computers & Operations Research*, 34(3): 692–704, 2007.

[49] R. Cuda, G. Guastaroba, and M. G. Speranza. A survey on two-echelon routing problems. *Computers & Operations Research*, 55:185–199, 2014.

[50] M. S. Daskin, L. V. Snyder, and R. T. Berger. Facility location in supply chain design. In A. Langevin and D. Riopel, editors, *Logistics Systems: Design and Optimization*, pages 39–65. Springer, Boston, 2005.

[51] M. A. Drexl. An approximation algorithm for the k-level concentrator location problem. *Operations Research Letters*, 39(5):355–358, 2011.

[52] Z. Drezner and H. W. Hamacher, editors. *Facility Location: Applications and Theory*. Springer-Verlag, Berlin, 2002.

[53] D. Du, X. Wang, and D. Xu. An approximation algorithm for the k-level capacitated facility location problem. *Journal of Combinatorial Optimization*, 20(4):361–368, 2009.

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

[55] N. J. Edwards. *Approximation algorithms for the multi-level facility location problem*. PhD thesis, Cornell University, Ithaca, New York, 2001.

[56] H. A. Eiselt and V. Marianov. *Foundations of Location Analysis*, volume 155. Springer, New York, International Series in Operations Research & Management Science edition, 2011.

[57] D. Erlenkotter. A dual-based procedure for uncapacitated facility location. *Operations Research*, 26(6):992–1009, 1978.

[58] F. Errico, T. G. Crainic, F. Malucelli, and M. Nonato. A Benders decomposition approach for the symmetric TSP with generalized latency arising in the design of semiflexible transit systems. *Transportation Science*, 2017. doi: 10.1287/trsc.2015.0636. Published online in Articles in Advanced.

[59] U. Feige. A threshold of ln n for approximating set cover. *Journal of the Association of Computing Machinery*, 45(4):634–652, 1998.

[60] D. R. M. Fernandes, C. Rocha, D. Aloise, G. M. Ribeiro, E. M. Santos, and A. Silva. A simple and effective genetic algorithm for the two-stage capacitated facility location problem. *Computers & Industrial Engineering*, 75(1):200–208, 2014.

[61] M. Fischetti, I. Ljubić, and M. Sinnl. Redesigning Benders decomposition for large-scale facility location. *Management Science*, 0(0):null, 2016. doi: 10.1287/mnsc.2016.2461.

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

[63] M. L. Fisher, G. L. Nemhauser, and L. A. Wolsey. An analysis of approximations for maximizing submodular set functions - II. *Mathematical Programming*, 8: 73–87, 1978.

[64] R. Fleischer, J. Li, S. Tian, and H. Zhu. Non-metric multicommodity and multilevel facility location. *Lecture Notes in Computer Science*, 4041:138–148, 2006.

[65] A. Frieze. A cost function property for plant location problems. *Mathematical Programming*, 7:245–248, 1974.

[66] A. F. Gabor and J.-K. C. W. van Ommeren. A new approximation algorithm for the multilevel facility location problem. *Discrete Applied Mathematics*, 158 (5):453–460, 2010.

[67] L.-l. Gao and E. P. J. Robinson. A dual-based optimization procedure for the two-echelon uncapacitated facility location problem. *Naval Research Logistics*, 39(2):191–212, 1992.

[68] B. Gendron and F. Semet. Formulations and relaxations for a multi-echelon capacitated location-distribution problem. *Computers & Operations Research*, 36(5):1335–1355, 2009.

[69] B. Gendron, P.-V. Khuong, and F. Semet. Multilayer variable neighborhood search for two-level uncapacitated facility location problems with single assignment. *Networks*, 66(3):214–234, 2015.

[70] B. Gendron, P.-V. Khuong, and F. Semet. A Lagrangian-based branch-and-bound algorithm for the two-level uncapacitated facility location problem with single-assignment constraints. *Transportation Science*, 50(4):1286–1299, 2016.

[71] B. Gendron, P.-V. Khuong, and F. Semet. Comparison of formulations for the two-level uncapacitated facility location problem with single assignment constraints. *Computers & Operations Research*, 2017. In press.

[72] A. M. Geoffrion. Multicommodity distribution system design by Benders decomposition. *Management Science*, 20(5):822–844, 1974.

[73] V. Ghezavati, F. Soltanzadeh, and A. Hafezalkotob. Optimization of reliability for a hierarchical facility location problem under disaster relief situations by a chance-constrained programming and robust optimization. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability*, 229(6):542–555, 2015.

[74] B. Goldengorin. Maximization of submodular functions: Theory and enumeration algorithms. *European Journal of Operational Research*, 198(1):102–112, 2009.

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

[76] S. Gollowitzer, L. Gouveia, and I. Ljubić. Enhanced formulations and branch-and-cut for the two level network design problem with transition facilities. *European Journal of Operational Research*, 225(2):211–222, 2013.

[77] É. Gourdin, M. Labbé, and H. Yaman. Telecommunication and location. In Z. Drezner and H. Hamacher, editors, *Facility Location: Applications and Theory*, pages 275–305. Springer, Berlin, 2002.

[78] H. J. Greenberg. Diagnosing infeasibility in min-cost network flow problems part I: Dual infeasibility. *IMA Journal of Mathematics in Management*, 1: 99–109, 1987.

[79] M. Grötschel. Introduction. In M. Grötschel, editor, *Documenta Mathematica, Optmization Stories*, pages 3–6. Bielefeld, Deutschland, 2012.

[80] S. Guha, A. Meyerson, and K. Munagala. Hierarchical placement and network design problems. *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 603–612, 2000.

[81] P. Guo, W. Cheng, and Y. Wang. Hybrid evolutionary algorithm with extreme machine learning fitness function evaluation for two-stage capacitated facility location problems. *Expert Systems with Applications*, 71:57–68, 2017.

[82] A. Gupta and J. Könemann. Approximation algorithms for network design: A survey. *Surveys in Operations Research and Management Science*, 16(1):3–20, 2011.

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

[84] S. L. Hakimi and S. N. Maheshwari. Optimum locations of centers in networks. *Operations Research*, 20(5):967–973, 1972.

[85] Y. Hinojosa, J. Puerto, and F. Fernández. A multiperiod two-echelon multicommodity capacitated plant location problem. *European Journal of Operational Research*, 123:271–291, 2000.

[86] D. S. Hochbaum. Heuristics for the fixed cost median problem. *Mathematical Programming*, 22(1):148–162, 1982.

[87] J. Hodgson. Alternative approaches to hierarchical location-allocation systems. *Geographical Analysis*, 16(3):275–281, 1984.

[88] A. A. V. Ignacio, V. J. M. F. Filho, and R. D. Galvão. Lower and upper bounds for a two-level hierarchical location problem in computer networks. *Computers & Operations Research*, 35(6):1982–1998, 2008.

[89] E. Kantor and D. Peleg. Approximate hierarchical facility location and applications to the bounded depth Steiner tree and range assignment problems. *Journal of Discrete Algorithms*, 7(3):341–362, Sept. 2009.

[90] L. Kaufman, M. Eede, and P. Hansen. A plant and warehouse location problem. *Operational Research Quarterly*, 28(3):547–554, 1977.

[91] A. Klose. An LP-based heuristic for two-stage capacitated facility location problems. *Journal of the Operational Research Society*, 50:157–166, 1999.

[92] A. Klose. A Lagrangean relax-and-cut approach for the two-stage capacitated facility location problem. *European Journal of Operational Research*, 126(2): 408–421, 2000.

[93] Y. Kochetov and D. Ivanenko. Computationally difficult instances for the uncapacitated facility location problem. In T. Ibaraki, K. Nonobe, and M. Yag-

iura, editors, *Metaheuritics: Progress as Real Problem Solvers. Operations Research/Computer Science Interfaces Series*, volume 32, pages 351–367. Springer, Boston, 2005.

[94] V. Korać, J. Kratica, and A. Savić. An improved genetic algorithm for the multi level uncapacitated facility location problem. *International Journal of Computers, Communications and Control*, 8(6):845–853, 2013.

[95] B. Korte and D. Hausman. An analysis of the greedy heuristic for independence systems. *Annals of Discrete Mathematics*, 2:65–74, 1978.

[96] J. Krarup and P. Pruzan. Ingredients of locational analysis. In P. B. Mirchandani and R. L. Francis, editors, *Discrete Location Theory*, pages 1–54. Wiley, New York, 1990.

[97] J. Kratica, D. Dugošija, and A. Savić. A new mixed integer linear programming model for the multi level uncapacitated facility location problem. *Applied Mathematical Modelling*, 38(7-8):2118–2129, 2014.

[98] R. Krishnaswamy and M. Sviridenko. Inapproximability of the multi-level uncapacitated facility location problem. In *Proceedings of the Twenty-third Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 718–734, Philadelphia, USA, 2012.

[99] R. Krishnaswamy and M. Sviridenko. Inapproximability of the Multilevel Uncapacitated Facility Location Problem. *ACM Transactions on Algorithms*, 13 (1):1–26, 2016.

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

[101] A. Kulik, H. Shachnai, and T. Tamir. Maximizing submodular set functions subject to multiple linear constraints. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 545–554, 2009.

[102] A. Kulik, H. Shachnai, and T. Tamir. Approximations for monotone and non-monotone submodular maximization with knapsack constraints. *Mathematics of Operations Research*, 38(4):729–739, 2013.

[103] M. Landete and A. Marín. New facets for the two-stage uncapacitated facility location polytope. *Computational Optimization and Applications*, 44(3):487–519, jan 2009.

[104] G. Laporte, S. Nickel, and F. Saldanha da Gama, editors. *Location Science*. Springer International Publishing, Cham, Switzerland, 2015.

[105] A. N. Letchford and S. J. Miller. An aggressive reduction scheme for the simple plant location problem. *European Journal of Operational Research*, 234(3):674–682, 2014.

[106] G. Li, D. Xu, D. Du, and C. Wu. Approximation algorithms for the multilevel facility location problem with linear/submodular penalties. *Lecture Notes in Computer Science*, 9130:162–169, 2015.

[107] J. Li, F. Chu, C. Prins, and Z. Zhu. Lower and upper bounds for a two-stage capacitated facility location problem with handling costs. *European Journal of Operational Research*, 236(3):957–967, 2014.

[108] I. Litvinchev and E. L. Ozuna Espinosa. Solving the two-stage capacitated facility location problem by the Lagrangian heuristic. *Computational Logistics: Third International Conference, ICCL 2012*, pages 92–103, 2012.

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

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

[111] T. L. Magnanti and R. T. Wong. Decomposition methods for facility location problems. In P. B. Mirchandani and R. L. Francis, editors, *Discrete Location Theory*, pages 209–262. Wiley, New York, 1990.

[112] T. L. Magnanti, P. Mireault, and R. T. Wong. Tailoring benders decomposition for uncapacitated network design. In G. Gallo and C. Sandi, editors, *Netflow at Pisa*, pages 112–154. Springer, Berlin, Heidelberg, 1986.

[113] S. Mancini. Multi-echelon distribution systems in city logistics. *European Transport/Trasporti Europei*, (54):1–24, 2013.

[114] S. Mancini, J. Gonzalez-Feliu, and T. G. Crainic. Planning and optimization methods for advanced urban logistics systems at tactical level. In J. Gonzalez-Feliu, F. Semet, and J.-L. Routhier, editors, *Sustainable Urban Logistics: Concepts, Methods and Information Systems*, pages 145–164. Springer, Berlin Heidelberg, 2014.

[115] R. Manzini and F. Bindi. Strategic design and operational management optimization of a multi stage physical distribution system. *Transportation Research Part E: Logistics and Transportation Review*, 45(6):915–936, 2009.

[116] M. Marić. An efficient genetic algorithm for solving the multi-level uncapacitated facility location problem. *Computing and Informatics*, 29:183–201, 2010.

[117] M. Marić, Z. Stanimirović, A. Djenić, and P. Stanojević. Memetic algorithm for solving the multilevel uncapacitated facility location problem. *Informatica*, 25(3):439–466, 2014.

[118] A. Marín. Lower bounds for the two-stage uncapacitated facility location problem. *European Journal of Operational Research*, 179(3):1126–1142, 2007.

[119] A. Marín and B. Pelegrín. Applying Lagrangian relaxation to the resolution of two-stage location problems. *Annals of Operations Research*, 86:179–198, 1999.

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

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

[122] L. P. Melo, F. K. Miyazawa, L. L. C. Pedrosa, and R. C. S. Schouery. Approximation algorithms for k-level stochastic facility location problems. *Journal of Combinatorial Optimization*, In press, 2016. doi: 10.1007/s10878-016-0064-2.

[123] M. T. Melo, S. Nickel, and F. Saldanha-da-Gama. Facility location and supply chain management - A review. *European Journal of Operational Research*, 196 (2):401–412, July 2009.

[124] G. Memişoğlu and H. Üster. Integrated Bioenergy Supply Chain Network Planning Problem. *Transportation Science*, 50(1):35–56, 2016.

[125] A. Meyerson, K. Munagala, and S. Plotkin. Cost-distance: Two metric network design. *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 624–630, 2000.

[126] P. B. Mirchandani. Generalized hierarchical facility locations. *Transportation Science*, 21(2):123–125, 1987.

[127] P. B. Mirchandani. The p-median problem and generalizations. In P. B. Mirchandani and R. L. Francis, editors, *Discrete Location Theory*, pages 119–172. Wiley, New York, 1990.

[128] S. Mišković and Z. Stanimirović. A memetic algorithm for solving two variants of the two-stage uncapacitated facility location problem. *Information Technology and Control*, 42(2):131–149, 2013.

[129] S. Mišković, Z. Stanimirović, and I. Grujičić. Solving the robust two-stage capacitated facility location problem with uncertain transportation costs. *Optimization Letters*, In press, 2016. doi: 10.1007/s11590-016-1036-2.

[130] P. Mitropoulos, I. Giannikos, and I. Mitropoulos. Exact and heuristic approaches for the locational planning of an integrated solid waste management system. *Operational Research-An International Journal*, 9(3):329–347, 2009.

[131] S. C. Narula. Hierarchical location-allocation problems: A classification scheme. *European Journal of Operational Research*, 15(1):93–99, 1984.

[132] G. L. Nemhauser and L. A. Wolsey. Maximizing submodular functions: formulations and analysis of algorithms. *Annals of Discrete Mathematics*, 11:279–301, 1981.

[133] G. L. Nemhauser and L. A. Wolsey. *Integer and Combinatorial Optimization*, pages 710–715. Wiley Intersciences Series in Discrete Mathematics and Optimization, New York, 1988.

[134] G. L. Nemhauser, L. A. Wolsey, and M. L. Fisher. An analysis of approximations

for maximizing submodular set functions - I. *Mathematical Programming*, 14: 265–294, 1978.

[135] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte. The multi-level facility location problem as the maximization of a submodular set function. *European Journal of Operational Research*, 247:1013–1016, 2015.

[136] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte. Formulations and approximation algorithms for multi-level uncapacitated facility location. *INFORMS Journal on Computing, In press*, 2017.

[137] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte. An exact algorithm for multi-level uncapacitated facility location. *Submitted to Transportation Science*, 2017.

[138] C. Ortiz-Astorquiza, I. Contreras, and G. Laporte. Multi-level facility location problems. *Submitted to European Journal of Operational Research*, 2017.

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

[140] H. Pirkul and V. Jayaraman. Production, transportation, and distribution planning in a multi-commodity tri-echelon system. *Transportation Science*, 30 (4):291–302, 1996.

[141] H. Pirkul and V. Jayaraman. A multi-commodity, multi-plant, capacitated facility location problem: formulation and efficient heuristic solution. *Computers & Operations Research*, 25(10):869–878, 1998.

[142] M. S. Pishvaee, J. Razmi, and S. A. Torabi. An accelerated Benders decomposition algorithm for sustainable supply chain network design under uncertainty

: A case study of medical needle and syringe supply chain. *Transportation Research Part E*, 67:14–38, 2014.

[143] S.-U. Rahman and D. K. Smith. Use of location-allocation models in health service development planning in developing nations. *European Journal of Operational Research*, 123(3):437–452, 2000.

[144] R. Rahmaniani, T. G. Crainic, M. Gendreau, and W. Rei. The Benders decomposition algorithm: a literature review. *Technical Report CIRRELT-2016-30*, 2016.

[145] H.-B. Ro and D.-W. Tcha. A branch and bound algorithm for the two-level uncapacitated facility location problem with some side constraints. *European Journal of Operational Research*, 18:349–358, 1984.

[146] D. Serra and C. S. ReVelle. The *pq*-median problem: location and districting of hierarchical facilities. *Location Science*, 1(4):299–312, 1993.

[147] D. Serra and C. S. ReVelle. The *pq*-median problem: Location and districting of hierarchical facilities II. Heuristic solution methods. *Location Science*, 2(2): 63–82, 1994.

[148] J.-B. Sheu and A. Y.-S. Lin. Hierarchical facility network planning model for global logistics network configurations. *Applied Mathematical Modelling*, 36(7): 3053–3066, 2012.

[149] D. B. Shmoys, É. Tardos, and K. Aardal. Approximation algorithms for facility location problems. In *Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing*, volume 5, pages 265–274, New York, 1997. Association for Computing Machinery.

[150] D. Simchi-Levi, P. Kaminsky, and E. Simchi-Levi. *Managing the Supply Chain: The Definitive Guide for the Business Professional*. McGraw-Hill, New York, 2004.

[151] K. Spielberg. Plant location with generalized search origin. *Management Science*, 16(3):165–178, 1969.

[152] M. Sviridenko. A note on maximizing a submodular set function subject to a knapsack constraint. *Operations Research Letters*, 32(1):41–43, Jan. 2004.

[153] D. W. Tcha and B. Lee. A branch-and-bound algorithm for the multi-level uncapacitated facility location problem. *European Journal of Operational Research*, 18(1):35–43, 1984.

[154] J. C. Teixeira and A. P. Antunes. A hierarchical location model for public facility planning. *European Journal of Operational Research*, 185(1):92–104, 2008.

[155] J. M. Tien, K. El-Tell, and G. R. Simons. Improved formulations to the hierarchical health facility location-allocation problem. *IEEE Transactions on Systems, Man and Cybernetics*, 13(6):1128–1132, 1983.

[156] S. Tragantalerngsak, J. Holt, and M. Ronnqvist. Lagrangian heuristics for the two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research*, 102:611–625, 1997.

[157] S. Tragantalerngsak, J. Holt, and M. Ronnqvist. An exact method for the two-echelon, single-source, capacitated facility location problem. *European Journal of Operational Research*, 123:473–489, 2000.

[158] J. Vondrák. *Submodularity in combinatorial optimization*. PhD thesis, Charles University, Prague, 2007.

[159] K. Wang and Q. Yang. Hierarchical facility location for the reverse logistics network design under uncertainty. *Journal of Uncertain Systems*, 8(4):255–270, 2014.

[160] Z. Wang, D. Du, A. F. Gabor, and D. Xu. An approximation algorithm for the k-level stochastic facility location problem. *Operations Research Letters*, 38(5): 386–389, 2010.

[161] Z. Wang, D. Du, A. F. Gabor, and D. Xu. Erratum to: "An approximation algorithm for the k-level stochastic facility location problem" [Oper. Res. Lett. 38 (2010) 386-389]. *Operations Research Letters*, 39(2):160–161, 2011.

[162] J. R. Weaver and R. L. Church. The nested hierarchical median facility location model. *INFOR*, 29(2):100–115, 1991.

[163] R. E. Wendell and A. P. Hurter. Optimal locations on a network. *Transportation Science*, 7(1):18–33, 1973.

[164] B. L. Wildbore. *Theoretical and computational analysis of the two-stage capacitated plant location problem*. PhD thesis, Massey University, Palmerston North, New Zealand, 2008.

[165] L. A. Wolsey. Fundamental properties of certain discrete location problems. In J.-F. Thisse and H. G. Zoller, editors, *Locational Analysis of Public Facilities*, pages 331–356. North-Holland, Amsterdam, 1983.

[166] T. Wu, F. Chu, Z. Yang, and Z. Zhou. A Lagrangean relaxation approach for a two-stage capacitated facility location problem with choice of facility size. *Proceedings - 2015 IEEE International Conference on Systems, Man, and Cybernetics, SMC 2015*, pages 713–718, 2015.

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

[168] R. Zanjirani Farahani, M. Hekmatfar, B. Fahimnia, and N. Kazemzadeh. Hierarchical facility location problem: Models, classifications, techniques, and applications. *Computers & Industrial Engineering*, 68:104–117, 2014.

[169] J. Zhang. Approximating the two-level facility location problem via a quasi-greedy approach. *Mathematical Programming*, 108:159–176, 2006.

[170] J. Zhang and Y. Ye. A note on the maximization version of the multi-level facility location problem. *Operations Research Letters*, 30(5):333–335, 2002.