universität wien

DISSERTATION

Titel der Dissertation

Mixed Integer Programming approaches to problems combining network design and facility location

Verfasser

Dipl.Ing. Stefan Gollowitzer

angestrebter akademischer Grad

Doctor of Philosophy (PhD)

Wien, 2012

Studienkennzahl lt. Studienblatt:	A 094 136
Dissertationsgebiet lt. Studienblatt:	Statistics and Operations Research
Betreuerin / Betreuer:	Univ.-Prof. Mag. Dr. Immanuel Bomze
	Ass.-Prof. Mag. Dr. Ivana Ljubic

Für Catherine

Abstract

In recent years, telecommunication service providers started to adapt their local access networks to the steadily growing demand for bandwidth of internet-based services. Most existing local access networks are based on copper cable and offer a limited bandwidth to customers. A common approach to increase this bandwidth is to replace parts of the network by fiber-optic cable. This requires the installation of facilities, where the optical signal is transformed into an electrical one and vice versa.

Several strategies are commonly used to deploy fiber-optic networks. Connecting each customer via a fiber-optic link is referred to as Fiber-to-the-Home. If there is a fiber-optic connection for every building this is commonly referred to as Fiber-to-the-Building. If a fiber-optic connection leads to each facility that serves an entire neighborhood, this is referred to as Fiber-to-the-Curb.

In this thesis we propose mathematical optimization models for the cost-efficient design of local access networks based on fiber-optic cable. These models cover several aspects, including the Fiber-to-the-Curb strategy under additional reliability constraints, mixed Fiber-to-the-Home and Fiber-to-the-Curb strategies and capacity planning of links and facilities for Fiber-to-the-Curb networks.

We provide a theoretical analysis of the proposed models and develop efficient solution algorithms. We use state-of-the-art methods from combinatorial optimization including polyhedral comparisons, reformulations on extended graphs, valid inequalities and branch-and-cut procedures.

Zusammenfassung

Viele heutzutage über das Internet angebotene Dienstleistungen benötigen wesentlich höhere Bandbreiten als von bestehenden lokalen Zugangsnetzen bereitgestellt werden. Telekommunikationsanbieter sind daher seit einigen Jahren bestrebt, ihre zum Großteil auf Kupferkabeln basierenden Zugangsnetze entsprechend zu modernisieren. Die gewünschte Erweiterung der bereitgestellten Bandbreiten wird oftmals erzielt, indem ein Teil des Kupfernetzes durch Glasfaser ersetzt wird. Dafür sind Versorgungsstandorte notwendig, an welchen die optischen und elektrischen Signale jeweils in einander umgewandelt werden.

In der Praxis gibt es mehrere Strategien für die Installation von optischen Zugangsnetzen. Fiber-to-the-Home bezeichnet Netze, in denen jeder Haushalt direkt per Glasfaser angebunden wird. Wird je Wohngebäude eine optische Verbindung bereitgestellt, nennt man dies Fiber-to-the-Building. Endet die Glasfaserverbindung an einem Versorgungsstandort, welcher die Haushalte eines ganzen Wohnviertels durch Kupferkabel versorgt, bezeichnet man dies als Fiber-to-the-Curb.

Inhalt dieser Dissertation sind mathematische Optimierungsmodelle für die kosteneffiziente Planung von auf Glasfaser basierenden lokalen Zugangsnetzen. Diese Modelle decken mehrere Aspekte der Planung ab, darunter die Fiber-to-the-Curb-Strategie mit zusätzlichen Restriktionen betreffend Ausfallssicherheit, gemischte Fiber-to-the-Home und Fiber-to-the-Curb-Netze sowie die Kapazitätenplanung von Fiber-to-the-Curb-Netzen.

Ergebnis dieser Dissertation sind die theoretische Analyse der beschriebenen Modelle sowie effiziente Lösungsalgorithmen. Es kommen Methoden der kombinatorischen Optimierung zum Einsatz, darunter Umformulierungen auf erweiterten Graphen, zulässige Ungleichungen und Branch-and-Cut-Verfahren.

Acknowledgements

I'd like to thank everyone who accompanied me during the sometimes tedious, but most often delightful work on this thesis.

Special thanks go to an incredibly successful teacher and collaborator, Ivana Ljubić. She introduced me to a precise way of working and the arts of programming and writing papers. I'd also like to thank Prof. Immanuel Bomze, who provided me with invaluable support throughout the last three years. He made me aware of the beauty of all these other fields of mathematics. To Prof. Bernard Gendron I am thankful for his great support throughout the application for my research grant and the warm welcome to Montreal. Luis Gouveia deserves many thanks for enabling my fruitful visits in Lisbon, organizing informative and entertaining winter schools and giving me advice whenever needed.

Cordial thanks go to my new and old friends. Those in Montreal for the great time I had and those at home for overlooking the many times I went around the globe instead of spending time with them. Last, and most of all, I'd like to thank my family. They have been encouraging throughout all my endeavors and stood up for all of my decisions.

Contents

Chapter 1

Introduction

The internet has been a success story, starting with its earliest predecessors in the 1960s. Since the last restrictions on commercial traffic were removed in 1995, it has made a significant impact on cultural and economic activities around the world. Weblogs, e-mails, instant messaging and social platforms have changed the way humans interact. Online shopping, auction websites and advertisements are new and widely used ways of commerce. Permanent and immediate access to music and films have changed the human cultural habits.

Many of the aforementioned activities require the transfer of increasing data volumes but the existing infrastructure in many places is neither designed for nor capable of providing the required services. Decision makers have recognized the economic and social importance of access to internet-based services. The United Nations stated that internet access is a human right [30] and the European Commission underlined the importance of communication tools provided by the internet in its *Digital Agenda for Europe* [10]. This has triggered major efforts to improve the infrastructure underlying the internet.

Regarding this infrastructure one can distinguish between two major components. Backbone networks connect geographically separate regions or cities. Local access networks (also known as *Last Mile*) connect end-users (e.g., households) to the backbone network via a so-called central office. Fiber-optic cables have been the standard medium for backbone networks for considerable time. However, local access networks are still based on copper wires of existing telephone lines. These provide only very limited bandwidths and are therefore not suitable for many of today's internet-based services.

In the last decade telecommunication providers started to make huge efforts to improve their local access networks [6, 7]. These efforts lead to a number of complex planning problems. This work will focus on mathematical models and their efficient solution for some of these problems.

1.1 Access networks based on optical fibers

In the last decade networks based on optical fibers have become the first choice to replace existing local access networks based on copper cable [32]. The bandwidth provided by the newly deployed fiber-optic networks depends on the extent to which the copper cable between an end-user and the central office is replaced. We distinguish the following strategies:

Fiber-to-the-Curb (FTTC) The first part of the connection between the central office and end-users is replaced by fiber-optic cable. A distance of up to a few hundred meters between end-users and a cabinet are covered by copper cable. The fiber-optic line terminates at the cabinet. A similar strategy to FTTC is Fiber-to-the-Neighbourhood (FTTN), where the closest cabinet is located up to 1500 meters from an end-user.

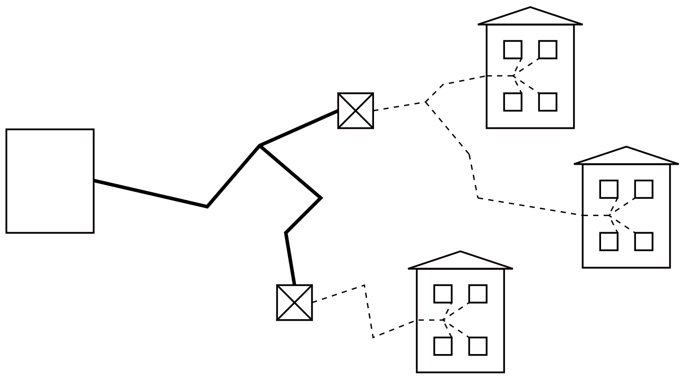
Fiber-to-the-Building (FTTB) An optical link leads to every building. Cabinets are often placed in the basement of, e.g., an apartment building.

Fiber-to-the-Home (FTTH) Existing copper cable is entirely replaced. The fiber-optic network is deployed all the way to the end-user.

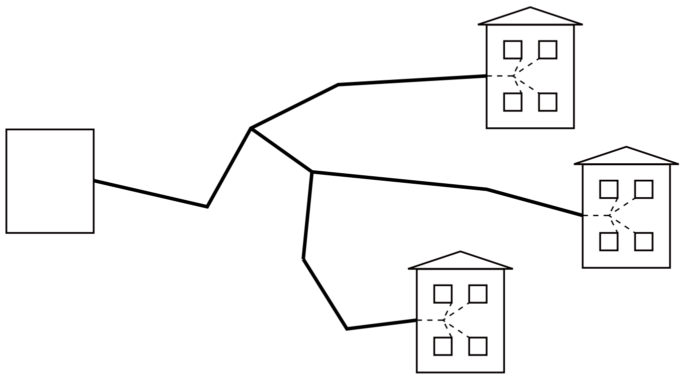
Figure 1.1 (by kind permission of Peter Putz [24]) illustrates the different strategies. The rectangle to the left represents the central office, solid lines represent fiber-optic links and dashed lines represent copper links. Crossed squares represent cabinets. On the right hand of the figure there are three buildings, in each building there are four customers.

Common to all the illustrated approaches is that the optical signal has to be transformed into an electrical signal at the locations where the optical link is continued by existing copper wires. Dependent on the number of connections that are served by one optical link, different devices are used to transform the signal. These can be multiplexors or splitters.

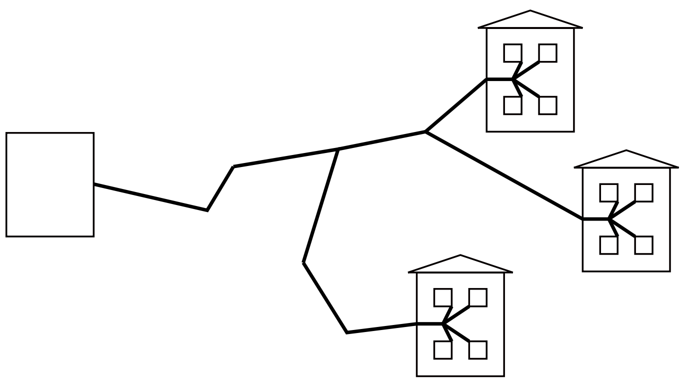
Fiber-optic networks are based on either of two common architectures, active optical networks (AONs) or passive optical networks (PONs). They differ in the way the signals to different end-points are distributed. In AONs a router or switch divides the signal such that each end-point only receives the signal that is intended for it. In PONs the signals for several end-points use the same fiber from the central office until close to the end-points. From



(a) FTTC. Fiber optic connections from the central server to two multiplexers and copper connections to the customers.



(b) FTTB. Fiber optic connections from the central server to the buildings that host the customers. Copper connections inside the buildings.



(c) FTTH. The entire connection from the central server to the customers consists of optical fiber.

Figure 1.1: Illustration of FTTC, FTTB and FTTH strategies.

there a splitter sends a copy of the signal to each end-point. There are architectures that allow multiple splitters between the central office and each end-point.

Another distinction is between greenfield and brownfield deployment. The former term refers to networks that are designed and installed from scratch, the latter indicates that existing infrastructure is taken into account.

Chapter 2

Models for fiber-optic networks

The optimization literature on planning issues of local access networks is not clearly structured. We will use the following notions in order to classify the existing literature.

Decomposed vs. integrated Planning local access networks requires several decision along the connection from the end-user to the central office: a) the location of cabinets, b) the assignment of end-users to one of the cabinets and c) the path of the optical link from the cabinet to the central office. We use the term integrated planning if steps a), b) and c) are done at once and decomposed if steps a) and b) are done separately from step c).

For FTTH and FTTB networks steps a) and b) can be omitted if there is a separate fiber connecting each end-user to the central office, otherwise cabinets host splitters where the fibers to multiple end-users are consolidated. Some deployments technologies even allow multiple levels of splitters.

Tree vs. star topology In the literature two different topologies are considered for both the fiber-optic and the copper subnetworks. While some models restrict the network topology to a star, others allow general trees. While models with star topologies (e.g., facility location, p-median and p-center) are often easier to solve, a tree topology allows a more detailed representation of the infrastructure like a street networks, trenches and ducts.

Capacitated vs. uncapacitated Models that consider capacities provide more detailed information but are usually much harder to solve. Also, the introduction of capacities often changes the solution topology and thus prohibits to use this additional information in solution algorithms. E.g., if it is known in advance that the solution has a tree structure this information can be used to derive better models than it is the case if the solution could contain cycles.

Several well-known problems have applications in telecommunication network design. If the decomposed modeling approach is chosen, star topologies can be modeled by the *Capacitated* or *Uncapacitated Facility Location problem* (CFLP, UFLP). There is wide range of literature on facility location problems and we refer the reader to a survey by Gourdin et al. [13].

In the following we discuss recent literature that has a strong focus on the application in the design of local access networks.

2.1 FTTH networks

If capacities are ignored, a common modeling approach for tree topologies is the *Steiner tree problem* (STP). For a general discussion of the STP we refer the reader to Du et al. [8].

Randazzo and Luna [26] consider a generalization of the STP that considers flow costs. Their model considers costs for cables as well as infrastructure like ducts and trenches. They present three different solution algorithms based on branch-and-bound, branch-and-cut and Benders decomposition. In a recent work Orlowski et al. [22] use an extended Steiner tree problem to give lower bounds on trenching costs for the installation of fiber-optic networks.

Li and Shen [16] consider the problem of optimally locating splitters in a greenfield deployment scenario. They use a continuous location problem to choose the location of one level of splitters. Their model takes into account fiber length limitations and splitter capacities.

Kim et al. [15] consider up to two levels of splitters and propose a heuristic to locate these splitters on a tree network. A more general approach is the one of Chardy et al. [5]. The authors consider the problem of locating splitters and routing fibers given the capacity limitations from existing infrastructure. Multiple levels of splitters are considered and the approach does not need additional assumptions on the structure of the underlying graph. The authors propose efficient preprocessing techniques to reduce the graph size and the number of customers. Their algorithm is based on a generalized flow problem. Chardy and Hervet [4] extend the approach of [5] by considering restrictions from operation, administration and maintenance of the resulting networks.

A different approach is the one of Gualandi et al. [14]. The authors consider an FTTH network that has a star-star topology. They allow to choose between several central offices and require that each end-user is connected to a cabinet which is itself connected to a central office. Their model is closely related to a two-level facility location problem. The proposed solution approaches are linear programming based rounding and local search algorithms.

2.2 FTTC and FTTB networks

A large amount of research has been done on the decomposed solution approach for planning FTTC and FTTB networks. For the first step, i.e., the location of cabinets and assignment of customers, there is a variety of models in the literature. In most applications the existing copper cable infrastructure can be used and facility location and p-median problems provide adequate models.

Wassermann [31] gives a detailed analysis of practical issues of the deployment of fiber-optic networks. The main focus from a modeling point of view is the location of cabinets and the assignment of users to these cabinets. Many additional restrictions, like specific technical limitations and budget constraints are considered. The author uses several variants of the p-median problem. Some of them exploit the structure of the underlying network infrastructure explicitly.

Once the location and capacities of cabinets are decided, the remaining problem can be modeled as the *Local Access Network Design problem* (LAN). The objective is to find a minimum cost routing of fibers from the central office to the terminals. The terminals can be cabinets for FTTC networks, basements for FTTB networks and end-users for FTTH networks. A thorough discussion of the LAN is provided in the recent work of Putz [24]. Other works that consider the LAN are by Salman et al. [28], Raghavan and Stanojević [25] and Ljubić et al. [20, 21].

If capacities are ignored, the integrated planning problem of FTTC and FTTB networks can be modeled as the *Connected Facility Location problem* (ConFL). This problem gained the interest of the scientific community much later than other problems that are common in the context of telecommunication network design. Early contributions were mainly from the theoretical computer science community. Various approximation algorithms with steadily improved approximation guarantees were developed, the most recent being due to Eisenbrand et al. [9]. Prior to this thesis there were only few contributions of the Operations Research community. Heuristic algorithms were proposed by Ljubić [17] and Tomazic and Ljubić [29], a dual-based local search was developed by Bardossy and Raghavan [3]. A detailed discussion of the literature on the ConFL and related problems is provided in Paper 1, which contains results of the master thesis of the author. It is included here for convenient reference, since that work provides part of the foundations for the developments constituting this PhD thesis.

2.3 Mixing FTTH and FTTC strategies

In its *Digital Agenda* the European Union recently announced the goal of providing half the European households with a bandwidth of at least 100 MBit/s and the other half with at least 30 Mbit/s by 2020 [10]. In the light of this policy mixed FTTH/FTTC networks seem to be a commendable alternative to pure FTTH or FTTC strategies. They enable telecommunication providers to obtain the required FTTH coverage but also ensure that the remaining households are not stuck with outdated copper connections that only provide a small bandwidth.

If the decomposed approach is chosen, a mixed strategy of FTTH and FTTC can be modeled as facility location problem together with the LAN. However, for the integrated approach the literature is scarce. Randazzo et al. [27] and Balakrishnan et al. [1, 2] consider a two-level network design problem that allows a tree topology for the fiber-optic as well as the copper subnetworks. However, all these approaches have the shortcoming of completely ignoring the significant costs occasioned by the installation of cabinets. The model of Gualandi et al. [14] allows to model mixed strategies but the assumption of a star-star topology puts strong limitations on the accuracy of the model. Further relevant literature for two level network design problems is discussed in Paper 3.

2.4 Models considered in this thesis

The models considered in the papers constituting the key part of this thesis extend the existing modeling approaches for fiber-optic local access networks in several ways.

ConFL models a fiber-optic network where the subnetworks that consist of copper have a star topology whereas the network connecting the installed facilities has a tree structure. Capacities are not considered in this model. It is suitable for designing networks if an existing copper infrastructure is used to connect end-users to cabinets.

In Paper 2 we consider a model for the integrated planning of FTTC and FTTB networks under additional reliability constraints. The usual tree topology of local access networks (e.g., ConFL) does not provide survivability as it is the case for common models for backbone networks. Local access networks based on such a survivable topology (e.g., mesh, two-vertex-connected or two-edge-connected) would incur significantly higher costs. Thus we improve the reliability of the local access network provided by a ConFL solution with the following additional restriction: We impose a limit on the number of edges that are allowed on the path from the central office to each installed facility. The problem obtained from this modeling approach was not studied in the literature before and we refer to it as the Hop Constrained Connected Facility Location problem (HC ConFL).

In Paper 3 we extend the modeling approaches for mixed FTTH and FTTC strategies described in the literature [1, 2, 14, 27]. We consider a two-level network design problem that explicitly models the costs for installing cabinets. A tree topology is assumed for both, the primary (fiber-optic) and secondary (copper) subnetworks. We refer to this model as the Two Level Network Design Problem with Transition Facilities (TLNDF).

In Paper 4 we introduce a model that extends the existing approaches in two ways. First, it extends the integrated modeling approach for FTTB and FTTC networks offered by ConFL by considering the capacities of multiplexors and splitters as well as the limited capacity of the core network that connects the chosen cabinets. It also combines the decomposed approaches for cabinet location and customer assignment [31] and the design of the core network interconnecting the cabinets [24]. Thereby this model offers greater flexibility than existing models. We refer to it as Capacitated Connected Facility Location problem (CapConFL).

Chapter 3

Contribution and scope

In this chapter we outline the main results of the papers collected in this thesis with respect to the models described in the preceding section. We state the contributions of each co-author to each of the papers and report the papers' current status with respect to publication.

Paper 1: MIP Models for Connected Facility Location: A Theoretical and Computational Study

This paper contains most results from the author's master thesis. It has been published in *Computers & Operations Research* [11]. As some of the results presented there are highly relevant for Papers 2 and 4 it is included in this dissertation.

The article comprises the first theoretical and computational study on mixed integer programming (MIP) models for the Connected Facility Location problem (ConFL). This problem combines facility location and Steiner trees: given a set of customers, a set of potential facility locations and some inter-connection nodes, ConFL searches for the minimum-cost way of assigning each customer to exactly one open facility, and connecting the open facilities via a Steiner tree. The costs needed for building the Steiner tree, facility opening costs and the assignment costs need to be minimized.

We propose several mixed integer programming models for ConFL, seven of which are compact and three of which are of exponential size. We also show how to transform ConFL into the Steiner arborescence problem by splitting the nodes that are potential facility locations. A full hierarchy of the models with respect to the bounds given by their respective linear programming relaxations is provided. For two models with an exponential number of constraints we develop a branch-and-cut algorithm. An extensive computational study is based on two benchmark sets of randomly generated instances with up to 1300 nodes and 115,000 edges. We empirically compare the presented models with respect to the quality of obtained bounds and the corresponding running time. We report optimal values for all but 16 instances for which the obtained gaps are below 0.6%.

Paper 2: Layered Graph Approaches to the Hop Constrained Connected Facility Location Problem

Preliminary results of this paper were published in the peer-reviewed conference proceedings of the International Symposium on Combinatorial Optimization (ISCO) 2010 [18]. The paper was accepted for publication in *INFORMS Journal on Computing* on December 8, 2011. It is available online in the Articles in Advance section since April 11, 2012 [19].

This article provides a first theoretical and computational study for HC ConFL. We propose two disaggregation techniques that enable the modeling of HC ConFL: (i) As a directed (asymmetric) ConFL on layered graphs, or (ii) as the Steiner arborescence problem (SA) on layered graphs. This allows us to use the best known mixed integer programming models for ConFL or SA to solve HC ConFL to optimality. In a polyhedral study we compare the obtained models with respect to the quality of their linear programming lower bounds. These models are finally computationally compared on a subset of the benchmark instances considered in Paper 1. We report optimal values for these instances with up to 1,300 nodes and 115,000 edges.

The layered graph models for HC ConFL were developed by Stefan Gollowitzer (SG) and Ivana Ljubić (IL) in equal shares. The branch-and-cut procedure was developed by IL. The computational experiments were carried out by SG. Writing of the paper was evenly split between SG and IL.

Paper 3: Enhanced Formulations and Branch-and-Cut for the Two Level Network Design Problem with Transition Facilities

Preliminary results of this paper were published in the peer-reviewed conference proceedings of the International Network Optimization Conference (INOC) 2011 [12]. The paper was submitted to the *European Journal of Operational Research* on November 23, 2011 and is currently under revision.

In this paper we first model the TLNDF on an extended graph where an additional set of arcs corresponds to the installation of facilities and propose a cut set based model that is defined on this extended graph. We present several theoretical results relating families of cut set inequalities on the extended graph with subfamilies of cut set inequalities on the original graph. We then show how a standard multi-commodity flow model defined on the original graph can be strengthened using disaggregation by technology. We prove that the disaggregated compact formulation on the original graph provides the same lower bound as the cut set formulation on the extended graph. We develop a branch-and-cut algorithm for solving the TLNDF. The performance of this algorithm is improved by separating subfamilies of cut set inequalities on the original graph. Our computational study confirms the efficiency and applicability of the new approach.

The cut set-based models, the extended graph formulation and the theoretical results were jointly developed by SG, IL and Luis Gouveia (LG). The flow models and corresponding theoretical results were obtained by SG. The branch-and-cut framework was implemented by SG. The computational study was planned by SG and IL and executed by SG. Writing of the paper was evenly split among SG, LG and IL.

Paper 4: Capacitated Network Design with Facility Location

This paper was submitted to *Computational Optimization and Applications* on July 27, 2012.

The Capacitated Connected Facility Location problem (CapConFL) combines the capacitated network design problem (CNDP) with the single-source capacitated facility location problem (SSCFLP). In this paper we first develop an integer programming formulation that uses the concept of single-commodity flows. Based on valid inequalities for the subproblems, CNDP and SSCFLP, we derive several (new) classes of valid inequalities for CapConFL. We use them in a branch-and-cut algorithm and show their applicability on a set of realistic benchmark instances.

The basic model was developed by Bernard Gendron (BG) and IL. The valid inequalities were developed by SG (65%) and IL (35%). The separation procedures and implementation of the branch-and-cut algorithm were done by SG. The computational experiments were jointly designed by SG and IL and executed by SG. Writing of the paper was evenly split among SG, BG and IL.

Chapter 4

Conclusions

In this thesis we proposed models for planning problems that arise in the cost-optimal design of fiber-optic local access networks. Based on the author's earlier work on the ConFL (Paper 1) we describe efficient exact solution algorithms for several network design problems. Compared to the existing literature these allow to model more reliable FTTC/FTTB networks based on existing copper infrastructure (Paper 2), mixed FTTC/FTTH strategies (Paper 3) and an integrated planning of FTTC/FTTB networks that considers the capacities of switchers, multiplexors and existing cable ducts (Paper 4).

Paper 1 provided an overview and comparison of different models for the ConFL. The models were assessed from a theoretical point of view by comparison of their linear programming bounds and their computational performance was compared on a set of large-scale benchmark instances. Two branch-and-cut algorithms, each based on a model with an exponential number of constraints, outperformed all compact models. This paper complements the existing literature on the ConFL that consists of approximation algorithms and heuristics by providing efficient exact algorithms for the ConFL.

The layered graph approach described in Paper 2 enabled us to use an arbitrary solution approach for ConFL to solve the more restricted problem with additional reliability constraints, HC ConFL. We compared several models, based on two different variants of the layered graphs and the two cut-based models from the paper on the ConFL. In addition, we provided preprocessing techniques that reduce the size of the layered graph. This allowed a faster solution of the more restricted HC ConFL, compared to ConFL, if the reliability constraints are tight.

In Paper 3 we showed how a multi-commodity flow formulation of the TLNDF can be disaggregated by technology such that the resulting formulation on the original graph provides the same linear programming bound as the cut set formulation on the extended graph. We also related cut set inequalities on the extended graph to cut set inequalities of different specific forms on the original graph. This allowed us to improve the computational performance of the cut set-based model by separating a subset of cut set inequalities on the original graph instead of the larger extended graph.

Finally, in Paper 4 we extended the Connected Facility Location problem by considering capacities of splitters, multiplexors and cable ducts. We combined valid inequalities for subproblems of the CapConFL and obtained valid inequalities that utilize both aspects of the problem, facility location and capacitated network design. Using these valid inequalities we strengthened a formulation based on single-commodity flows. Our computational results on a new set of realistic benchmark instances proved the efficiency of our approach.

As already indicated by Wassermann [31] the continuing improvement of wireless telecommunication technologies will raise interest in hybrid networks, partially based on, e.g., fiber-optic cable and partially based on wireless technologies. The allocation of frequencies to wireless access points, network robustness, the integration of mobile phone and other wireless networks are topics that do already or will soon raise the interest of the discrete optimization community.

Bibliography

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

[2] A. Balakrishnan, T. L. Magnanti, and P. Mirchandani. Modeling and heuristic worst-case performance analysis of the two-level network design problem. *Management Science*, 40(7):846–867, 1994.

[3] M. G. Bardossy and S. Raghavan. Dual-based local search for the connected facility location and related problems. *INFORMS Journal on Computing*, 22(4):584–602, 2010.

[4] M. Chardy and C. Hervet. Ftth network design under OA&M constraints. In Pahl et al. [23], pages 98–104.

[5] M. Chardy, M.-C. Costa, A. Faye, and M. Trampont. Optimizing splitter and fiber location in a multilevel optical ftth network. *European Journal of Operational Research*, 2012. ISSN 0377-2217. doi: 10.1016/j.ejor.2012.05.024.

[6] Der Standard. 1,5 Milliarden für Breitband-Internet. Online, July 3, 2009. URL `http://derstandard.at/1246541324152`.

[7] Deutsche Telekom. Medienmappe Breitband. Online, 2012. URL `http://www.telekom.com/breitband`.

[8] D.-Z. Du, B. Lu, H. Ngo, and P. M. Pardalos. Steiner tree problems. In *Encyclopedia of Optimization*, pages 3723–3736. 2009.

[9] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Approximating connected facility location problems via random facility sampling and core detouring. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1174–1183. SIAM, 2008.

[10] European Commission. Digital agenda for europe. Online, 05 2010. URL http://europa.eu/rapid/pressReleasesAction.do?reference=IP/10/581.

[11] S. Gollowitzer and I. Ljubić. MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research*, 38(2):435–449, 2011.

[12] S. Gollowitzer, L. Gouveia, and I. Ljubic. A node splitting technique for two level network design problems with transition nodes. In Pahl et al. [23], pages 57–70.

[13] E. Gourdin, M. Labbe, and H. Yaman. Telecommunication and location. In Z.Drezner and H. Hamacher, editors, *Facility Location: Applications and Theory*. Springer, 2002.

[14] S. Gualandi, F. Malucelli, and D. L. Sozzi. On the design of the next generation access networks. In A. Lodi, M. Milano, and P. Toth, editors, *CPAIOR*, volume 6140 of *Lecture Notes in Computer Science*, pages 162–175. Springer, 2010.

[15] Y. Kim, Y. Lee, and J. Han. A splitter location-allocation problem in designing fiber optic access networks. *European Journal of Operational Research*, 210(2):425–435, 2011.

[16] J. Li and G. Shen. Cost minimization planning for greenfield passive optical networks. *Journal of Optical Communications and Networking*, 1(1):17–29, Jun 2009. doi: 10.1364/JOCN.1.000017.

[17] I. Ljubić. A hybrid VNS for connected facility location. In T. Bartz-Beielstein, M. J. B. Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 157–169. Springer, 2007.

[18] I. Ljubić and S. Gollowitzer. Modeling the hop constrained connected facility location problem on layered graphs. In *Electronic Notes in Discrete Mathematics*, volume 36, pages 207–214, 2010. Proceedings of the International Symposium on Combinatorial Optimization (ISCO).

[19] I. Ljubić and S. Gollowitzer. Layered graph approaches to the hop constrained connected facility location problem. *INFORMS Journal on Computing*, April 11, 2012. doi: 10.1287/ijoc.1120.0500. Published online before print.

[20] I. Ljubić, P. Putz, and J. Salazar-González. A MIP-based heuristic approach to solve a prize-collecting local access network design problem. Technical report, University of Vienna, 2012.

[21] I. Ljubić, P. Putz, and J. Salazar-González. Exact approaches to the single-source network loading problem. *Networks*, 59(1):89–106, 2012.

[22] S. Orlowski, A. Werner, and R. Wessäly. Estimating trenching costs in fttx network planning. Technical Report ZIB-Report 11-49, Konrad-Zuse-Zentrum für Informationstechnik Berlin, Takustraße 7, D-14195 Berlin-Dahlem, Germany, October 2011.

[23] J. Pahl, T. Reiners, and S. Voß, editors. *Network Optimization - 5th International Conference, INOC 2011, Hamburg, Germany, June 13-16, 2011. Proceedings*, volume 6701 of *Lecture Notes in Computer Science*, 2011. Springer.

[24] P. Putz. *Fiber To The Home, Cost Optimal Design of Last-Mile Broadband Telecommunication Networks*. PhD thesis, University of Vienna, 2012.

[25] S. Raghavan and D. Stanojević. A note on search by objective relaxation. In *Telecommunications Planning: Innovations in Pricing, Network Design and Management*, volume 33 of *Operations Research/Computer Science Interfaces Series*, pages 181–201. Springer US, 2006.

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

[27] C. D. Randazzo, H. P. L. Luna, and P. Mahey. Benders decomposition for local access network design with two technologies. *Discrete Mathematics & Theoretical Computer Science*, 4(2):235–246, 2001.

[28] F. S. Salman, R. Ravi, and J. N. Hooker. Solving the capacitated local access network design problem. *INFORMS Journal on Computing*, 20 (2):243–254, 2008.

[29] A. Tomazic and I. Ljubić. A GRASP algorithm for the connected facility location problem. In *Proceedings of 2008 International Symposium on Applications and the Internet (SAINT)*, pages 257–260, 2008.

[30] United Nations General Assembly. Report of the special rapporteur on the promotion and protection of the right to freedom of opinion and expression, Frank La Rue. Online, 05 2011. URL http://www2.ohchr.org/english/bodies/hrcouncil/docs/17session/A.HRC.17.27_en.pdf.

[31] B. Wassermann. *Operations Research in action: a project for designing telecommunication access networks*. PhD thesis, University of Vienna, 2012.

[32] Wikipedia. Last mile. Online, 2012. URL http://en.wikipedia.org/wiki/Last_mile#Optical_fiber.

MIP Models for Connected Facility Location: A Theoretical and Computational Study

Stefan Gollowitzer, Ivana Ljubić[1]

Department of Statistics and Decision Support Systems,
Faculty of Business, Economics, and Statistics,
University of Vienna, Austria

Abstract

This article comprises the first theoretical and computational study on mixed integer programming (MIP) models for the connected facility location problem (ConFL). ConFL combines facility location and Steiner trees: given a set of customers, a set of potential facility locations and some inter-connection nodes, ConFL searches for the minimum-cost way of assigning each customer to exactly one open facility, and connecting the open facilities via a Steiner tree. The costs needed for building the Steiner tree, facility opening costs and the assignment costs need to be minimized.

We model ConFL using eight compact and two mixed integer programming formulations of exponential size. We also show how to transform ConFL into the Steiner arborescence problem. A full hierarchy between the models is provided. For the two exponential size models we develop a branch-and-cut algorithm. An extensive computational study is based on two benchmark sets of randomly generated instances with up to 1,300 nodes and 115,000 edges. We empirically compare the presented models with respect to the quality of obtained bounds and the corresponding running time. We report optimal values for all but 16 instances for which the obtained gaps are below 0.6%.

Keywords: Facility Location, Steiner Trees, Mixed Integer Programming Models, LP-relaxations

1. Preliminary Discussion

Improving the quality of broadband connections is nowadays one of the highest priorities of telecommunication companies. Solutions are sought that search for the optimal way of "pushing" rapid and high-capacity fiber-optic networks closer to the customers. Developing respective models and answering questions related to the design of "last-mile" networks defines a new challenging area of computer science and operations research. The *Connected Facility Location Problem* (ConFL) models the following telecommunication network design problem: Traditional wired local area networks require copper cable connections between end users. To reduce the signal loss, these lines are limited by a maximum distance. To increase the quality of internet communications, telecommunication companies decide to partially or completely replace the existing copper connection by

Email addresses: `stefan.gollowitzer@univie.ac.at` (Stefan Gollowitzer), `ivana.ljubic@univie.ac.at` (Ivana Ljubić)

[1]Supported by the Hertha-Firnberg Fellowship of the Austrian Science Foundation (FWF)

fiber-optic cables. In order to do so, different strategies, known as *fiber-to-the-home* (FTTH), *fiber-to-the-node* (FTTN), *fiber-to-the-curb* (FTTC) or *fiber-to-the-building* (FTTB), are applied.

ConFL models the FTTN / FTTC strategy: Fiber optic cables run to a cabinet serving a neighborhood. End users connect to this cabinet using the existing copper connections. Expensive switching devices are installed in these cabinets. The problem is to minimize the costs by determining positions of cabinets, deciding which customers to connect to them, and how to reconnect cabinets among each other and to the backbone.

1.1. What is Connected Facility Location? - Problem Definition

Gupta et al. [18] define the Connected Facility Location problem as follows: We are given a graph $G = (V, E)$ with a set of customers ($R \subseteq V$), a set of facilities ($F \subseteq V$) and a set of Steiner nodes ($\tilde{S} \subseteq V$) such that $\tilde{S} \cap F = \emptyset$. For all $e \in E$ we are given an edge cost $c_e \geq 0$ and for all $i \in F$ we are given facility opening costs $f_i \geq 0$. Then ConFL consists of finding an assignment of each customer to exactly one facility and connecting these facilities via a Steiner tree. Thereby, assignment costs $c_{ij}, i \in F, j \in R$ are given as the shortest path distance between i and j in G.
The overall costs in this problem are defined as $\sum_{j \in R} d_j c_{i(j)j} + \sum_{i \in \mathcal{F}} f_i + \sum_{e \in T} M c_e$, where $d_j \geq 1$ is demand of customer j, $i(j)$ denotes the facility serving j, $\mathcal{F}$ is the set of open facilities, T is the Steiner tree connecting open facilities and $M \geq 1$ is a constant.
Let $S = \tilde{S} \cup F$ denote the set of *core* nodes. We observe that without loss of generality we can assume that $S \cap R = \emptyset$. Otherwise, we only need to replace each node $u \in S \cap R$, with a pair of nodes, $u_1 \in S$ and $u_2 \in R$, connecting all $i \in S$, core neighbors of u, to u_1, and all $i \in F$, facility neighbors of u to u_2, without changing the edge/assignment costs. Finally, if $u \in F \cap R$, we need to connect customer neighbors to u_1 and add the service link $\{u_1, u_2\}$ into E, set its costs to zero and define $f_{u_1} = f_u$. We also observe that demands different from 1 can be set to 1 by adapting the respective assignment costs. We set $c_{ij} := d_j c_{ij} \quad \forall j \in R, \forall i \in F$ and reflect the demand in the cost structure implicitly [28]. Alternatively, we can make d_j copies of customer j, each with demand equal to one (see, e.g., [13]).
For the development of approximation algorithms there are two usual assumptions: The parameter M is used to distinguish between "cheap" assignment and "expensive" core network edges, and c is assumed to be a metric. As we will see later, both these assumptions are not necessary in our approaches. Therefore, we concentrate on a general cost structure.

Definition 1 (ConFL). For a given undirected graph (V, E) where $\{S, R\}$ is a disjoint partition of V with $R \subset V$ being the set of customers, $S \subset V$ the set of possible Steiner nodes and $F \subseteq S$ the set of facilities, edge costs $c_e \geq 0, e \in E$ and facility opening costs $f_i \geq 0, i \in F$, in the *Connected Facility Location* problem we search for a subset of open facilities such that:

- each customer is assigned to the closest open facility,
- a Steiner tree connects all open facilities, and
- the sum of assignment, facility opening and Steiner tree costs is minimized.

Optionally, a root $r \in F$ may be considered as an open facility always included in the network. In that case, we speak of the *rooted ConFL*. Obviously, every optimal ConFL solution will be a tree in which customers (and possibly the root r) are leaves. In the telecommunications field a "central office" connecting to the backbone network is often predefined and may be considered as a root node active in any feasible solution. Therefore, in the following we assume that the root is given in advance. In Section 3 we show how to solve unrooted instances.

The remainder of this paper is organized as follows: The following section will provide an exhaustive literature review on the topic. In Section 3 we propose ten mixed integer programming models for ConFL and we show a transformation of ConFL into the Steiner Arborescence (SA) problem. In Section 4 we provide a full hierarchy of the models based on the theoretical comparison of the quality of their lower bounds. Section 5 describes a branch-and-cut (B&C) framework that has been used to solve two formulations of exponential size. The computational results provided in Section 6 are conducted on two sets of benchmark instances introduced earlier in the literature.

2. Literature Review

The Connected Facility Location Problem has lately started to attract stronger interest in the scientific community. Compared to some closely related problem classes, there is just a small number of papers on the topic. A large share of publications about ConFL comes from the computer science community who present approximation algorithms of different kinds and qualities. The operations research community has developed a small number of heuristic methods. Preliminary results of one of our exact approaches have been published in [28].

Approximation Algorithms. A majority of the publications about ConFL concentrates on approximation algorithms. However, not a single one contains computational results. Thus, no conclusion can be drawn to the practical applicability of the described algorithms.

Karger and Minkoff [20] describe an adapted version of the Steiner tree problem. They consider the distribution of single data items from a root to a set of clients. It is not known beforehand which clients demand the data item in question. For each client, there is a known probability to become active and request data. Consider caching nodes at a certain cost, i.e. nodes storing the demanded data for resending it to clients becoming active later-on. The problem of finding a tree with minimal expected cost is equal to the Connected Facility Location Problem. The authors gather the clients into clusters connected to a common facility. Second, they connect these facilities by a Steiner tree. They present a bicriterion approximation algorithm producing a solution of at most 41 times the optimum cost.

Krick et al. [25] present a similar problem as the one in [20], although in an other context. They consider a computer network where clients (corresponding to customers) issue read and write requests. The data for the requests is stored in memory modules (facilities) at a certain cost. Read and write requests are served by the nearest installed memory module for the respective client. To keep data consistent throughout the network, all other memory modules are updated with the latest version. This requires connectivity between the memory modules. Krick et al. give a constant approximation algorithm with a larger constant than the one given by Karger and Minkoff [20].

In the context of reserving bandwidth for virtual private networks, Gupta et al. [18] introduce the term Connected Facility Location. They give a proof for ConFL to be NP-hard. They present a first cut-based integer programming formulation. Their formulation will be described and discussed in detail in Section 3.2. Their approximation algorithm for ConFL has a constant factor of 10.66. For the closely related *rent-or-buy problem* (RoB), in which all nodes are potential facilities with opening costs equal to 0, the algorithm gives an approximation factor of 9.002.

Swamy and Kumar [39] develop a primal-dual approximation algorithm for ConFL, RoB and k-ConFL. The latter comprises the additional restriction that in an optimum solution at most k facilities can be opened. The integer programming formulation used is the same as in Gupta et al. [18]. As results the authors give approximation ratios of 8.55, 4.55 and 15.55 for ConFL, RoB and k-ConFL, respectively.

The approximation factors have been successively improved in Jung et al. [19] and Williamson and van Zuylen [41]. Finally, Eisenbrand et al. [13] combine approximation algorithms for the basic facility location problem and the connectivity problem of the opened facilities by running a what they call *core detouring scheme*. The randomised version of the approximation algorithm gives new best expected approximation ratios for ConFL (4.00), RoB (2.92) and k-ConFL (6.85). The ratios for the de-randomised version are 4.23, 3.28 and 6.98 respectively.

Heuristics and Exact Methods. Ljubić [28] describes a hybrid heuristic combining Variable Neighborhood Search with a reactive tabu search method. The author compares it with an exact branch-and-cut approach. The corresponding integer programming model for the branch-and-cut approach will be explained in detail and compared to other formulations in Section 3. Ljubić [28] also presents two classes of test instances as a result of combining Steiner tree and uncapacitated facility location instances. Results for these instances with up to 1300 nodes are presented.

Tomazic and Ljubić [40] present a Greedy Randomized Adaptive Search Procedure (GRASP) for the ConFL problem. Results for a new set of test instances with up to 120 nodes (facilities plus customers) are presented.

2.1. Related Problems

The Connected Facility Location problem is a combination of two other well-known problems in graph theory. These are the Steiner tree problem (STP) and the Uncapacitated Facility Location problem (UFL). ConFL contains them both as special cases. For a set of possible facility locations connected to a root via a star, we have UFL. In case each customer can only be served by one predefined facility, we know the set of facilities that needs to be opened in advance. Thus, we then have an STP to solve.

Rent-or-buy Problem (RoB). The rent-or-buy problem is often viewed as a special case of the ConFL problem. In the RoB problem facility opening costs are 0 and facilities can be opened anywhere. Thus, also customer nodes can act as facilities and have other customers assigned to them. The cost for each edge in a solution to the RoB depends on its adjacent nodes. If an edge is used to assign a customer to a facility, only assignment costs are incurred. If an edge connects two facilities, a comparatively higher cost, i.e. M times the assignment cost, has to be paid for.

The (general) Steiner tree-star problem ((G)STS). The Steiner tree-star problem was introduced by Lee et al. [26]. It arises in the design of some specific telecommunication networks, where bridging occurs. The Steiner tree-star problem is the following: Given a graph with disjoint sets of possible facility nodes and customers, we want to find a minimum cost tree such that each customer is assigned to a facility and that all open facilities are connected by a Steiner tree. Facility opening costs are incurred for any facility in the solution tree, regardless of whether any customers are assigned to it or not.
Exact methods to solve the STS problem have been described by Lee et al. [26, 27], a tabu search based heuristic was developed by Xu et al. [43]. Khuller and Zhu [21] introduced the *general*

Steiner tree-star problem. There, the sets of possible facilities and customers must not be disjoint. Nodes can act in both ways and an open facility can serve the customer in its own place at no additional cost. Khuller and Zhu [21] derive two approximation algorithms for the general STS with approximation factors of 5.16 and 5 respectively.

General Connected Facility Location (GConFL). Bardossy and Raghavan [4] develop a dual-based local search (DLS) heuristic for a family of problems combining facility location decisions with connectivity requirements, namely the (general) Steiner tree-star, ConFL and RoB. They introduce the general ConFL problem, into which any of the aforementioned 4 problem classes can be transformed. The presented DLS heuristic works in two phases. After applying dual-ascent in order to get a lower and upper bound in the first phase, in the second phase a local search procedure is carried out on the facilities and Steiner nodes selected before. Computational results for instances with up to 100 nodes are presented. Running time and the quality of solutions of Ljubić' VNS heuristic and DLS are compared for the set of instances introduced in [28].

Tree of Hubs Location problem (THLP). Another related problem with a tree-star topology is the tree of hubs location problem proposed by Contreras et al. [10]. This is a network hub location problem with single assignment in which a fixed number of hubs needs to be located, with an additional requirement that the hubs are connected by means of a tree. The sum of costs for routing the flow between each pair of source-destination nodes is minimized. In [10] the authors propose a compact MIP model, a number of valid inequalities and present computational results for instances with up to 25 nodes. A tighter formulation, a bounding heuristic and a Lagrangian relaxation approach are presented in [9]. The new approach solves instances with up to 100 nodes.

3. MIP Formulations for ConFL

It is well known that the MIP formulations for optimization problems with tree topology provide stronger lower bounds when defined on directed graphs (see, e.g., [8, 16, 32]). In this section we will first describe how to transform undirected instances for ConFL into directed ones. A range of MIP formulations for the ConFL will be presented afterwards. As the exponential size formulations are hard to implement by means of a modeling language, various compact MIP formulations will be described in this section as well. They are either flow formulations or based on sub-tour elimination constraints.

3.1. Transformation Into Directed Graphs

Throughout this paper, an arc from i towards j will be denoted by ij, and the corresponding undirected edge by $\{i, j\}$. Let (V, E) be a given instance of ConFL with $\{S, R\}$ being a partition of V and $F \subseteq S$. This instance can be transformed into a bidirected instance (V, A) as follows (cf. [40]):

- Replace core edges $e \in E$ with $e = \{i, j\}, i, j \in S$ by two directed arcs $ij \in A$ and $ji \in A$ with cost $c_{ij} = c_{ji} = c_e$. Since we are modelling an arborescence directed away from the root node, edges $\{r, j\}$ are replaced by a single arc rj only.
- Replace assignment edges $e \in E$ with $e = \{j, k\}, j \in F, k \in R$ by an arc $jk \in A$ with cost $c_{jk} = c_e$ respectively.

Rooting Unrooted Instances. To obtain an optimal solution for a directed, unrooted instance (V, A) by solving a model for rooted instances we adapt the input instance and the corresponding model as follows:

- Expand the set of facilities F by adding an artificial root r to $V' = V \cup \{r\}$ with cost $f_r = 0$.
- Expand the set of arcs by adding an arc rj for all core nodes $j \in F$ with $c_{rj} = 0$.
- Limit the number of arcs emanating from the root r to 1.

In the remainder of this paper we will refer to the Connected Facility Location problem on directed graphs as the following:

Definition 2 (ConFL on directed graphs)**.** We are given a directed graph (V, A) with edge costs $c_{ij} \geq 0, ij \in A$, facility opening costs $f_i \geq 0, i \in F$ and a disjoint partition $\{S, R\}$ of V with $R \subset V$ being the set of customers, $S \subset V$ the set of possible Steiner tree nodes, $F \subset S$ the set of facilities, and the root node $r \in F$. Find a subset of open facilities such that

- each customer is assigned to exactly one open facility,
- a *Steiner arborescence* rooted in r connects all open facilities, and
- the cost defined as the sum of assignment, facility opening and Steiner arborescence cost, is minimized.

To model the problem, we will use the following binary variables:

$$x_{ij} = \begin{cases} 1, & \text{if } ij \text{ belongs to the solution} \\ 0, & \text{otherwise} \end{cases} \quad \forall ij \in A \qquad z_i = \begin{cases} 1, & \text{if } i \text{ is open} \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in F$$

We will use the following notation: $A_R = \{ij \in A \mid i \in F, j \in R\}$, $A_S = \{ij \in A \mid i, j \in S\}$. Furthermore, for any $W \subset V$ we denote by $\delta^-(W) = \{ij \in A \mid i \notin W, j \in W\}$ and $\delta^+(W) = \{ij \in A \mid i \in W, j \notin W\}$.

3.2. Cut-Based Formulations

There are two different formulations of exponential size for ConFL given in the literature. They are both based on cut sets and differ in strength.

Cut Set Formulation of Gupta et al. [18]. Gupta et al. [18] first introduced an undirected ILP formulation for ConFL. To ensure comparability, a directed version will be presented here. One might think of any ConFL solution as a Steiner arborescence rooted at r with customers as leaves and with node weights that need to be payed for any node that is adjacent to a customer. Therefore, instead of requiring connectivity among open facilities and assignment of customers to open facilities, we are going to ask for the solution that ensures a directed path between r and any customer $j \in R$, using the arcs from A.

The cut-based model reads then as follows:

$$
\begin{aligned}
(CUT_R) \quad & \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i \\
\text{s.t.} \quad & \sum_{uv \in \delta^-(U)} x_{uv} \geq \sum_{j \in U: jk \in A_R} x_{jk} && \forall U \subseteq S \setminus \{r\},\ U \cap F \neq \emptyset,\ \forall k \in R && (1) \\
& \sum_{jk \in A_R} x_{jk} = 1 && \forall k \in R && (2) \\
& x_{jk} \leq z_j && \forall jk \in A_R && (3) \\
& z_r = 1 && && (4) \\
& x_{ij} \in \{0,1\} && \forall ij \in A && (5) \\
& z_i \in \{0,1\} && \forall i \in F && (6)
\end{aligned}
$$

The objective comprises the cost for the Steiner arborescence ($\sum_{ij \in A_S} x_{ij} c_{ij}$), the cost to connect customers to facilities (that we also refer to as *assignment cost*, i.e. $\sum_{ij \in A_R} x_{ij} c_{ij}$) and the facility opening cost ($\sum_{i \in F} z_i f_i$). Constraints (2) ensure that every customer is connected to at least one facility, constraints (3) ensure that each facility is opened if customers are assigned to it, equation (4) defines the root node. Inequalities (1) represent the set of cuts. For every subset $U \subseteq S \setminus \{r\}$ and for each customer $k \in R$, an open arc from a facility in U toward j, necessitates a directed path from r towards U. Constraints (2) can be replaced by inequality in case that $c_{ij} \geq 0$, for all $ij \in A_R$. Furthermore, the same optimization problem with continuous assignment variables x_{ij}, for all $ij \in A_R$, returns an optimal ConFL solution. This is because the underlying assignment matrix is totally unimodular, whenever z_i values are fixed to zero or one.

Observation 1. *Using equations* (2)*, we can re-write constraints* (1) *as follows:*

$$\sum_{uv \in \delta^-(U)} x_{uv} + \sum_{jk \in A_R : j \notin U} x_{jk} \geq 1, \quad \forall U \subseteq S \setminus \{r\},\ U \cap F \neq \emptyset \quad \forall k \in R. \tag{7}$$

Denote by $W = S \setminus U$, and let $A_S^W := \delta^+(W) \cap A_S$ and $A_R^W = \delta^+(W) \cap A_R$. Now, we can interpret these constraints as follows: every cut separating customer k from r (involving all arcs from $A_S \cup A_R$) has to be greater than or equal to one, i.e.:

$$\sum_{uv \in A_S^W} x_{uv} + \sum_{jk \in A_R^W} x_{jk} \geq 1, \quad \forall W \subseteq S,\ r \in W,\ W \cap F \neq F,\ \forall k \in R.$$

Figure 1 illustrates an example of these cut set inequalities.

According to the result of Swamy and Kumar [39], the integrality gap of the LP-relaxation of (CUT_R) is not greater than 8.55, if c is a metric, and core costs are M times more expensive than the assignment costs ($M \geq 1$).

Ljubić' Cut Set Formulation. Ljubić [28] presents a slightly different formulation where the cuts are defined according to the open facilities:

$$
\begin{aligned}
(CUT_F) \quad & \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i \\
\text{s.t.} \quad & \sum_{uv \in \delta^-(W)} x_{uv} \geq z_i && \forall W \subseteq S \setminus \{r\},\ \forall i \in W \cap F \neq \emptyset && (8)
\end{aligned}
$$

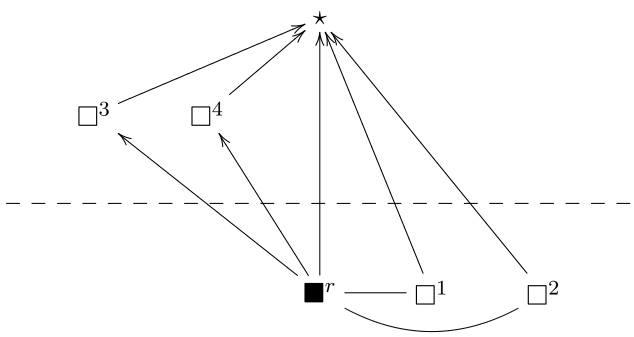


Figure 1: Graphic illustration for cut inequalities (2). $W = \{r, 1, 2\}$, $U = \{3, 4\}$

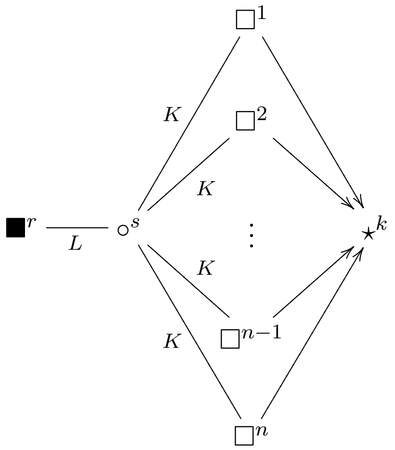


Figure 2: In this example the cost structure is as follows: all facility opening and assignment costs are 1. $c_{rs} = L$ and $c_{si} = K$, for all $i \in \{1, \dots, n\}$.

(2) - (6)

Lemma 1. *There are instances for which the values of the LP-relaxation of the CUT_F model can be as bad as $\frac{1}{|F|-1} OPT$, where OPT denotes the integer solution value.*

Proof. Figure 2 illustrates such a situation. In this example $n := |F| - 1$. The optimal solution value for the LP relaxation of CUT_F is $v_{LP}(CUT_F) = \frac{L}{n} + K + 3$ and the optimal integer solution value is $OPT = L + K + 3$. For $K >> L$, we get $\frac{v_{LP}(CUT_F)}{OPT} \approx \frac{1}{n}$. $\square$

3.3. Flow-Based Formulations

Extending flow formulations for the (prize-collecting) Steiner tree problem (see, e.g., [29, 38]), several ways to model ConFL as a flow problem are possible. One option is to have a flow from the root to each customer. Alternatively, flow can be allowed from the root node to open facilities only, with additional constraints ensuring customers to be assigned to an open facility. Further it is possible to consider just one single commodity or separate commodities for each customer or facility respectively.
In the following we propose six different flow formulations for ConFL. The strength of the different formulations is discussed later in Section 4.

Single-Commodity Flow Between Root and Facilities. This single commodity-flow formulation with flow between root node and facilities is an extension of the single-commodity flow formulation for the prize-collecting Steiner tree problem (see, e.g., Ljubić [29]). The amount of flow terminating in a facility is linked to the variable indicating whether the facility is open or not. For all $ij \in A_S$, continuous variable g_{ij} denotes the amount of flow that is simultaneously routed from r toward all open facilities over arc ij.

$$(SCF_F) \quad \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i$$

$$\text{s.t.} \quad \sum_{ji \in A_S} g_{ji} - \sum_{ij \in A_S} g_{ij} = \begin{cases} z_k & i = k,\ k \in F \\ -\sum_{k \in F} z_k & i = r \\ 0 & i \in S \setminus \{F\} \end{cases} \quad \forall i \in S \tag{9}$$

$$0 \leq g_{ij} \leq (|F| - 1) \cdot x_{ij} \quad \forall ij \in A_S \tag{10}$$

$$(2) - (6)$$

Constraints (9) ensure that each facility $j \in F$ receives z_j units of flow from the root. The coupling constraints (10) ensure that on every arc ij, there is enough capacity to simultaneously route that flow. They also force an arc ij to be installed if there is a flow sent through it. Model SCF_F comprises $O(|A|)$ constraints and $O(|A|)$ binary and continuous variables.

The following result is due to the usage of "big-M" constraints in (10):

Lemma 2. *There are instances for which*

a) the values of the LP-relaxation of the SCF_F model can be as bad as $\frac{1}{|F|-1} OPT$, and

b) the ratio $\frac{v_{LP}(SCF_F)}{v_{LP}(CUT_F)} \approx \frac{1}{|F|}$.

Proof. a) The example given in Figure 2 provides $v_{LP}(SCF_F) = \frac{L}{n} + \frac{K}{n} + 3$ which gives ratio $\frac{v_{LP}(SCF_F)}{OPT} \approx \frac{1}{|F|}$.

b) If $K >> L$ in the same example, we obtain $\frac{v_{LP}(SCF_F)}{v_{LP}(CUT_F)} = \frac{\frac{L}{n} + \frac{K}{n} + 3}{\frac{L}{n} + K + 3} = \frac{1}{|F|-1} \approx \frac{1}{|F|}$.

□

Single-Commodity Flow between Root and Customers. We now consider single commodity-flow from the root node to each of the customers. At the expense of more flow variables this allows us to drop constraints (2) used in SCF_F:

$$(SCF_R) \quad \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i$$

$$\text{s.t.} \quad \sum_{ji \in A_S} f_{ji} - \sum_{ij \in A} f_{ij} = \begin{cases} 1 & i \in R \\ -|R| & i = r \\ 0 & i \in S \setminus \{r\} \end{cases} \quad \forall i \in V \tag{11}$$

$$0 \leq f_{ij} \leq |R| \cdot x_{ij} \quad \forall ij \in A \tag{12}$$

$$(3) - (6)$$

Constraints (11) ensure that each customer receives one unit of flow from the root node and constraints (12) are similar to (10). However, one easily observes that, although redundant for the MIP formulation, assignment constraints (2) can strengthen the quality of lower bounds. We denote by SCF_R^+ the formulation SCF_R extended by (2). Models SCF_R and SCF_R^+ comprise $O(|A|)$ constraints and $O(|A|)$ binary variables.

Lemma 3. *There are instances for which*

a) the values of the LP-relaxation of the SCF_R *(*SCF_R^+*) model can be as bad as* $\frac{1}{|R|}OPT$*, and*

b) the ratio $\frac{v_{LP}(SCF_R)}{v_{LP}(CUT_R)} \approx \frac{1}{|R|}$.

Multi-Commodity Flow with One Commodity per Facility. The two flow formulations presented above can be improved by disaggregation of commodities.

Choosing one commodity per facility, each variable indicating an open facility is linked to a distinct commodity. A multi-commodity flow formulation with one commodity per facility is given by:

$$(MCF_F) \quad \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i$$

$$\text{s.t.} \quad \sum_{ji \in A_S} g_{ji}^k - \sum_{ij \in A_S} g_{ij}^k = \begin{cases} z_k & i = k \\ -z_k & i = r \\ 0 & i \neq k, r \end{cases} \quad \forall i \in S,\ \forall k \in F \tag{13}$$

$$0 \leq g_{ij}^k \leq x_{ij} \qquad \forall ij \in A_S,\ \forall k \in F \tag{14}$$

$$(2) \text{ - } (6)$$

Equations (13) are the flow preservation constraints defining the flow from the root node to each facility. These constraints ensure the existence of a connected path from r to every open facility. The stronger coupling constraints ensure that the arc is open if a flow is sent through it. Formulation MCF_F comprises $O(|A_S||F| + |A_R|)$ constraints, $O(|A_S||F|)$ continuous and $O(|A|)$ binary variables.

Multi-Commodity Flow with One Commodity per Customer. Another choice for the commodities we use, is the set of customers. Assigning a commodity of size 1 to each customer allows to remove the $\mathbf{z}$ variables from the flow preservation constraints. Using one commodity per customer, ConFL can be stated as:

$$(MCF_R) \quad \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i$$

$$\text{s.t.} \quad \sum_{ji \in A} f_{ji}^k - \sum_{ij \in A} f_{ij}^k = \begin{cases} 1 & i = k \\ -1 & i = r \\ 0 & i \neq k, r \end{cases} \quad \forall i \in V,\ \forall k \in R \tag{15}$$

$$0 \leq f_{ij}^k \leq x_{ij} \qquad \forall ij \in A,\ \forall k \in R \tag{16}$$

$$(3) \text{ - } (6)$$

Formulation MCF_R comprises $O(|A||R|)$ constraints, $O(|A||R|)$ continuous and $O(|A|)$ binary variables.

Observation 2. *Variables* x_{ij}*,* $ij \in A_R$*, are redundant in this formulation, as every LP-optimal solution of* MCF_R *also satisfies*

$$f_{jk}^l = \begin{cases} x_{jk}, & \text{if } l = k \\ 0, & \text{otherwise} \end{cases} \qquad \forall l \in R,\ \forall jk \in A_R.$$

Therefore, constraints (2) are redundant, for both, the MCF_R model and its LP-relaxation. However, we keep variables $x_{ij}, ij \in A_R$ in this model for better readability.

3.3.1. Strong Formulations Comprising Common Flow Variables

Polzin and Daneshmand [38] have developed a formulation which they call *Common Flow* formulation for the Steiner arborescence problem. It is based on a disaggregation of multi commodity-flow formulation with additional 4-index variables. These variables indicate the common flow from the root towards any pair of terminals. For ConFL this gives two choices on the common flows considered, towards facilities or towards customers. The variant, in which common flows towards facilities are considered, is an extension of MCF_F, the other one is an augmentation of MCF_R and it is the strongest one among all formulations presented in this paper (see Section 4).

Common Flow Between Root and Facilities. Let $\bar{g}_{ij}^{kl}$ denote the common flow towards facilities k and l, $k, l \in F, k \neq l$, over an arc ij. Then a MIP formulation of ConFL using common flows from the root to facilities is given by:

$$(CF_F) \quad \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i$$

$$\text{s.t.} \quad \sum_{ji \in A_S} g_{ji}^k - \sum_{ij \in A_S} g_{ij}^k = \begin{cases} z_k & i = k \\ -z_k & i = r \\ 0 & i \neq k, r \end{cases} \quad \forall i \in S,\ \forall k \in F \tag{17}$$

$$\sum_{ij \in A_S} \bar{g}_{ij}^{kl} - \sum_{ji \in A_S} \bar{g}_{ji}^{kl} \leq \begin{cases} \min(z_k, z_l) & i = r \\ 0 & i \neq r \end{cases} \quad \forall i \in S,\ \forall k, l \in F,\ k \neq l \tag{18}$$

$$0 \leq \bar{g}_{ij}^{kl} \leq \min(g_{ij}^k, g_{ij}^l) \quad \forall ij \in A_S,\ \forall k, l \in F,\ k \neq l \tag{19}$$

$$0 \leq g_{ij}^k + g_{ij}^l - \bar{g}_{ij}^{kl} \leq x_{ij} \quad \forall ij \in A_S,\ \forall k, l \in F,\ k \neq l \tag{20}$$

$$(2) - (6)$$

Constraints (17) are flow preservation constraints as in MCF_F. Constraints (18) ensure that the common flow from the root toward facilities k and l is non-increasing. Inequalities (19) define the relation between common flow and commodity flow variables. The coupling constraints (20) ensure that the arc is installed whenever there is a flow sent through it. Inequalities (18) and (19) are written in a compact way: *min* indicates that each of them is to be replaced by two constraints with either of the min-arguments on the right hand side.
Formulation CF_F comprises $O(|A_S||F|^2)$ constraints, $O(|A_S||F|^2)$ continuous and $O(|A|)$ binary variables.

Common Flow Between Root and Customers. Starting from the MCF_R model, we can now derive the other common flow formulation. Let $\bar{f}_{ij}^{kl}$ denote the common flow towards customers k and l,

$k \neq l$. Then the common flow formulation with flows from the root to customers is given by:

$$(CF_R) \quad \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i$$

$$\text{s.t.} \quad \sum_{ji \in A} f_{ji}^k - \sum_{ij \in A} f_{ij}^k = \begin{cases} 1 & i = k \\ -1 & i = r \\ 0 & i \neq k, r \end{cases} \quad \forall k \in R \tag{21}$$

$$\sum_{ij \in A_S} \bar{f}_{ij}^{kl} - \sum_{ji \in A_S} \bar{f}_{ji}^{kl} \leq \begin{cases} 1 & i = r \\ 0 & i \neq r \end{cases} \quad \forall i \in S,\ \forall k, l \in R,\ k \neq l \tag{22}$$

$$0 \leq \bar{f}_{ij}^{kl} \leq \min(f_{ij}^k, f_{ij}^l) \quad \forall ij \in A,\ \forall k, l \in R,\ k \neq l \tag{23}$$

$$0 \leq f_{ij}^k + f_{ij}^l - \bar{f}_{ij}^{kl} \leq x_{ij} \quad \forall ij \in A,\ \forall k, l \in R,\ k \neq l \tag{24}$$

$$(3) - (6)$$

Constraints (21) are flow preservation constraints as in MCF_R. Inequalities (22) ensure that the common flow from the root to customers k and l is non-increasing. Constraints (23)-(24) are equivalents of (19) - (20). In (23), *min* again indicates that the corresponding inequalities are to be replaced by ones with either of the arguments on the right hand side.
Formulation CF_R comprises $O(|A||R|^2)$ constraints, $O(|A||R|^2)$ continuous and $O(|A|)$ binary variables.

3.4. Formulations Based on Sub-tour Elimination Constraints

Another well-studied group of MIP formulations for problems on graphs are based on sub-tour elimination. We present here one compact and one exponential size model.

Miller-Tucker-Zemlin Formulation. One very simple strategy for sub-tour elimination was proposed by Miller, Tucker and Zemlin [35] and has been applied to a number of problems, including (Asymmetric) Traveling Salesman, Vehicle Routing, Minimum Spanning Tree and Steiner Tree Problem [11, 12, 17, 36]. In addition to x and z variables, we now introduce *level variables* $u_i \geq 0$, for all $i \in S$, determining the level of node i in the tree solution. The root node is assigned to the level zero.

Using the lifted Miller-Tucker-Zemlin (MTZ) constraints (see, e.g., [11]), ConFL can be stated as:

$$(MTZ) \quad \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i$$

$$\text{s.t.} \quad \sum_{i \in S \setminus \{k\}} x_{ij} \geq x_{jk} \quad \forall j \in S \setminus \{r\},\ \forall k \in V \tag{25}$$

$$|S| \cdot x_{ij} + u_i \leq u_j + |S| - 1 \quad \forall ij \in A_S \tag{26}$$

$$u_r = 0 \tag{27}$$

$$u_i \geq 0 \quad \forall i \in S \setminus \{r\} \tag{28}$$

$$(2) \text{ - } (6)$$

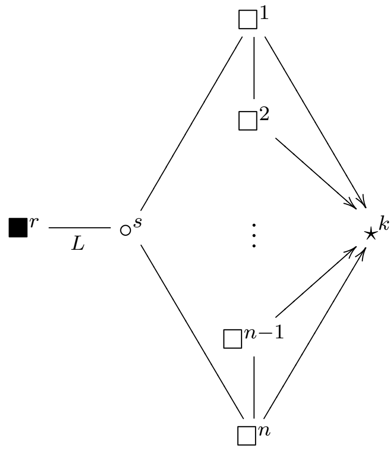


Figure 3: In this example $n := |F| - 1$. The cost structure is as follows: all facility opening, arc opening and assignment costs are 1, except for $c_{rs} = L$, where $L \gg 0$ is an arbitrarily large number.

Constraints (25) limit the out-degree of a node by its in-degree. Constraints (26) are Miller-Tucker-Zemlin sub-tour elimination constraints, setting the difference $u_j - u_i$ for an open arc ij to at least 1, thereby eliminating cycles in the Steiner tree connecting the facilities. Constraint (27) sets the level of the root node to zero.

Formulation *MTZ* comprises $O(|A|)$ constraints, $O(|S|)$ continuous and $O(|A|)$ binary variables. The formulation is small in the number of constraints and variables, compared to the aforementioned formulations based on flows or cut sets. The quality of the lower bounds, i.e. the strength of the formulations will be analyzed in the subsequent section.

Lemma 4. *The values of the LP-relaxation of the MTZ model can be arbitrarily bad.*

Proof. Consider the example in Figure 3: The LP-solution opens each facility with $1/n$, and builds one directed cycle of $\{s\} \cup \{1, \dots, n\}$ where for each arc ij in the cycle $x_{ij} = 1/n$. It assigns $v_{LP}(MTZ) = 4 + \frac{1}{n}$ and $OPT = L + 4$, which gives ratio $\frac{v_{LP}(MTZ)}{OPT} \approx \frac{1}{L}$. $\square$

Note that for our computational experiments we replaced constraints (26) by the following stronger ones:

$$(|S| - 2) \cdot x_{ji} + |S| \cdot x_{ij} + u_i \leq u_j + |S| - 1 \qquad \forall ij \in A_S$$

The polyhedral results in Section 4 are for the weaker model.

Formulation Based on Generalized Sub-tour Elimination Constraints. To model the Steiner tree in the core network, one might consider another formulation extended by the following node variables:

$$w_i = \begin{cases} 1, & \text{if } i \text{ belongs to the solution,} \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in S$$

Such a model has been used for the node-weighted Steiner tree problems (see, e.g., [15, 31, 32]).

$$(GSEC) \quad \min \sum_{ij \in A} x_{ij} c_{ij} + \sum_{i \in F} z_i f_i$$

$$\sum_{uv \in A: u,v \in U} x_{uv} \leq \sum_{i \in U \setminus \{k\}} w_i \quad \forall U \subset S, \forall k \in U \tag{29}$$

$$\sum_{uv \in A} x_{uv} = \sum_{i \in S \setminus \{r\}} w_i \tag{30}$$

$$w_i \geq z_i \quad \forall i \in F \tag{31}$$

$$0 \leq w_i \leq 1 \quad \forall i \in S \tag{32}$$

$$(2) - (6)$$

Equality (30) ensures that the set of edges is equal to the number of selected nodes minus one. In order to ensure the tree structure, sub-tours are eliminated by deploying constraints (29). Since facility nodes can also be used only as Steiner nodes, in which case $w_i = 1$ and $z_i = 0$, inequalities (31) must hold.

We will see in the following section that the results known for Steiner trees with respect to *GSEC*, directly apply to ConFL.

4. Polyhedral Comparison

In this section we provide a theoretical comparison of the MIP models described above with respect to optimal values of their LP-relaxations. The examples given below are used in the proofs of this section. These examples employ the following notation:

$\blacksquare$ represents the root node,	$\circ$ represents a Steiner node,
$\square^l$ represents a facility with label l,	$\star$ represents a customer.

Arc costs different from 1 are displayed next to the respective arc. Facility opening, assignment and core costs are all 1 in all examples, unless stated differently. All the values of facility node variables stated in the descriptions below refer to optimal LP solutions. The core network is presented as undirected graph, except in Figure 6.

	Fig. 4	Fig. 5	Fig. 6	Fig. 7	Fig. 8
MTZ	16	18	20	9	10
SCF_F	11	$14\frac{3}{8}$	$14\frac{1}{5}$	16	8
SCF_R	$7\frac{1}{4}$	$18\frac{1}{8}$	7	$17\frac{1}{4}$	$3\frac{1}{4}$
SCF_R^+	11	$22\frac{1}{4}$	$14\frac{1}{5}$	21	7
MCF_F	16	18	22	26	10
MCF_R	16	28	22	26	10
CF_F	16	18	24	26	10
CF_R	16	28	24	26	10

Table 1: Optimal LP solutions for examples in Figures 4 - 8

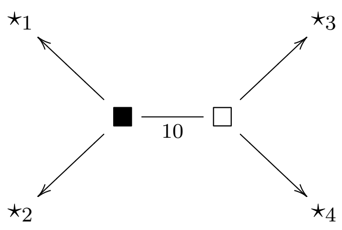


Figure 4: This simple example demonstrates the weakness of formulation SCF_R. The facility node variable is 1/4 for SCF_R and 1 for all other models.

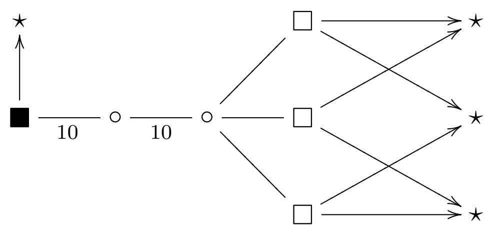


Figure 5: This example is a small variant of the one in Figure 2. It will show the weakness of models where the flows are only defined on the core subgraph A_S. Facility node variables are 1/8 for SCF_R and 1/2 for all other models.

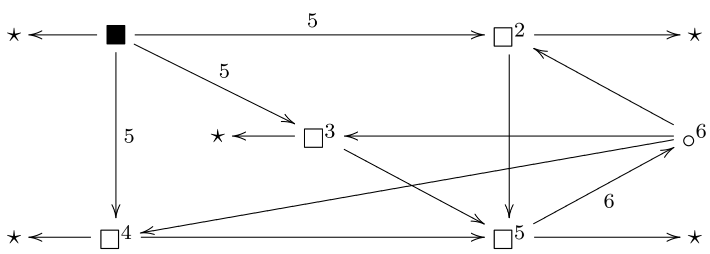


Figure 6: In this example the core network is directed and there is exactly one customer that can be assigned to each facility. Thus, every facility needs to be open in a feasible solution. Facility node variables are 1/5 for SCF_R and 1 for all other models. A version of this example was described by Polzin and Daneshmand [38].

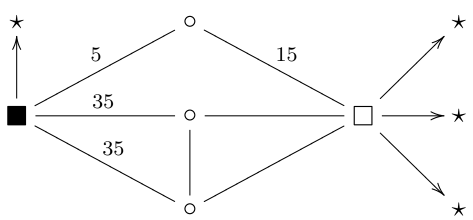


Figure 7: This example demonstrates the weakness of Miller-Tucker-Zemlin constraints. The facility node variable is 1/4 for SCF_R and 1 for all other models. In the LP solution for model MTZ there is a cycle consisting of the arcs of weight 1. The open facility is not connected to the root.

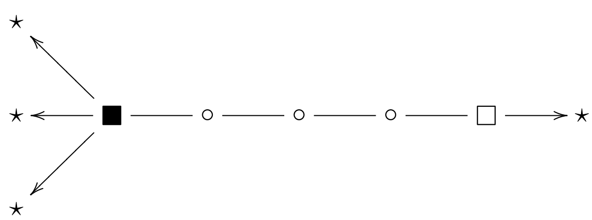


Figure 8: This example demonstrates the weakness of "big-M" constraints in the models comprising single commodity flow. The facility node variable is 1/4 for SCF_R and 1 for all other models.

Let $\upsilon_{LP}(.)$ denote the optimal solution value of the LP relaxation of a given model. By comparing the optimal LP solution values for the aforementioned examples, provided by the models in Section 3, we can state the following result.

Lemma 5. *The following pairs of formulations are incomparable with respect to the quality of lower bounds:*

a) MTZ and SCF_F,	*d)* SCF_R (SCF_R^+) *and* MCF_F,
b) MTZ and SCF_R (SCF_R^+),	*e)* SCF_R (SCF_R^+) *and* CF_F,
c) SCF_F *and* SCF_R (SCF_R^+),	*f)* MCF_R *and* CF_F.

Proof. a) In Figure 4 we have $\upsilon_{LP}(SCF_F) = 11 < 16 = \upsilon_{LP}(MTZ)$ and in Figure 7 we have $\upsilon_{LP}(MTZ) = 9 < 10 = \upsilon_{LP}(SCF_F)$.

b) In Figure 4 we have $\upsilon_{LP}(SCF_R) = 7.25 < \upsilon_{LP}(SCF_R^+) = 11 < \upsilon_{LP}(MTZ) = 16$ and in Figure 7 we have $\upsilon_{LP}(MTZ) = 9 < 17.25 = \upsilon_{LP}(SCF_R) < \upsilon_{LP}(SCF_R^+) = 21$.

c) In Figure 5 we have $\upsilon_{LP}(SCF_F) = 14.325 < 18.125 = \upsilon_{LP}(SCF_R)$ and in Figure 8 we have $\upsilon_{LP}(SCF_R) = 3.25 < \upsilon_{LP}(SCF_R^+) = 7 < \upsilon_{LP}(SCF_F) = 8$.

d) For Figure 5 we have $\upsilon_{LP}(SCF_R) = 18.125 > 18 = \upsilon_{LP}(MCF_F)$. For Figure 4 we have $\upsilon_{LP}(SCF_R) = 7.25 < \upsilon_{LP}(SCF_R^+) = 11 < \upsilon_{LP}(MCF_F) = 16$.

e) For Figure 4 we have $\upsilon_{LP}(SCF_R) = 7.25 < \upsilon_{LP}(SCF_R^+) = 11 < \upsilon_{LP}(CF_F) = 16$, for Figure 5 we have $\upsilon_{LP}(CF_F) = 18 < \upsilon_{LP}(SCF_R) = 18.125 < \upsilon_{LP}(SCF_R^+) = 22.25$.

f) Consider Examples 5 and 6. For Figure 5 we have $\upsilon_{LP}(CF_F) = 18 < 28 = \upsilon_{LP}(MCF_R)$, for Figure 6 we have $\upsilon_{LP}(MCF_R) = 22 < 24 = \upsilon_{LP}(CF_F)$.

□

Denote by $\mathcal{P}_.$ the polytope of the LP-relaxation of any of the MIP models described above, and with $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_.)$ the natural projection of that polytope onto the space of variables $\mathbf{x}$ and $\mathbf{z}$.

Lemma 6. *The following results hold:*

a) $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{CF_F}) \subset Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{MCF_F}) \subset Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{SCF_F})$, *and*

b) $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{CF_R}) \subset Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{MCF_R}) \subset Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{SCF_R^+}) \subset Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{SCF_R})$.

Proof. The results follow immediately from the corresponding results for Steiner trees, see e.g., [38]. Instances that prove the strict inclusion can be found in Table 1. □

Lemma 7. *The following results hold:*

a) $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{MCF_F}) = \mathcal{P}_{CUT_F} = Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{GSEC})$, *and*

b) $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{MCF_R}) = \mathcal{P}_{CUT_R}$.

Proof.

a) The first equality follows from the min-cut max-flow theorem, the second one follows from the related result for node-weighted Steiner trees, see e.g. [32].

b) This result follows from the min-cut max-flow theorem.

□

Lemma 8. *The following results hold:*

a) $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{MCF_R}) \subset Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{MCF_F})$ *and*

b) $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{CF_R}) \subset Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{CF_F})$.

Proof.

a) According to Lemma 7, it is enough to show this relationship by comparing $\mathcal{P}_{CUT_R}$ and $\mathcal{P}_{CUT_F}$. Then it is easy to see that every solution $(\mathbf{x}', \mathbf{z}') \in \mathcal{P}_{CUT_R}$ also belongs to $\mathcal{P}_{CUT_F}$. Figure 5, with $v_{LP}(CUT_R) = 28 > 18 = v_{LP}(CUT_F)$, proves that the opposite is not true.

b) $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{CF_R}) \subseteq Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{CF_F})$**:** Let $(\mathbf{f}', \bar{\mathbf{f}}', \mathbf{x}', \mathbf{z}')$ be in $\mathcal{P}_{CF_R}$. We define the capacities on the subgraph $G_S = (S, A_S)$ as x_{ij}, for all $ij \in A_S$. Since $x_{ij} = max_{k \in R} f_{ij}^k$, and $z_i = max_{ij \in A_R} x_{ij}$, there will be enough capacity to independently route z_i units of flow, for all $i \in F$, such that $z_i > 0$. Now, we are going to construct $(\mathbf{g}, \bar{\mathbf{g}}, \mathbf{x}, \mathbf{z}) \in \mathcal{P}_{CF_F}$ as follows: We fix the ordering of the outgoing arcs of every node $i \in S$ and then apply an adapted Ford-Fulkerson maximum flow algorithm. To define $\mathbf{g}$, we send z_i units of flow from r towards $i \in F$, for all $i \in F$ such that $z_i > 0$. When searching for augmenting paths, we always follow the fixed ordering. Therefore, the outgoing arcs of a node always get saturated in the same order, independently on the commodity under consideration. It follows directly from construction that the common flow $\bar{\mathbf{g}}$ for any pair of facilities k and l, once it splits up, will never meet again, i.e., ineqalities (18) will be satisfied.

$Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{CF_F}) \nsubseteq Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{CF_R})$**:** Consider Figure 5, where $v_{LP}(CF_R) = 28 > 18 = v_{LP}(CF_F)$.

□

Lemma 9. *Formulation GSEC (i.e., CUT_F, MCF_F) is strictly stronger than formulation MTZ, i.e.* $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{MCF_F}) \subset Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{MTZ})$.

Proof. Let C_S denote the set of all cycles in S and let C be the set of arcs defining an arbitrary cycle in C_S. Padberg and Sung [37] show that, variables u_i and constraints (26) can be projected out by using the following set of *cycle constraints*:

$$\sum_{ij \in C} x_{ij} \leq |C| - \frac{|C|}{|S|} \qquad \forall C \subseteq C_S \tag{33}$$

It is not difficult to see that cycle constraints (33) are implied by the generalized sub-tour elimination constraints (29), i.e.:

$$\sum_{ij \in C} x_{ij} \leq |C| - 1 \leq |C| - \frac{|C|}{|S|} \qquad \forall C \subseteq C_S$$

For Figure 7 we have $v_{LP}(MTZ) = 9 < 26 = v_{LP}(GSEC)$. Thus, $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{GSEC}) \subset Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{MTZ})$.

□

4.1. Reformulation as the Steiner Arborescence Problem

As we already observed in [40], the ConFL can be transformed into the Steiner Arborescence Problem. This transformation is done by using the well-known *node splitting* technique that has proven useful for different network design problems, see e.g., [3, 6].

To solve an instance of ConFL as SA, we use the following procedure:

- Generate a directed graph $\tilde{G} = (\tilde{V}, \tilde{A})$ with costs $\tilde{\mathbf{c}} : \tilde{A} \mapsto \mathrm{R}_0^+$, as follows:
  - Initialize $\tilde{V} = V$, $\tilde{A} = A$ and $\tilde{\mathbf{c}} = \mathbf{c}$.
  - For any facility node i, add a node i' to the graph, connect i to i', and set $\tilde{c}_{ii'} = f_i$.
  - Replace arcs $ik \in A_R$ by $i'k$.
- Solve the Steiner arborescence problem on the transformed graph $\tilde{G}$ with customers as terminals.

Recall that, given a directed graph $\tilde{G} = (\tilde{V}, \tilde{A})$, with arc weights $\tilde{\mathbf{c}} : \tilde{A} \mapsto \mathbb{R}$, a root $r \in \tilde{V}$, and a set of terminal nodes $R \subset \tilde{V}$, the Steiner arborescence problem searches for the cheapest subtree rooted at r that connects all terminals. Figure 9 shows a simple example that illustrates the transformation of ConFL into the SA problem, according to the procedure described above:

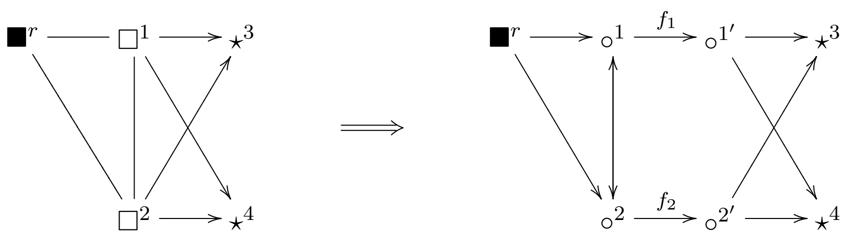


Figure 9: Initial undirected ConFL instance and transformed SA instance

For each facility $i \in F$, i corresponds to node's function as Steiner node, while i' corresponds to its function as open facility. With this transformation we ensure that the arc ii' belongs to a solution if and only if facility i is open. Similarly, facility i is used as Steiner node if and only if i belongs to the solution, but arc ii' does not. A similar, but undirected transformation has been used by Bardossy and Raghavan to transform (G)STS, ConFL and RoB into the GConFL [4].

To solve the SA problem as a MIP, let us define binary variables v_{ij} as follows:

$$v_{ij} = \begin{cases} 1, & \text{if } ij \text{ belongs to the solution} \\ 0, & \text{otherwise} \end{cases}, \quad \forall ij \in \tilde{A}.$$

We extend the directed cut-based formulation for Steiner trees (originally proposed by Chopra and

Rao [8]) by the root out-degree constraint as follows:

$$(SA) \qquad \min \sum_{ij \in \tilde{A}} \tilde{c}_{ij} v_{ij} \tag{34}$$

$$\sum_{ij \in \delta^-(W)} v_{ij} \geq 1 \qquad \forall W \subseteq \tilde{V} \setminus \{r\}, W \cap R \neq \emptyset \tag{35}$$

$$\sum_{ij \in \delta^-(\{j\})} v_{ij} = 1 \qquad \forall j \in R \tag{36}$$

$$v_{rr'} = 1 \tag{37}$$

$$v_{ij} \in \{0,1\} \quad \forall ij \in \tilde{A} \tag{38}$$

Let us denote by

$$Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{SA}) = \{(\mathbf{x},\mathbf{z}) \in [0,1]^{|A|} \times [0,1]^{|F|} \mid \mathbf{v} \in \mathcal{P}_{SA} \text{ and}$$

$$x_{kl} = v_{kl} \ \forall kl \in A_S; \ x_{ij} = v_{i'j} \ \forall ij \in A_R; \ z_i = v_{ii'} \ \forall i \in F\},$$

the projection of the $\mathcal{P}_{SA}$ polytope onto the space of variables $(\mathbf{x},\mathbf{z})$.

We show the following result:

Lemma 10. *The LP-relaxation of the Steiner arborescence formulation is equally strong as the LP-relaxation of CUT_R, i.e.:*

$$Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{SA}) = \mathcal{P}_{CUT_R}.$$

Proof. We prove equality by showing mutual inclusion.

$Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{SA}) \subseteq \mathcal{P}_{CUT_R}$**:** Let $\mathbf{v}'$ be a feasible solution of the LP-relaxation of SA, and $(\mathbf{x}',\mathbf{z}')$ its projection into $Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{SA})$. Obviously, (1), (2) and (4) are satisfied by $(\mathbf{x}',\mathbf{z}')$. It only remains to show that $x'_{ij} \leq z'_i, \quad \forall ij \in A_R$. Assume that there exist $j \in F$ and $k \in R$ such that $x_{jk} > z_j$. From inequalities (36) follows

$$1 = \sum_{i \in F \setminus \{j\}} x_{ik} + x_{jk} > \sum_{i \in F \setminus \{j\}} x_{ik} + z_j = \sum_{ij \in \delta^-(W)} v_{ij}$$

where $W = \{k, j'\}$ in contradiction to constraints (35).

$\mathcal{P}_{CUT_R} \subseteq Proj_{\mathbf{x},\mathbf{z}}(\mathcal{P}_{SA})$**:** Let $(\mathbf{x}',\mathbf{z}')$ be a fractional solution satisfying (1)-(4), and let us assume that the corresponding solution $\mathbf{v}'$ from $\mathcal{P}_{SA}$ is not feasible. In other words, assume that there exists a cut-set $\tilde{W} \subseteq \tilde{V} \setminus \{r\}, \tilde{W} \cap R \neq \emptyset$, such that $\sum_{ij \in \delta^-(\tilde{W})} v_{ij} < 1$. Obviously, there must exist at least one $i \in F \setminus \{r\}$, such that $ii' \in \delta^-(\tilde{W})$. We now construct a new cut-set $\tilde{W}_n$ such that $\delta^-(\tilde{W}_n) = \delta^-(\tilde{W}) \cup \{i'j \mid j \in \tilde{W}\} \setminus \{ii'\}$. Obviously, if $\sum_{ij \in \delta^-(\tilde{W})} v_{ij} < 1$, then also $\delta^-(\tilde{W}_n) < 1$. By repeating this procedure for all $i \in F$ such that $ii' \in \delta^-(\tilde{W})$, we end up with a cut-set containing only arcs from $A_R \cup A_S$, that violates inequality (35), which is a contradiction.

$\square$

4.2. Full Hierarchy of Formulations

The hierarchical scheme given in Figure 10 summarizes the relationships between the LP relaxations of the MIP models considered throughout this paper. A filled arrow specifies that the target formulation is strictly stronger than the source formulation. A dashed connection specifies that the formulations are not comparable to each other.
Note that we do not display formulation SCF_R^+ separately, because it has the same relations as the formulation SCF_R.

Note that all three models SCF_F, MCF_F and CF_F may have lower bounds as bad as $OPT/|F|$. Model CF_R is the strongest one among all considered throughout this paper. Observe that there are several other tree models known for Steiner trees, that can directly be interpreted in ConFL context. Therefore we do not mention them here, but refer the interested reader to Magnanti and Wolsey [32] and Polzin and Daneshmand [38].

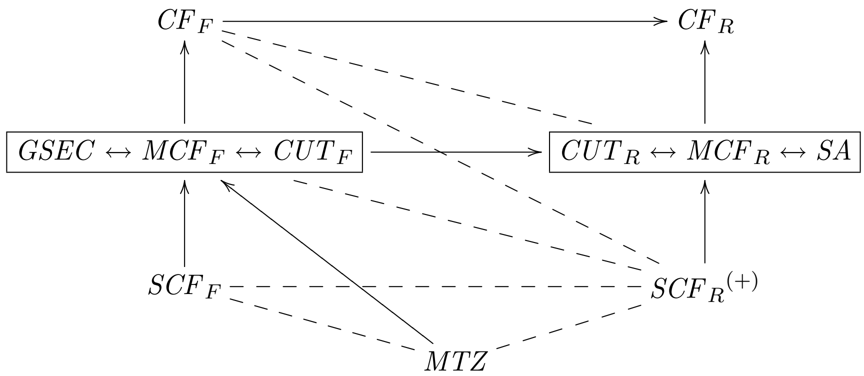


Figure 10: Relations between LP-relaxations of MIP models for ConFL

5. Branch-and-Cut Framework

We are going to calculate lower bounds and provably optimal solutions of CUT_F and CUT_R models using the same branch-and-cut framework described below. The only difference is in the separation of cut set inequalities. The main ingredients of our implementation are provided in this section.

Initialization:. We initialize the LP with assignment, capacity- and root-inequalities (2)-(4). The following flow-balance constraints introduced by Koch and Martin [22] are also introduced in the initialization phase. These constraints ensure that the in-degree of each Steiner node is less or equal than its out-degree:

$$\sum_{kl \in A} x_{kl} \leq \sum_{lk \in A} x_{lk}, \quad \forall l \in S \setminus F. \tag{39}$$

These constraints are not induced by any of the MIP formulations presented above, i.e., they can further strengthen the quality of lower bounds (see, e.g., [30, 38]).

Finally, we insert the following in-degree inequalities,

$$\sum_{kl \in A} x_{kl} \leq 1 \quad \forall l \in S \setminus \{r\}$$

and the *sub-tour elimination constraints* of size two,

$$x_{kl} + x_{lk} \leq 1 \quad \forall \{k,l\} \in E,\ k,l \in S,\ k \neq r.$$

The latter two groups of constraints are not necessarily binding, but they can speed up the cutting plane phase at the root node of the branch-and-bound (B&B) tree.

Branching:. Branching on single arc variables produces a huge disbalance in the branch-and-bound tree. Whereas discarding an edge from the solution (setting x_{ij} to zero) doesn't bring much, setting the facility variable to one significantly reduces the size of the search subspace. Therefore we set the highest branching priorities to variables z_i, $i \in F$.

5.1. Separation

Separation of Cut Set inequalities (8)*:.* In each node of the branch-and-bound tree we separate the cut-inequalities (8). For a given LP-solution $(\hat{\mathbf{x}}, \hat{\mathbf{z}})$, we construct a support graph $G_S = (S, A_S, \hat{\mathbf{x}})$ with arc capacities set to $\hat{x}_{ij}$, for all $ij \in A_S$. Then we calculate the maximum flow from the root node r to each potential facility node $i \in F$ such that $\hat{z}_i > 0$. If this maximum flow value is less than z_i, we have found a violated inequality (8), induced by the corresponding min-cut in the graph G_S, and we insert it into the LP. For the calculation of the maximum flow we used an adaptation of Cherkassky and Goldberg's maximum flow algorithm [7].

Separation of Cut Set Inequalities (1)*:.* In order to separate cut set inequalities (1), we build a support graph by copying $G = (V, A)$. For a given fractional solution $(\hat{\mathbf{x}}, \hat{\mathbf{z}})$, we set the capacities to $\hat{x}_{ij}$, for all $ij \in A$. We then calculate the maximum flow that can be sent from r to each of the customers $j \in R$. If there exists customer j such that the value of the maximum flow is less than one, we obtain a cut set, say $W \subset V$, $r \in W$, such that capacity of $\delta^+(W)$ is less than one. Obviously, $W \cap F \neq F$, since all the cuts involving only arcs from A_R are satisfied by (2). According to Observation 1, the violated cut set inequality (1) induced by W can then be written as: $\sum_{ij \in A_S^W} x_{ij} + \sum_{ij \in A_R^W} x_{ij} \geq 1$.

Enhancing Separation. To improve computational efficiency, we search for *nested, back* and *minimum-cardinality cuts* and insert at most 100 violated inequalities in each separation phase. For more details, see our implementation of the B&C algorithm for the prize-collecting Steiner tree problem, where the same separation procedure has been used [29, 30]. It is important to mention that the performance of the branch-and-cut algorithm can further be improved if we permute the order in which the minimum cuts between r and $i \in F$, $z_i > 0$, in CUT_F case, and between r and j, $j \in R$, in CUT_R case, are calculated. Since this permutation is done randomly, we fix the seed value for the results reported in Section 6.

5.2. Primal Heuristic

The primal heuristic works as follows: First, we initialize the set of open facilities according to fractional values z_i: if $z_i > \pi$, we label the facility as *selected.* Default value of π is set to 0.1. Denote by $\mathcal{F} = \{i \in F \mid z_i > \pi\}$, the set of initially selected facilities. Starting with $\mathcal{F}$, we then calculate a feasible ConFL solution according to the pseudo-code provided in Algorithm 1. We use the following notation:

- vector $\mathbf{x^S}$ refers to the core tree structure, i.e., $x^S_{ij} = 1$ if $ij \in A_S$ belongs to the solution, and it is zero otherwise.

- vector $\mathbf{x^A}$ refers to assignment values, i.e., $x^A_{ij} = 1$ if customer j is assigned to facility i and $x^A_{ij} = 0$, otherwise, for all $ij \in A_R$.
- vector $\hat{\mathbf{z}}$ is set to one if facility i is open, and to zero otherwise.
- T^S denotes the core Steiner tree (the set of nodes and edges) that is uniquely defined by $\mathbf{x^S}$.

Outline. The algorithm works in three phases: In the *assignment phase* (*Assign*), the cheapest assignment of customers to facilities from $\mathcal{F}$ is found. If there are non-assigned customers, solution is discarded. The set $\mathcal{F}$ is updated to contain only *open* facilities, i.e., those that serve at least one customer. In the *Steiner tree phase*, the set of *open facilities* is connected by a Steiner tree. For that purpose, we use the minimum spanning tree heuristic (*MSTHeuristic*) described below. Finally, we apply a *local improvement* procedure (*Peeling*) that tries to remove leaves of the Steiner tree in the core network and to re-assign customers to already open facilities, by decreasing the overall costs.

Data: Binary vector $\hat{\mathbf{z}}$: a facility i is *selected* if $\hat{z}_i = 1$.
Result: Locally improved solution $(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}})$.

if *Hash*$(\hat{\mathbf{z}})$ *defined* **then**
    $(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}}) = Hash(\hat{\mathbf{z}})$;
else
    if *Assignment exists* **then**
        $(\mathbf{x^A}, \hat{\mathbf{z}}) := Assign(\hat{\mathbf{z}})$;
        $(\mathbf{x^S}, \hat{\mathbf{z}}) := MSTHeuristic(\hat{\mathbf{z}})$;
        $(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}}) := Peeling(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}})$;
        Insert $(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}})$ into *Hash*;
    else
        return infeasible;
    end
end
return $(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}})$;

Algorithm 1: The primal heuristic: calculation of the objective function for a given vector $\hat{\mathbf{z}}$.

Hashing. Given a vector of selected facilities, $\hat{\mathbf{z}}$, we first check if the objective value for this configuration has been already calculated before (see, e.g., [24]). If so, we get the corresponding solution $(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}})$ from the hash-table *Hash*. Otherwise, we run a three-step procedure whose steps are described below.

Detailed Description.

Step 1: $(\mathbf{x^A}, \hat{\mathbf{z}}) := Assign(\hat{\mathbf{z}})$**:** For each customer $j \in R$, we find the cheapest possible assignment to a facility from $\hat{\mathbf{z}}$. The assignment values are stored in vector $\mathbf{x^A}$. We close those facilities i from $\mathcal{F}$ that do not serve any customer, i.e., we set $\hat{z}_i := 0$. If such assignment is not possible (e.g., the subgraph induced by A_R is not a complete bipartite graph), we discard the solution.

This operation is calculated from scratch. Thus, the total computational complexity for finding the cheapest assignment in the worst case is $O(|\mathcal{F}||R|)$.

Step 2: $(\mathbf{x^S}, \hat{\mathbf{z}}) := \mathit{MSTHeuristic}(\hat{\mathbf{z}})$**:** We consider the graph $G' = (S, E_S)$ – a subgraph of G induced by the set of facilities and Steiner nodes with the edge costs $\mathbf{c}$. For G', we generate the so-called *distance network*[2] - a complete graph whose nodes correspond to facilities $i \in F$, and whose edge-lengths l_{ij} are defined as shortest paths in G', for all $i, j \in F$.

We use the minimum spanning tree (MST) heuristic [34] to find a spanning tree T^S that connects all open facilities ($\hat{z}_i = 1$).

1. Let G'' be the subgraph of G' induced by $\mathcal{F}$.
2. Calculate the minimum spanning tree MST''_G of the distance sub-network G''.
3. On the subgraph of (S, E_S) obtained by back-mapping the edges from MST''_G, re-calculate the minimum spanning tree (T^S) to obtain vector $\mathbf{x^S}$.

Step 3: $(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}}) := \mathit{Peeling}(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}})$**:** We finally want to get rid of some of those facilities that are still part of the Steiner tree, but that are not used at all. We do this by applying the so-called *peeling procedure*. Our peeling heuristic tries to recursively remove all redundant leaf nodes (including corresponding tree-paths) from the tree-solution defined by $\mathbf{x^S}$. Let k denote a leaf node of T^S, and let P_k be a path that connects k to the next open facility from $\mathcal{F}$, or to the next branch, towards the root r.

1. If the leaf node is not an (open) facility, i.e. if $\hat{z}_k = 0$, we simply delete P_k.
2. Otherwise, we try to re-assign customers (originally assigned to k) to already open facilities (if possible). If such obtained solution is better, we delete P_k and continue processing other leaves.

The main steps of this procedure are given in Algorithm 2.

If, for each customer, the set of facilities is sorted in increasing order with respect to its assignment costs[3], this procedure can be implemented very efficiently. Indeed, in order to find an open facility from $\mathcal{F}$, nearest to j and different from k (denoted by $i^k(j)$), we only need to proceed this ordered list starting from k until we encounter a facility i such that $\hat{z}_i = 1$.

The algorithm stops when only one node is left, or when all the leaves from the tree have been proceeded. Thus, the worst-case running time of the whole peeling method is $O(|\mathcal{F}||R|)$.

6. Computational Results

In our computational study, two groups of instances were considered:

[2] Calculation of the distance network is done only once, during the initialization of the branch-and-cut algorithm.
[3] Also sorting of these lists is done once, in the initialization phase of the branch-and-cut algorithm.

Data: Assignment $\mathbf{x^A}$, open facilities $\hat{\mathbf{z}}$ and a Steiner tree T^S corresponding to $\mathbf{x^S}$.
Result: Locally improved solution $(\mathbf{x^S}, \mathbf{x^A}, \hat{\mathbf{z}})$.

for *all leaves k in T^S* **do**
  Determine path P_k and its costs $c(P_k) := \sum_{e \in P_k} c_e$;
  if $\hat{z}_k = 0$ **then**
    $T^S := T^S - P_k$;
  else
    $R_k := \{j \mid j \in R,\ x^A_{kj} = 1\}$;
    $i^k(j) = \arg\min\{c_{ij} \mid i \in F,\ \hat{z}_i = 1,\ i \neq k\}, \forall j \in R_k$;
    if $\exists j \in R_k : i^k(j) = \emptyset$ **then**
      continue;
    end
    if $\sum_{j \in R_k} c_{i^k(j)j} < f_k + c(P_k) + \sum_{j \in R_k} c_{kj}$ **then**
      $\hat{z}_k := 0$;
      $T^S := T^S - P_k$;
      $x^A_{kj} := 0,\ x^A_{i^k(j)j} := 1, \forall j \in R_k$;
    end
  end
end

Algorithm 2: Peeling procedure.

Randomly Generated Graphs From [40]. For this set of instances the parameters for the generation were set as follows: $|S| \in \{20, 50, 100\}$, $|R| \in \{20, 50, 100\}$. Edges of the core network are generated with probability $p(S) \in \{0.1, 0.5, 1\}$, while the connections between facilities and customers are established with probability $p(R) \in \{0.18, 0.55, 1\}$. Edge weights were uniformly randomly set to an integer value between 50 and 100. Finally, the facility opening costs were uniformly randomly assigned to values between 150 and 200. Increasing only the core costs did not significantly change the behavior of the GRASP algorithm for this set of instances. The core network was generated by MAPLE [2], using the parameters given above. Finally, customers are randomly linked to the existing nodes using the density values $p(R)$.
As the original instances are unrooted we selected the facility with the highest index for the root node respectively.

Graphs Derived From OR-library [5] and UflLib [33]. We consider another class of benchmark instances, obtained by merging data from two public sources. In general, we combine an UFLP instance with an STP instance, to generate ConFL input graphs in the following way: first $|F|$ nodes of the STP instance are selected as potential facility locations, and the node with index 1 is selected as the root. The number of facilities, the number of customers, opening costs and assignment costs are provided in UFLP files. STP files provide edge-costs and additional Steiner nodes.

- We consider two sets of non-trivial UFLP instances from UflLib [33]:

- mp-{1,2} and mq-{1,2} instances have been proposed by Kratica et al. [24]. They are designed to be similar to UFLP real-world problems and have a large number of near-optimal solutions. There are 6 classes of problems, and for each problem $|F| = |R|$. We took 2 representatives of the 2 classes MP and MQ of sizes 200×200 and 300×300, respectively.
- The gs-{250,500}a-{1,2} benchmark instances were initially proposed by Koerkel [23] (see also Ghosh [14]). Here we chose two representatives of the 250×250 and 500×500 classes, respectively. The authors drew uniformly at random connection costs from [1000, 2000], and the facility opening costs from [100, 200].

- STP instances: Instances {c,d}n, for $n \in \{5, 10, 15, 20\}$ were chosen randomly from the OR-library [5] as representatives of medium size instances for the STP. These instances define the core networks with between 500 and 1000 nodes and with up to 25,000 edges.

Combined with assignment graphs, the largest instances of this data set contain 1,300 nodes and 115,000 edges.

All experiments were performed on a Intel Core2 Quad 2.33 GHz machine with 3.25 GB RAM, where each run was performed on a single processor. For solving the linear programming relaxations and for a generic implementation of the branch-and-cut approach, we used the commercial packages IBM CPLEX (version 11.2) [1] and ILOG Concert Technology (version 2.7).

6.1. Testing Randomly Generated Instances

For the following tests we turn the primal heuristics off, in order to compare lower bounds of all presented MIP formulations. Furthermore, our preliminary results have shown that turning all CPLEX general purpose cuts off speeds up the performance. Therefore, and in order to avoid biased results, all the results reported in this paper are obtained without usage of these cuts.

LP-gaps. We first test the performance and the quality of lower bounds for proposed formulations. For that purpose, we run the models as linear programs. Table 3 provides the average gaps calculated as $(OPT - v_{LP}(.))/OPT$, where optimal values are obtained by running the branch-and-cut approach (see below). The set of 81 instances is divided into 3 groups according to the size of the core- and the assignment-subgraph.

Not surprisingly, the worst gaps are obtained by running SCF_R model in which "big-M" constraints affect all the arcs in G. Comparing gap values of SCF_F model on these three groups, we observe that the gap increases with the size of the nodes of the core network. This is also not surprising, since "big-M" constraints of the SCF_F model affect only the core network. We observe that there is a correlation between the size of the two subgraphs and the quality of obtained lower bounds for the other models as well. The gaps obtained by MTZ model are surprisingly good, and very close to those obtained by MCF_F. The best LP-gaps are obtained by MCF_R model. Interestingly, the most difficult instances for the latter three models appear to be those with the equal number of facilities and customers.

Finally, we tried to make the same experiment with CF_F and CF_R models, but apparently in almost all cases the execution has been terminated because of memory overconsumption.

Solving MIPs. Table 2 shows the running times in seconds ($t[s]$) and the number of branch-and-bound nodes ($B\&B$) needed to solve this set of instances. Each row corresponds to three instances generated according to the same probabilities $p(R)$ and $p(S)$. We provide values for $t[s]$ and $B\&B$ averaged over the respective group. We set the time limit to 1000 seconds. If at least one of the three instances per group is not solved to optimality, we denote this by "-".
As expected, due to the weak lower bounds of the SCF_R^+, most of the instances could not be solved to optimality within the given time limit. The exceptions are graphs with complete bipartite structure of the assignment subgraph A_R that appear to be easy for SCF_R^+. The second worse performance was shown by the MCF_R model, which is easily explained by its huge number of variables.
This test gives two surprising results:

1. Despite the fact, that the integrality gap of model CUT_F can be as bad as $\frac{1}{|F|}$ it outperforms even the strongest cut set based model CUT_R with respect to the running time. On average, the number of $B\&B$ nodes needed by CUT_F is greater by a factor of 2.3 than the corresponding number for CUT_R. However, averaged over all 81 instances, CUT_F is about 4.6 times faster than CUT_R.

2. The compact MTZ model with arbitrarily bad lower bounds performs comparatively well. It outperforms CUT_R: The average running time over all instances for MTZ is 5.6% less than the corresponding time for CUT_R.

$\|S\|$	$\|R\|$	$p(S)$	$p(R)$	Opt	MTZ $t[s]$	MTZ $B\&B$	SCF_R^+ $t[s]$	SCF_R^+ $B\&B$	MCF_R $t[s]$	MCF_R $B\&B$	CUT_F $t[s]$	CUT_F $B\&B$	CUT_R $t[s]$	CUT_R $B\&B$
20	100	0.1	0.18	9,768	**0.10**	1	-	-	2.00	0	**0.10**	0	0.47	0
20	100	0.5	0.18	9,577	0.29	10	-	-	2.77	0	**0.09**	0	0.48	0
20	100	1.0	0.18	9,554	0.56	48	-	-	9.53	2	**0.12**	0	0.57	0
20	100	0.1	0.55	7,428	2.52	103	-	-	26.92	36	**1.57**	70	5.09	43
20	100	0.5	0.55	7,289	1.46	52	-	-	301.27	31	**1.26**	55	8.18	38
20	100	1.0	0.55	7,316	1.97	57	-	-	-	-	**1.41**	67	9.24	49
20	100	0.1	1.00	6,675	2.39	48	1.59	29	10.54	4	**1.21**	28	4.04	4
20	100	0.5	1.00	6,683	2.02	25	1.41	22	110.56	10	**1.40**	37	6.50	11
20	100	1.0	1.00	6,632	1.97	25	1.20	27	258.67	9	**1.05**	19	9.92	11
50	50	0.1	0.18	5,295	4.47	171	-	-	-	-	**2.50**	81	20.39	45
50	50	0.5	0.18	5,019	10.61	242	-	-	-	-	**2.09**	22	22.04	28
50	50	1.0	0.18	4,987	4.43	42	-	-	-	-	**3.38**	67	16.31	37
50	50	0.1	0.55	4,045	5.24	123	-	-	217.10	12	**3.97**	94	9.05	14
50	50	0.5	0.55	4,011	**6.67**	55	-	-	-	-	7.09	118	38.06	31
50	50	1.0	0.55	3,896	7.52	47	-	-	-	-	**4.13**	74	28.82	21
50	50	0.1	1.00	3,615	4.91	51	**1.10**	4	25.53	3	1.81	29	2.77	4
50	50	0.5	1.00	3,596	5.44	26	2.18	16	284.24	5	**1.64**	21	4.87	8
50	50	1.0	1.00	3,596	7.30	16	**2.17**	21	-	-	2.74	21	10.28	13
100	20	0.1	0.18	2,489	1.84	16	251.27	171,598	122.51	6	**1.22**	14	2.72	3
100	20	0.5	0.18	2,463	10.43	35	-	-	-	-	**5.30**	33	23.57	16
100	20	1.0	0.18	2,487	144.35	378	-	-	-	-	**7.79**	44	42.62	39
100	20	0.1	0.55	1,921	4.44	51	118.32	27,557	43.20	7	**1.75**	30	2.68	9
100	20	0.5	0.55	1,876	8.66	31	-	-	-	-	4.08	29	**3.26**	5
100	20	1.0	0.55	1,873	14.16	13	-	-	-	-	**2.12**	13	4.95	4
100	20	0.1	1.00	1,638	1.16	4	0.82	4	3.07	0	**0.29**	3	1.02	0
100	20	0.5	1.00	1,638	2.70	1	1.26	1	8.98	0	**0.47**	2	1.59	0
100	20	1.0	1.00	1,633	7.26	2	1.31	1	21.08	0	**0.91**	2	2.59	0

Table 2: Running times (in seconds) and the number of Branch-and-Bound nodes for selected MIP formulations with CPLEX cuts turned off.

Testing the influence of the factor M. In the following test, we multiply the costs of the core network by a factor $M \in \{3, 5, 10\}$. Our goal is to test the influence of the cost structure of the core network on the overall performance of proposed MIP models. For that purpose, we select the best performing models according to the results obtained above, namely: MTZ, CUT_F and CUT_R. As a reference value, we take the average running time the model CUT_L needed to solve the problems with $M = 1$ to optimality. For each of the three MIP models, and for each of possible M values, we divide the corresponding average running time with the reference time to calculate the so-called *slow down factor* shown in Figure 11.

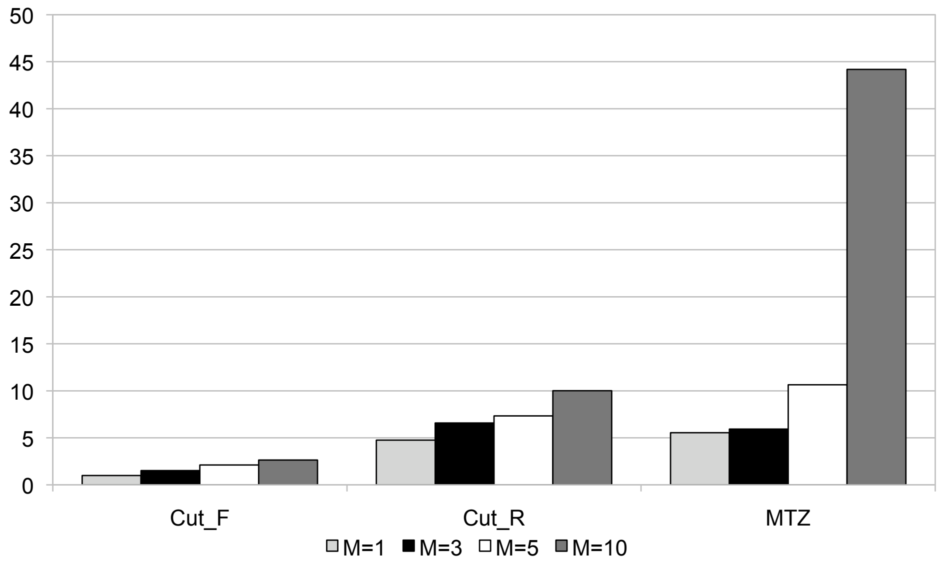


Figure 11: Results for randomly generated instances from [40]: Average slow-down factors for three MIP models and for $M \in \{1, 3, 5, 10\}$.

The obtained slow down factors indicate that the MTZ model is the most affected by increasing the costs of the core network: MTZ needs about 7 times more time to solve the instances to optimality, if the costs of the core network are multiplied by factor $M = 10$. This result is due to decreasing quality of lower bounds of the MTZ model with increasing M values. On the other hand, models CUT_F and CUT_R are less affected by that effect: In the worst case, when $M = 10$, the average running time increases by roughly a factor of 2.6 and 2.1 for CUT_F and CUT_R, respectively. We also observe that CUT_F outperforms MTZ by a factor of 5 for $M = 1$, and by a factor of 16 for $M = 10$.

Branching. We also tested our branching strategy described in Section 5 against the CPLEX default branching strategy. For each of the 27 density settings, Figure 12 shows the speed up factor obtained by dividing two running times: one needed to solve the instance with default CPLEX setting to optimality and the other one obtained with our branching strategy. The values are averaged over three instances per setting. In most of the cases our branching strategy significantly reduces the overall running time. On average over all 81 instances, our branching strategy outperforms CPLEX default branching by a factor of 1.4, 3.3 and 2.9, when models MTZ, CUT_F and CUT_R are solved, respectively.

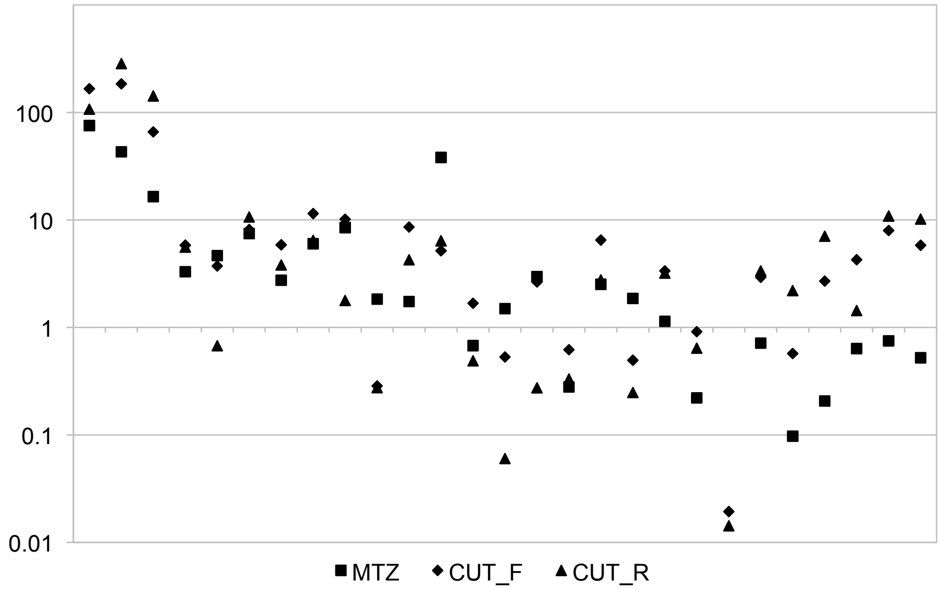


Figure 12: Results for randomly generated instances from [40]: Speed-up factors obtained by using branching priorities for facility nodes against default branching times.

$\lvert S\rvert$	$\lvert R\rvert$	MTZ	SCF_F	SCF_R	MCF_F	MCF_R
20	100	1.36 %	5.44 %	96.24 %	1.33 %	0.73 %
50	50	2.57 %	7.33 %	93.28 %	2.51 %	1.36 %
100	20	2.48 %	8.33 %	85.19 %	2.43 %	1.22 %

Table 3: Average integrality gaps ($(OPT - v_{LP}(.))/OPT$) for selected MIP formulations

6.2. Testing Larger Graphs

The set of instances is divided into three groups according to the underlying instance for the assignment graph. We refer to them as `mp`, `mq` and `gs` group. Tables 4 and 5 report on the results obtained through this experiment. Note that the optimal values, as well as lower bounds reported in this paper differ from those reported in [28]. This is due to in-degree inequalities used in [28], that turned out to model the Steiner tree star problem, instead of ConFL.

Comparing Two Branch-and-Cut Approaches:. First, we compare the two branch-and-cut approaches by running them with the proposed primal heuristic. Regarding 32 instances obtained by combining `stein` and `mp/q` instances, CUT_F solves all 32 instances to provable optimality within 213 seconds on average. The gaps we report for each model were calculated as

$$gap[\%] = \frac{UB - LB}{UB},$$

where UB and LB are the upper and lower bound obtained by the respective model. In addition, we report on the running time in seconds (t $[s]$), the model CUT_F needs to solve the instances of the `mp/q` group to optimality. Note that CUT_R solves only 7 out of 32 `mp/q` instances to optimality. For the majority of instances CUT_R does not branch at all, as it has not finished the cutting plane phase at the root node of the branch-and-bound tree. This is because the assignment graphs for these instances are complete bipartite, which means that many dense cuts of the CUT_R model need to be separated.

Comparing MIP Models Initialized with Best Upper Bound:. Second, we run all three models, MTZ, CUT_F and CUT_R, but we deactivate the primal heuristic. Instead, we initialize the models with the best upper bound found in the previous setting. For the `gs` group of instances, the best lower and upper bounds obtained with this setting can be found in the right hand half of Table 5. Each of the models MTZ and CUT_R solves only 8 instances to optimality. For the `mp` subgroup, MTZ gives much smaller gaps though, on average 0.17% compared to 1.42% for CUT_R. For the group of `mq` instances MTZ also outperformes CUT_R with an average gap of 1.86% vs. 3.18% for the latter. In the last group of large scale instances derived from the `gs` group, the performance of MTZ is comparatively better. CUT_F obtains the smallest gap in 11 cases, but MTZ performs best on 7 instances. Not a single instance of `gs` group has been solved to optimality. Note that for this last group of instances the cost structure is special. The factor M, describing the scale between core and assignment costs is about 0.001.

			PH on, no UB given					PH off, best UB given						
			CUT_R		CUT_F			MTZ		CUT_R		CUT_F		
Stein	UFL	OPT	gap[%]	$B\&B$	gap[%]	$B\&B$	t [s]	gap[%]	$B\&B$	gap[%]	$B\&B$	gap[%]	$B\&B$	t [s]
c05	mp1	2,691.5	**0.00**	13	**0.00**	27	73	0.34	605	**0.00**	23	**0.00**	33	50
c10	mp1	2,661.7	**0.00**	17	**0.00**	17	67	**0.00**	86	**0.00**	23	**0.00**	25	47
c15	mp1	2,634.7	1.45	1	**0.00**	15	100	0.15	1084	1.39	3	**0.00**	17	73
c20	mp1	2,618.7	1.91	3	**0.00**	33	185	**0.00**	58	1.50	1	**0.00**	11	104
d05	mp1	2,677.9	**0.00**	9	**0.00**	27	62	**0.00**	19	**0.00**	9	**0.00**	37	40
d10	mp1	2,676.5	2.39	0	**0.00**	21	92	0.24	542	2.39	1	**0.00**	21	66
d15	mp1	2,635.7	1.05	5	**0.00**	13	67	**0.00**	43	**0.00**	15	**0.00**	11	41
d20	mp1	2,619.7	1.59	0	**0.00**	27	229	0.06	49	1.59	1	**0.00**	15	82
c05	mp2	2,692.5	**0.00**	11	**0.00**	15	37	**0.00**	58	**0.00**	17	**0.00**	13	26
c10	mp2	2,661.5	**0.00**	9	**0.00**	5	27	**0.00**	97	**0.00**	7	**0.00**	11	23
c15	mp2	2,640.5	0.61	3	**0.00**	10	47	0.13	1772	0.89	0	**0.00**	5	28
c20	mp2	2,626.5	**0.00**	11	**0.00**	11	55	0.06	300	**0.00**	11	**0.00**	11	43
d05	mp2	2,710.6	**0.00**	25	**0.00**	19	41	**0.00**	1048	**0.00**	31	**0.00**	17	31
d10	mp2	2,682.5	1.14	0	**0.00**	29	50	0.26	574	0.94	3	**0.00**	27	50
d15	mp2	2,647.5	0.53	7	**0.00**	7	43	**0.00**	14	0.53	7	**0.00**	7	31
d20	mp2	2,628.5	2.14	0	**0.00**	11	222	0.09	70	2.14	0	**0.00**	11	142
c05	mq1	3,907.0	3.08	1	**0.00**	53	261	1.56	11	3.08	1	**0.00**	41	193
c10	mq1	3,866.5	4.12	0	**0.00**	35	214	1.49	20	4.12	0	**0.00**	37	146
c15	mq1	3,842.5	3.09	0	**0.00**	41	183	1.61	12	3.09	0	**0.00**	35	142
c20	mq1	3,826.5	3.08	0	**0.00**	33	289	1.43	7	3.08	0	**0.00**	35	173
d05	mq1	3,879.0	2.56	1	**0.00**	31	210	**0.00**	25	2.12	3	**0.00**	51	127
d10	mq1	3,869.1	2.99	0	**0.00**	43	242	1.72	15	2.92	0	**0.00**	29	156
d15	mq1	3,843.5	2.68	3	**0.00**	61	173	1.07	28	2.02	5	**0.00**	37	134
d20	mq1	3,828.5	2.80	0	**0.00**	45	483	1.87	5	2.80	0	**0.00**	39	387
c05	mq2	3,768.6	2.89	0	**0.00**	73	561	2.99	10	2.88	0	**0.00**	71	283
c10	mq2	3,732.6	5.14	0	**0.00**	63	320	2.99	9	5.14	1	**0.00**	50	190
c15	mq2	3,689.6	2.31	0	**0.00**	41	259	1.23	6	2.31	0	**0.00**	69	231
c20	mq2	3,686.5	4.58	0	**0.00**	45	620	2.33	3	4.03	0	**0.00**	27	317
d05	mq2	3,741.5	2.60	0	**0.00**	47	276	1.34	8	2.59	0	**0.00**	73	236
d10	mq2	3,720.9	**4.24**	0	**0.00**	31	285	4.07	6	2.52	0	**0.00**	43	396
d15	mq2	3,696.5	3.96	0	**0.00**	41	328	1.49	5	2.44	0	**0.00**	33	198
d20	mq2	3,685.5	5.73	0	**0.00**	27	727	2.60	2	5.73	0	**0.00**	33	402

Table 4: Results for large scale instances I: The best obtained gaps per setting and instance are shown in bold.

		PH on, no UB given						PH off, best UB given							
				CUT_R		CUT_F				MTZ		CUT_R		CUT_F	
Stein	UFL	best UB	best LB	*gap*[%]	*B&B*	*gap*[%]	*B&B*	best UB	best LB	*gap*[%]	*B&B*	*gap*[%]	*B&B*	*gap*[%]	*B&B*
c5	gs250a-1	258,568.0	258,088.8	0.27	2	**0.19**	162	258,540.0	258,112.9	0.20	180	0.27	5	**0.17**	289
c10	gs250a-1	258,480.0	257,955.7	0.25	1	**0.20**	147	258,464.0	257,986.5	0.20	201	0.20	7	**0.18**	227
c15	gs250a-1	258,387.0	257,823.3	**0.22**	0	-	-	258,387.0	257,858.5	**0.20**	280	0.23	3	-	-
c20	gs250a-1	258,250.0	257,786.4	0.50	0	**0.18**	15	258,250.0	257,798.6	**0.18**	28	0.52	0	0.49	28
c5	gs250a-2	258,287.0	257,724.9	**0.22**	0	0.31	68	258,077.0	257,744.4	0.23	125	0.42	2	**0.13**	192
c10	gs250a-2	257,990.0	257,600.0	0.24	0	**0.15**	92	257,990.0	257,625.1	**0.14**	120	0.22	3	0.19	175
c15	gs250a-2	257,911.0	257,564.4	0.45	0	**0.13**	17	257,911.0	257,536.4	**0.15**	109	0.27	1	-	-
c20	gs250a-2	258,193.0	257,462.5	0.53	0	**0.28**	6	258,054.0	257,471.5	0.28	11	0.53	0	**0.23**	15
c5	gs500a-1	513,476.0	510,860.9	0.53	0	**0.51**	0	513,364.0	510,866.9	0.51	0	**0.49**	0	0.55	0
c10	gs500a-1	513,148.0	510,733.5	0.48	0	**0.47**	0	513,091.0	510,734.9	0.47	0	0.52	0	**0.46**	2
c15	gs500a-1	512,919.0	510,637.7	0.47	0	**0.45**	0	512,919.0	510,635.8	**0.45**	0	0.47	0	**0.45**	0
c20	gs500a-1	513,158.0	510,568.0	0.51	0	**0.50**	0	513,131.0	510,568.0	-	-	0.52	0	**0.50**	0
c5	gs500a-2	513,663.0	510,844.5	0.61	0	**0.55**	0	513,544.0	510,846.2	0.55	0	0.61	0	**0.53**	0
c10	gs500a-2	513,357.0	510,717.7	0.57	0	**0.51**	0	513,357.0	510,719.7	**0.52**	0	0.55	0	**0.52**	0
c15	gs500a-2	513,127.0	510,616.9	**0.49**	0	**0.49**	0	513,127.0	510,617.4	**0.49**	0	**0.49**	0	**0.49**	0
c20	gs500a-2	513,511.0	510,545.7	**0.58**	0	0.59	0	513,254.0	510,545.7	-	-	**0.53**	0	0.58	0

Table 5: Results for large scale instances II: The best obtained gaps per setting and instance are shown in bold.

7. Conclusion

We provide a first theoretical comparison of MIP models for ConFL. We show that there are basically two groups of models, derived from the way the connectivity requirements in the whole graph are defined. Our "F" models require connectivity among open facilities and the root node, and in addition a proper assignment of customers. We derive the stronger "R" models by requiring connectivity between customers and the root node. There is also the weak Miller-Tucker-Zemlin formulation which follows a sub-tour elimination concept, instead of a connectivity-based one. In contrast to known results for the traveling salesman problem [42], we show that *MTZ* is not dominated by the two single commodity flow models. The second interesting result is that, in general, the integrality gap of all "F" models is not a constant value.

In our computational study we also obtain two surprising results. First, the branch-and-cut algorithm for the correspondingly weaker "F" cut-based model, significantly outperforms all other models in practice. Second, the weak but small *MTZ* formulation performs comparatively well, and in most cases outperforms even the branch-and-cut derived for the stronger "R" model.

Acknowledgements

The authors thank to Markus Chimani for the proof of Lemma 8b.

Bibliography

[1] IBM CPLEX. URL `http://www.ilog.com/products/cplex/`.

[2] Maple. URL `http://www.maplesoft.com/`.

[3] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network Flows*. Prentice Hall, 1993.

[4] M. G. Bardossy and S. Raghavan. Dual-based local search for the connected facility location and related problems. *INFORMS Journal on Computing*, 2010. , to appear.

[5] J. E. Beasley. OR-Library: distributing test problems by electronic mail. *Journal of the Operational Research Society*, 41(11):1069–1072, 1990. URL `http://people.brunel.ac.uk/{~}mastjjb/jeb/orlib/steininfo.html`.

[6] S. Chen, I. Ljubić, and S. Raghavan. The regenerator location problem. *Networks*, 2009. doi: 10.1002/net.20366. Published Online: Dec 22 2009.

[7] B. V. Cherkassky and A. V. Goldberg. On implementing push-relabel method for the maximum flow problem. *Algorithmica*, 19:390–410, 1994.

[8] S. Chopra and M. R. Rao. The Steiner tree problem I: Formulations, compositions and extension of facets. *Mathematical Programming*, 64:209–229, 1994.

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

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

[11] A. M. Costa, J.-F. Cordeau, and G. Laporte. Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints. *Networks*, 53(2):141–159, 2009.

[12] M. Desrochers and G. Laporte. Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints. *Operations Research Letters*, 10(1):27 – 36, 1991.

[13] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Approximating connected facility location problems via random facility sampling and core detouring. In *SODA '08: Proceedings of the nineteenth annual ACM-SIAM symposium on Discrete algorithms*, pages 1174–1183. SIAM, 2008.

[14] D. Ghosh. Neighborhood search heuristics for the uncapacitated facility location problem. *European Journal of Operations Research*, 150:150–162, 2003.

[15] M. X. Goemans. The Steiner tree polytope and related polyhedra. *Mathematical Programming*, 63:157–182, 1994.

[16] M. X. Goemans and Y. Myung. A catalog of Steiner tree formulations. *Networks*, 23, 1993.

[17] L. Gouveia. Using the Miller-Tucker-Zemlin constraints to formulate a minimal spanning tree problem with hop constraints. *Comput. Oper. Res.*, 22(9):959–970, 1995.

[18] A. Gupta, J. Kleinberg, A. Kumar, R. Rastogi, and B. Yener. Provisioning a virtual private network: A network design problem for multicommodity flow. In *In Proceedings of the 33rd Annual ACM Symposium on Theory of Computing*, pages 389–398, 2001.

[19] H. Jung, M. Hasan, and K.-Y. Chwa. Improved primal-dual approximation algorithm for the connected facility location problem. *Combinatorial Optimization and Applications*, pages 265–277, 2008.

[20] D. R. Karger and M. Minkoff. Building Steiner trees with incomplete global knowledge. In *FOCS '00: Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, pages 613–623, 2000.

[21] S. Khuller and A. Zhu. The general Steiner tree-star problem. *Information Processing Letters*, 84(4):215–220, 2002.

[22] T. Koch and A. Martin. Solving Steiner tree problems in graphs to optimality. *Networks*, 32: 207–232, 1998.

[23] M. Koerkel. On the exact solution of large-scale simple plant location problems. *European Journal of Operations Research*, 39:157–173, 1989.

[24] J. Kratica, D. Tošić, V. Filipović, and I. Ljubić. Solving the simple plant location problem by genetic algorithms. *RAIRO - Operations Research*, 35(1):127–142, 2001.

[25] C. Krick, H. Räcke, and M. Westermann. Approximation algorithms for data management in networks. In *SPAA '01: Proceedings of the thirteenth annual ACM symposium on Parallel algorithms and architectures*, pages 237–246. ACM, 2001.

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

[27] Y. Lee, Y. Chiu, and J. Ryan. A branch and cut algorithm for a Steiner tree-star problem. *INFORMS Journal on Computing*, 8(3):194–201, 1996.

[28] I. Ljubić. A hybrid VNS for connected facility location. In T. Bartz-Beielstein, M. J. B. Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 157–169. Springer, 2007.

[29] I. Ljubić. *Exact and Memetic Algorithms for Two Network Design Problems.* PhD thesis, Faculty of Computer Science, Vienna University of Technology, November 2004.

[30] I. Ljubić, R. Weiskircher, U. Pferschy, G. Klau, P. Mutzel, and M. Fischetti. An algorithmic framework for the exact solution of the prize-collecting Steiner tree problem. *Mathematical Programming*, 105(2-3):427–449, 2006.

[31] A. Lucena and M. G. C. Resende. Strong lower bounds for the prize-collecting Steiner problem in graphs. *Discrete Applied Mathematics*, 141:277–294, 2004.

[32] T. Magnanti and L. Wolsey. Optimal trees. *Handbook in Operations Research and Management Science*, pages 503–615, 1995.

[33] Max-Planck-Institut für Informatik. UflLib, 2003. URL `http://www.mpi-inf.mpg.de/departments/d1/projects/benchmarks/UflLib/`.

[34] K. Mehlhorn. A faster approximation for the Steiner problem in graphs. *Information Processing Letters*, 27:125–128, 1988.

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

[36] T. Öncan, I. K. Altınel, and G. Laporte. Invited review: A comparative analysis of several asymmetric traveling salesman problem formulations. *Comput. Oper. Res.*, 36(3):637–654, 2009.

[37] M. Padberg and T.-Y. Sung. An analytical comparison of different formulations of the travelling sales man problem. *Mathematical Programming*, 52(2):315–357, 1991.

[38] T. Polzin and S. V. Daneshmand. A comparison of Steiner tree relaxations. *Discrete Applied Mathematics*, 112(1-3):241–261, 2001.

[39] C. Swamy and A. Kumar. Primal-dual algorithms for connected facility location problems. *Algorithmica*, 40:245–269, 2004.

[40] A. Tomazic and I. Ljubić. A GRASP algorithm for the connected facility location problem. In *Proceedings of 2008 International Symposium on Applications and the Internet (SAINT)*, pages 257–260. IEEE Computer Society, 2008.

[41] D. P. Williamson and A. van Zuylen. A simpler and better derandomization of an approximation algorithm for single source rent-or-buy. *Operations Research Letters*, 35(6):707–712, 2007.

[42] R. T. Wong. Integer programming formulations of the traveling salesman problem. *Proceedings of the IEEE international conference of circuits and computers*, pages 149–52, 1980.

[43] J. Xu, S. Y. Chiu, and F. Glover. Tabu search for dynamic routing communications network design. *Telecommunication Systems*, 8(1):55–77, 1997.

Layered Graph Approaches to the Hop Constrained Connected Facility Location Problem☆

Ivana Ljubić, Stefan Gollowitzer

Department of Statistics and Operations Research,
Faculty of Business, Economics, and Statistics,
University of Vienna, Austria

Abstract

Given a set of customers, a set of potential facility locations and some inter-connection nodes, the goal of the *Connected Facility Location* problem (ConFL) is to find the minimum-cost way of assigning each customer to exactly one open facility, and connecting the open facilities via a Steiner tree. The sum of costs needed for building the Steiner tree, facility opening costs and the assignment costs needs to be minimized. If the number of edges between a pre-specified node (the so-called root) and each open facility is limited, we speak of the Hop Constrained Facility Location problem (HC ConFL). This problem is of importance in the design of data-management and telecommunication networks.

In this article we provide the first theoretical and computational study for this new problem that has not been studied in the literature so far. We propose two disaggregation techniques that enable the modeling of HC ConFL: i) as a directed (asymmetric) ConFL on layered graphs, or ii) as the Steiner arborescence problem (SA) on layered graphs. This allows for usage of best-known MIP models for ConFL or SA to solve the corresponding hop constrained problem to optimality. In our polyhedral study, we compare the obtained models with respect to the quality of their LP lower bounds. These models are finally computationally compared in an extensive computational study on a set of publicly available benchmark instances. Optimal values are reported for instances with up to 1300 nodes and 115 000 edges.

Keywords: Hop constrained Minimum Spanning trees, Hop constrained Steiner trees, Connected Facility Location, Mixed Integer Programming Models, LP-relaxations

1. Introduction

The Connected Facility Location problem (ConFL) models a problem arising in the design of local access telecommunication networks, more precisely, *Fiber-to-the-Curb* (FTTC) networks. In an

Email addresses: ivana.ljubic@univie.ac.at (Ivana Ljubić), stefan.gollowitzer@univie.ac.at (Stefan Gollowitzer)

FTTC network, fiber optic cables run from a central office to a cabinet serving a neighborhood. End users connect to this cabinet using the existing copper connections. Expensive switching devices are installed in these cabinets. Telecommunication companies work rapidly on the expansion of local access networks by partially replacing the outdated copper technology using fiber optic cables. Thereby, the underlying network design problem consists of determining positions of cabinets, deciding to which cabinet customers are connected (via existing copper cables), and how to connect the cabinets among each other and to the *central office* (i.e., to the backbone network).

ConFL also has applications in the design of content distribution networks (CDN). There are two types of servers used by a CDN: *origin* and *replica servers* (see, e.g., [28]). An origin server stores the definitive version of the content. A replica server stores a copy of the content and may be used as a media server, web server or as a cache server. The origin server communicates with replica servers located in the network, in order to update the content stored therein. ConFL models the following network design problem in the context of CDNs: replica servers are to be located on a network that will cache information. Demand nodes make requests for the information. Each demand node is served from the one among the replica servers it can be assigned to at the least cost. Updates to the information on the servers are made over time. Every piece of information that is updated at a single server location, must also be updated at every other server on the network. Therefore, we are looking for a network that opens a set of facilities such that each demand node is assigned to exactly one facility and facilities can communicate to each other (and with the given origin server). In Krick et al. [19], the authors considered the unrooted ConFL variant in which a similar CDN problem arises *without* the existence of the origin server node.

If connection costs are non-negative, there always exists an optimal ConFL solution that obeys a tree structure. In such simply connected graphs, reliability against a single edge/node failure is not provided. More precisely, the probability that a session will be interrupted by a link/node failure increases with the number of links/nodes in the path between the root and an installed facility. In both CDN and telecommunication networks, economic arguments do not allow the installation of more survivable networks with higher edge/node connectivity. Since paths with fewer hops have a better performance, we model these reliability constraints by generalizing the ConFL problem to the Hop Constrained ConFL problem (HC ConFL).

Problem Definition. ConFL is closely related to the Steiner tree problem in graphs. Given a graph $G = (V, E)$ with costs on the edges, a set of *terminal nodes* $R \subset V$ and a set of intermediate (Steiner) nodes $V \setminus R$, recall that the Steiner tree problem consists of finding a subtree of G that connects all terminals at minimum cost. Thereby, Steiner nodes may be used to interconnect the terminals, if this would produce a cheaper solution.

Assuming that a root facility is given and needs to be open in any feasible solution, ConFL can now be stated as follows:

Definition 1 (Rooted ConFL)**.** We are given an undirected graph (V, E) with a disjoint partition $\{S, R\}$ of V with $R \subset V$ being the set of *customers*, $F \subseteq S$ the set of *facilities*, $S \setminus F$ the set of *Steiner nodes* and the *root* node $r \in F$. The set of edges is partitioned into the set of *core edges* $E_S \subseteq S \times S$ and *assignment edges* $E_R \subseteq F \times R$ ($E_R \cup E_S = E$, $E_R \cap E_S = \emptyset$). We are also given costs of core edges $c_e \geq 0$, $e \in E_S$, assignment costs $c_e \geq 0$, $e \in E_R$ and facility opening costs $f_i \geq 0$, $i \in F$. The root node is always considered as an open facility. The goal is to find a subset of open facilities such that:

- each customer is assigned to an open facility,

- a Steiner tree (consisting of core edges) connects all open facilities, and
- the sum of assignment, facility opening and Steiner tree costs is minimized.

If a facility node $i \in F$ is part of the core network without serving any customer, then i does not incur any opening costs and is considered as a Steiner node.

In the tree representing a feasible ConFL solution, the number of edges on the path between the root node and an open facility is usually called the number of *hops*. Based on this definition the *Hop Constrained Connected Facility Location Problem* is:

Definition 2 (HC ConFL)**.** Given an instance of the rooted ConFL, find a minimum-cost solution that is valid for ConFL and in which there are at most H hops between the root and any open facility.

An instance of HC ConFL is shown in Figure 1a). Figure 1b) illustrates a solution for $H \geq 2$. In this and all succeeding examples we use the following symbols: $\boxed{\text{r}}$ represents the root node, ○ represents a Steiner node. $\boxed{\text{i}}$ represents a facility i. ◇ represents a customer. In these examples the default edge/arc values, facility opening and assignment costs are all set to one. Costs different from one are displayed next to the respective arc/node. The core network is presented as an undirected graph.

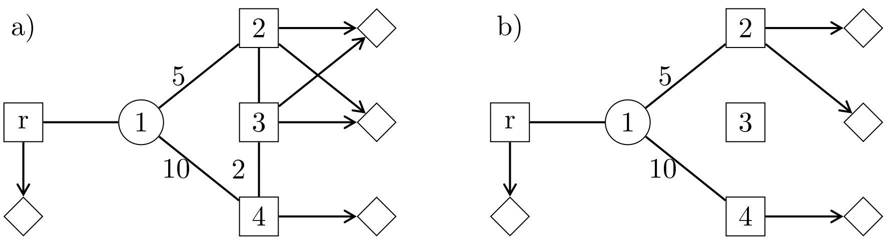


Figure 1: a) Original instance; b) Feasible solution.

Observation 1. *Using the transformation given in [11], any (HC) ConFL instance, in which $S \cap R \neq \emptyset$, can be transformed into an equivalent one such that $\{S, R\}$ is a proper partition of V.*

Computational Complexity of HC ConFL. A polynomial time algorithm M for an NP-hard minimization problem is an approximation algorithm with *approximation ratio* $\alpha > 1$ if for every instance I, $c(M(I)) \leq \alpha OPT(I)$, where $c(M(I))$ is the objective value of the solution $M(I)$, and $OPT(I)$ is the value of the optimal solution. APX is a class of NP-hard optimization problems for which there exist polynomial-time approximation algorithms with approximation ratio bounded by a constant.

Lemma 1. *HC ConFL ($H \geq 2$) is not in APX — it is at least $O(\log |V|)$-hard to approximate HC ConFL, unless $P = NP$. The result holds even if the edge weights are all equal to 1 ($c_e = 1$, for all $e \in E$) and, consequently, even if the edge weights satisfy the triangle inequality.*

Proof. See Appendix. □

Observe that HC ConFL becomes the uncapacitated facility location problem for $H = 1$: Steiner nodes can be removed, and weights of the edges between the root and each potential facility i can be incorporated into facility opening costs. Hence, if the edge weights satisfy the triangle inequality and $H = 1$, HC ConFL belongs to APX (see, e.g., an approximation algorithm given by Mahdian et al. [24]).

Our Contribution. We describe the Hop Constrained Connected Facility Location problem, that has not previously been considered in the literature. By extending the ideas given by Gouveia et al. [15] we propose two possibilities for modeling the HC ConFL: i) as a directed (asymmetric) ConFL on layered graphs, or ii) as the Steiner arborescence (SA) (i.e., a directed Steiner tree) problem on layered graphs. This allows for using the best-performing mixed integer programming (MIP) models for ConFL or SA in order to solve HC ConFL to optimality. Our layered graphs correspond to two different levels of disaggregation of MIP variables. In a polyhedral comparison we show that the strongest models on different layered graphs provide lower bounds of the same quality. Hence, we use the layered graph with less edges and facilities to conduct our computational study. In an extensive computational study, we compare the performance of several branch-and-cut algorithms developed to solve the proposed MIP models. This is a first theoretical and computational study on MIP models for this challenging combinatorial optimization problem.

Preliminary results of this paper appeared in the Proceedings of the International Symposium on Combinatorial Optimization (ISCO), 2010 ([22]).

The remainder of this paper is organized as follows: The following section provides a literature review on some problems related to HC ConFL. In Section 3 we describe MIP formulations for HC ConFL based on the concept of layered graphs. In Section 4 a polyhedral comparison of these formulations is given. Section 5 describes the implementation of branch-and-cut algorithms that are used to compare these models computationally. Section 5 also contains an extensive computational study conducted on a set of publicly available benchmark instances. In Section 6 we discuss variants of HC ConFL obtained from applications other than the ones arising in the telecommunications, and in Section 7 we provide some concluding remarks.

2. Literature Review

The Hop Constrained Connected Facility Location Problem is closely related to two well-known network design problems: the *Connected Facility Location* problem and the *Steiner tree problem with hop constraints.*

The Connected Facility Location problem. Early work on ConFL mainly includes approximation algorithms. The problem can be approximated within a constant ratio and the currently best-known approximation ratio is provided by [10]. Ljubić [21] describes a hybrid heuristic combining Variable Neighborhood Search with a reactive tabu search method. The author compares it with an exact branch-and-cut approach, using two new classes of test instances. Results for these instances with up to 1300 nodes are presented. Tomazic and Ljubić [30] present a Greedy Randomized Adaptive Search Procedure (GRASP) for the ConFL problem and results for a new set of test instances with up to 120 nodes. The authors also provide a transformation that enables solving ConFL as the Steiner arborescence problem. Bardossy and Raghavan [4] develop a dual-based local search (DLS) heuristic for a generalization of the ConFL problem. The presented DLS heuristic computes lower and upper bound using a dual-ascent and then improves the solution with a local search procedure. Computational results for instances with up to 100 nodes are presented. In Leitner and Raidl [20], the authors present a branch-and-cut-and-price approach for a variant of ConFL with *capacities on facilities.*

In [11] we study MIP formulations for ConFL, both theoretically and computationally. We provide a complete hierarchy of ten MIP formulations with respect to the quality of their LP-bounds. We describe two cut-set based formulations (among others) for the (directed) ConFL problem. The models differ in the way they make use of the connectivity concept. In the first one, connectivity is ensured between the root and any open facility, and additional assignment constraints are required between the facilities and customers. The second model uses cut-sets that ensure connectivity between the root and every customer. We show that the second model provides theoretically stronger lower bounds, but is outperformed by the first model in practice. In the computational study, instances with up to 1300 nodes and 115 000 edges have been solved to optimality using a branch-and-cut approach.

The Hop Constrained Steiner tree problem (HCSTP). In the hop constrained Steiner tree problem, the goal is to connect a given subset of customers at minimum cost, while using a subset of Steiner nodes, so that the number of hops between a root and each terminal does not exceed H. A large body of work has been done for the Hop Constrained Minimum Spanning Tree problem (HCMST), a special case of the HCSTP where each node in the graph is a terminal. A recent survey for the HCMST can be found in [9]. Gouveia et al. [15] use a reformulation on layered graphs to develop the strongest MIP models known so far for the HCMST.

Much less has been said about the Hop Constrained Steiner tree problem. The problem was first mentioned by [13], who develops a strengthened version of a multi-commodity flow model for the HCMST and HCSTP. The LP lower bounds of this model are equal to the ones from a Lagrangean relaxation approach of a weaker MIP model introduced in [12]. Results for instances with up to 100 nodes and 350 edges are presented.

[31] presents MIP formulations based on Miller-Tucker-Zemlin subtour elimination constraints. The models are then strengthened by disaggregation of variables indicating used arcs. The author develops a simple heuristic to find starting solutions and improves these with an exchange procedure based on tabu search. Numerical results are given for instances with up to 2500 nodes and 65 000 edges. [14] gives a survey of hop-indexed tree and flow formulations for the hop constrained spanning and Steiner tree problem.

Costa et al. [7] give a comparison of three heuristic methods for a generalization of the HCSTP, namely the Steiner tree problems with revenues, budget and hop constraints (STPRBH). The considered methods comprise a greedy algorithm, a destroy-and-repair method and a tabu search approach. Computational results are reported for instances with up to 500 nodes and 12 500 edges. In Costa et al. [8] the authors introduce two new MIP models for the STPRBH. They are both based on the generalized sub-tour elimination constraints and a set of hop constraints of exponential size. The authors provide a theoretical and computational comparison with the two models based on Miller-Tucker-Zemlin constraints presented in Voß [31] and Gouveia [14].

3. (M)ILP Formulations for HC ConFL

In this section we will show several ways of modeling HC ConFL as a mixed integer linear program. MIP formulations for trees on directed graphs often give better lower bounds than their undirected counterparts (see, e.g., [23]). By replacing each core edge e between nodes i and j from S by two directed arcs ij and ji and each assignment edge between a facility $i \in F$ and a customer $k \in R$ by an arc ik without changing the edge costs, undirected instances can be transformed into directed ones. In the remainder of this paper we will focus on the Hop Constrained Connected Facility Location problem on the directed graph $G = (V, A)$ obtained that way.

It is well-known that compact MIP formulations based on flow variables can be used to model hop constrained network design problems in general. In the case of HC ConFL, the corresponding flow-based models can be derived from the formulations for related hop constrained problems presented in Balakrishnan and Altinkemer [3], Gouveia [12] and Gouveia [13]. In this work, we are not going to consider such formulations. According to our computational experience for the much simpler ConFL problem (see, Gollowitzer and Ljubić [11]), flow-based MIP formulations are of limited usage if they are simply plugged into a MIP solver without using advanced decomposition techniques (e.g., column generation, Lagrangean relaxation or Benders decomposition). In this work we will use the cutting plane method as a decomposition technique for models with an exponential number of constraints. These models are developed on layered graphs that implicitly model hop constraints.

For comparison purposes, in Section 3.3 we will also present a three-index model with a polynomial number of variables and constraints. This model, according to our preliminary computational results, performs best in practice, as far as compact models are concerned.

Notation. To model the problem, we will use the following binary variables:

$$x_{ij} = \begin{cases} 1, & \text{if } ij \text{ belongs to the solution} \\ 0, & \text{otherwise} \end{cases} \quad \forall ij \in A \qquad z_i = \begin{cases} 1, & \text{if } i \text{ is open} \\ 0, & \text{otherwise} \end{cases} \quad \forall i \in F$$

Some of the MIP models provided below do not explicitly use variables $\mathbf{x}$ and $\mathbf{z}$. The variables are rather provided in an extended space of layered graphs, and the values of their corresponding counterparts are projected back into the space of $(\mathbf{x}, \mathbf{z})$.

We will use the following notation: $A_R = \{ij \in A \mid i \in F, j \in R\}$, $A_S = \{ij \in A \mid i, j \in S\}$. We will refer to A_R as *assignment arcs* and to A_S as *core arcs*. Consequently, subgraphs induced by A_R and A_S will be referred to as *core* and *assignment graph*, respectively. For any $W \subset V$ we denote by $\delta^-(W) = \{ij \in A \mid i \notin W, j \in W\}$, $\delta^+(W) = \{ij \in A \mid i \in W, j \notin W\}$ and $x(D) = \sum_{ij \in D} x_{ij}$, for every $D \subseteq A$.

Unless explicitly provided in the text below, the proofs of the lemmata in the subsequent sections are given in the Appendix.

3.1. Modeling Hop Constraints on Layered Graphs

We develop two variants of a layered graph to model HC ConFL as ConFL on a directed graph. In the first variant we build a layered graph, denoted by $LG_{x,z}$, by a disaggregation of both the core and the assignment graph. In the second variant we transform only the core graph into the layered graph, define nodes at the level H as potential facilities and leave the assignment graph unchanged. We denote this graph by LG_x.

3.1.1. Layered Core and Assignment Graph $LG_{x,z}$

Consider a graph $LG_{x,z} = (V_{x,z}, A_{x,z})$ defined as an instance of directed ConFL with the set of potential facilities $F_{x,z}$ and the set of core nodes $S_{x,z}$ given as follows:

$$
\begin{aligned}
V_{x,z} &:= \{r\} \cup S_{x,z} \cup R \quad \text{where} \quad && F_{x,z} = \{(i,p) : i \in F \setminus \{r\}, 1 \le p \le H\}, \\
& && S_{x,z} = F_{x,z} \cup \{(i,p) : 1 \le p \le H-1, i \in S \setminus F\} \text{ and} \\
A_{x,z} &:= \bigcup_{i=1}^{5} A_i \quad \text{where} \quad && A_1 = \{(r,(j,1)) : rj \in A_S\}, \\
& && A_2 = \{((i,p),(j,p+1)) : 1 \le p \le H-2, ij \in A_S\}, \\
& && A_3 = \{((i,H-1),(j,H)) : ij \in A_S, i \in S \setminus \{r\}, j \in F \setminus \{r\}\}, \\
& && A_4 = \{rk : rk \in A_R\} \\
& && A_5 = \{((i,p),k) \mid ik \in A_R, (i,p) \in F_{x,z}, k \in R\}.
\end{aligned}
$$

The cost of an arc from $A_1 \cup A_2 \cup A_3$ and $A_4 \cup A_5$ is set to the cost of the corresponding arc from A_S and A_R, respectively. The facility opening costs are f_i for all (i,p) with $p = 1, \ldots, H$, $i \in F \setminus \{r\}$. A node (i,p) will also be referred to as "node i at level p".

Lemma 2. *Given the graph transformation from G to $LG_{x,z}$ described above, there always exists an* optimal *solution of the directed ConFL on $LG_{x,z}$ that can be transformed into a ConFL solution on G with at most H hops and the same cost. Conversely, every* feasible *HC ConFL solution on G corresponds to a directed ConFL solution on $LG_{x,z}$.*

Figure 2 illustrates the layered graph $LG_{x,z} = (V_{x,z}, A_{x,z})$: Figure 2a) shows the complete layered graph $LG_{x,z} = (V_{x,z}, A_{x,z})$ for the instance depicted in Figure 1a) and $H = 3$; Figure 2b) shows the layered graph after the preprocessing; The optimal solution on $LG_{x,z}$ is shown in Figure 2d). The projection onto the original graph $G = (V, A)$ of the solution in d) is shown in Figure 1b).

Preprocessing. The following three preprocessing steps may significantly reduce the size of a layered graph.

1. Without loss of generality, all arcs $((j,p),k)$ with $j \in F \setminus \{r\}$ and $k \in R$ such that $c_{rk} < c_{jk}$ can be removed from $LG_{x,z}$, for all $p = 1, \ldots, H$.
2. A node $(i,p) \in S_{x,z}$ whose in-degree is zero, can be removed from $LG_{x,z}$. The removal is performed starting from level 1 to H.
3. A node $(i,p) \in S_{x,z}$ whose out-degree is zero, cannot be part of any cost-optimal solution to ConFL on $LG_{x,z}$. The removal of those redundant nodes is performed starting from level H to 1.

We perform these steps iteratively in the order given above.

We will associate binary variables to the arcs in $A_{x,z}$ as follows: X^1_{rj} corresponds to $(r,(j,1)) \in A_1$, X^p_{ij} to $((i,p-1),(j,p)) \in A_2$, X^H_{ij} to $((i,H-1),(j,H)) \in A_3$, X^1_{rk} to $rk \in A_4$ and X^p_{ik} corresponds to $((i,p),k) \in A_5$.

Let $X[\delta^-(W)]$ denote the sum of all variables $\mathbf{X}$ in the cut $\delta^-(W)$ in $LG_{x,z}$ defined by $W \subseteq V_{x,z} \setminus \{r\}$. In Gollowitzer and Ljubić [11] we describe two cut-set based formulations for the

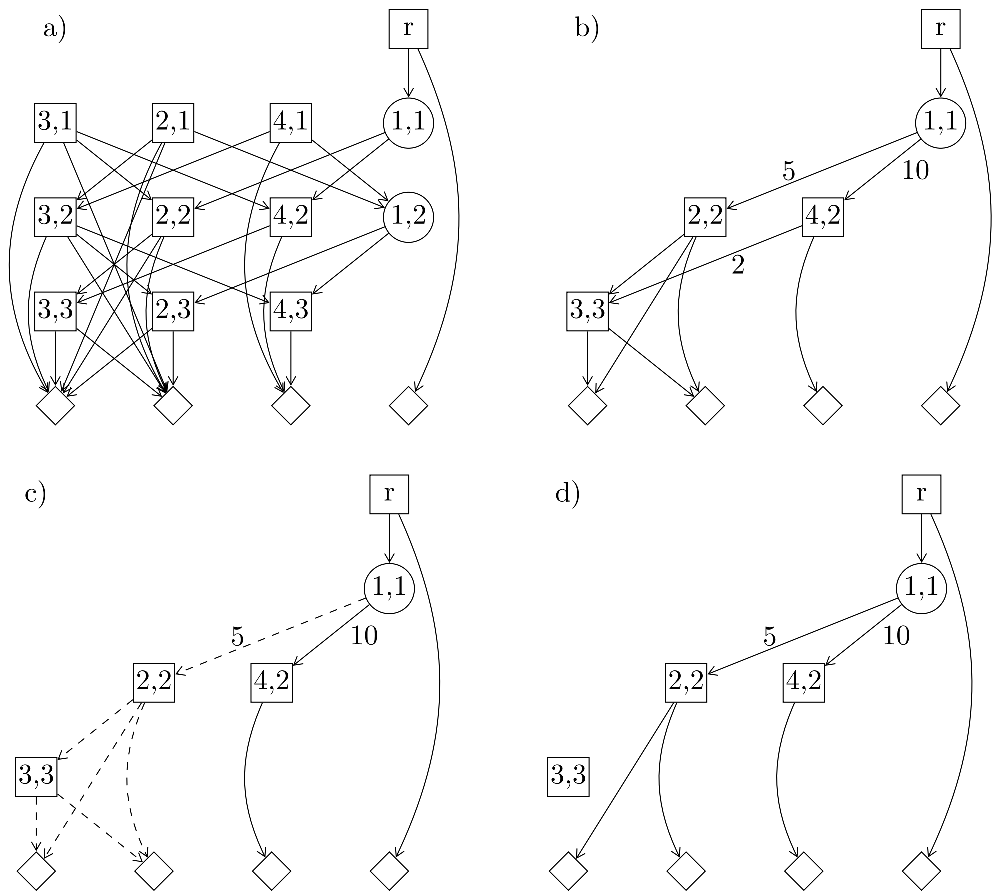


Figure 2: $LG_{x,z}$ for the example shown in Figure 1a) and $H = 3$: a) $LG_{x,z}$ before, and b) after preprocessing; c) The optimal LP-solution for $CUT^F_{x,z}$ – dashed and solid arcs take LP-value of 1/2 and 1, respectively; d) The optimal LP-solution for CUT^R_x which is already MIP-optimal; Figure 1b) shows the projection of the solution in e) back onto the original graph.

(directed) ConFL problem. In the model called CUT_F, connectivity is ensured between the root and any open facility, and additional assignment constraints are required between the facilities and customers. The second model, referred to as CUT_R, uses cut-sets that ensure connectivity between the root and every customer.

We now use these two models to derive corresponding cut-set formulations on $LG_{x,z}$, denoted by $CUT^F_{x,z}$ and $CUT^R_{x,z}$. For notational convenience we will also introduce the following variables:

- X^p_{ri}, for $ri \in A$, $p = 2, \dots, H$,
- X^1_{ij} for $ij \in A_S$, $i \neq r$, and
- X^H_{ij} for $ij \in A_S$, $j \in S \setminus F$.

These variables will be fixed to zero (see constraints (5) below).

Connectivity Cuts Between Root and Facilities. The model $CUT^F_{x,z}$ reads as follows:

$$(CUT^F_{x,z}) \quad \min \sum_{ij \in A} c_{ij} \sum_{p=1}^{H} X^p_{ij} + \sum_{i \in F \setminus \{r\}} f_i \sum_{p=1}^{H} Z^p_i + f_r z_r$$

$$X[\delta^-(W)] \geq Z^p_i \qquad \forall W \subseteq S_{x,z} \setminus \{r\},\ (i,p) \in F_{x,z} \cap W \tag{1}$$

$$\sum_{jk \in A_R} \sum_{p=1}^{H} X^p_{jk} = 1 \qquad \forall k \in R \tag{2}$$

$$X^p_{jk} \leq Z^p_j \qquad \forall jk \in A_R,\ p = 1, \ldots, H,\ j \neq r \tag{3}$$

$$z_r = 1 \tag{4}$$

$$X^p_{ij} = 0 \qquad ij \in A, \begin{cases} i = r,\ p = 2, \ldots, H \\ i \neq r,\ j \notin R,\ p = 1 \\ j \in S \setminus F,\ p = H \end{cases} \tag{5}$$

$$X^p_{ij} \in \{0,1\} \qquad \forall ij \in A, p = 1, \ldots, H \tag{6}$$

$$Z^p_i \in \{0,1\} \qquad \forall (i,p) \in F_{x,z} \tag{7}$$

Constraints (1) are *connectivity cuts* on $LG_{x,z}$ between the root r and each open facility i at a level p, $(i,p) \in F_{x,z}$. Equalities (2) are *assignment constraints.* They ensure that each customer $k \in R$ is assigned to exactly one facility from $F_{x,z} \cup \{r\}$. Inequalities (3) are *coupling constraints* - they necessitate a facility j at a level p to be open if a customer is assigned to it. Equation (4) forces the facility at the root node to be open. In this model, both arc- and facility variables are disaggregated, and their projection into the space of $(\mathbf{x}, \mathbf{z})$ variables is given as: $x_{ij} := \sum_{p=1}^{H} X^p_{ij}$, for all $ij \in A$ and $z_i := \sum_{p=1}^{H} Z^p_i$, for all $i \in F \setminus \{r\}$.

One observes that, since $f_i \geq 0$ for all $i \in F \setminus \{r\}$ and $c_{ij} \geq 0$ for all $ij \in A_R$, there always exists an optimal solution on $LG_{x,z}$ that also satisfies

$$\sum_{p=1}^{H} Z^p_i \leq 1 \qquad \forall i \in F \setminus \{r\}.$$

The validity of this claim follows from Lemma 10 (see Appendix) and from the fact that for each $i \in F$, $Z^p_i \leq X[\delta^-(\{(i,p)\})]$, for all $p = 1, \ldots, H$. Consequently, we can show the following

Lemma 3. *In the model $CUT^F_{x,z}$, connectivity cuts* (1) *can be replaced by the following stronger ones:*

$$X[\delta^-(W)] \geq \sum_{p=1}^{H} Z^p_i \qquad \forall W \subseteq S_{x,z} \setminus \{r\},\ i \in F \setminus \{r\} \tag{8}$$

Proof. For all $i \in F$ each facility in the corresponding set of facility nodes, $F_i = \{(i,p) \mid p = 1, \ldots, H\}$, in $LG_{x,z}$ serves the same subset of customers with the same assignment costs. Therefore, there always exists an optimal solution for which at most one among the facilities of the same group F_i is opened, which explains the validity of these constraints. □

The new MIP formulation, in which (1) is replaced by (8) will be denoted by $CUT^{F+}_{x,z}$.

Connectivity Cuts Between Root and Customers. By replacing (1) and (2) in the model $CUT^F_{x,z}$ with the following inequalities,

$$X[\delta^-(W)] \geq 1 \qquad \forall W \subseteq V_{x,z} \setminus \{r\}, W \cap R \neq \emptyset, \tag{9}$$

we obtain a new model that we denote by $CUT^R_{x,z}$.

Inequalities (9) are connectivity cuts on $LG_{x,z}$ between sets containing the root and a customer respectively. Our study on ConFL in Gollowitzer and Ljubić [11] has shown that these connectivity constraints ensure stronger lower bounds than the bounds obtained using the connectivity cuts between the root and facilities.

In a recent study by Gouveia, Simonetti, and Uchoa [15], it has been shown that cut-set based MIP models on layered graphs represent the tightest formulations known so far for modeling the HCMST. In a similar way, one can show that the same holds for HC ConFL. Layered graph models dominate not only extended formulations (derived by using flow variables, hop-indexed trees or Miller-Tucker-Zemlin constraints mentioned above), but also formulations projected in the space of $(\mathbf{x}, \mathbf{z})$ variables based on exponentially many *path* or *jump* inequalities (see Costa et al. [8] and Dahl et al. [9], respectively).

3.1.2. Layered Core Graph LG_x

In this section we will show an alternative way of building a layered graph to model the HC ConFL problem. In this new layered graph only the core network will be disaggregated while the assignment graph will be left unchanged. Consider a graph $LG_x = (V_x, A_x)$ representing an instance of directed ConFL with the set of customers R defined as above and the set of potential facilities F_x and the set of core nodes S_x defined as follows:

$$\begin{aligned}
V_x &:= \{r\} \cup S_x \cup R \quad \text{where} \quad && F_x = \{(i,H) : i \in F \setminus \{r\}\}, \\
& && S_x = F_x \cup \{(i,p) : 1 \leq p \leq H-1, i \in S \setminus \{r\}\} \text{ and} \\
A_x &:= \bigcup_{i=1}^{4} A_i \cup A_6 \cup A_7 \quad \text{where} \quad && A_1, A_2, A_3 \text{ and } A_4 \text{ are defined as for } A_{x,z}, \\
& && A_6 = \{((i,p),(i,H)) : 1 \leq p \leq H-1, i \in F \setminus \{r\}\} \text{ and} \\
& && A_7 = \{((j,H),k) : jk \in A_R, j \neq r\}
\end{aligned}$$

The facility opening and assignment costs are left unchanged. Set $A_{S_x} := A_1 \cup A_2 \cup A_3 \cup A_6$ determines the *layered core graph.* The cost of an arc from $A_1 \cup A_2 \cup A_3$ and $A_4 \cup A_7$ is set to the cost of the corresponding arc from A_S and A_R, respectively. Arcs between (i,p) and (i,H) are assigned costs of 0 for all $p = 1, \ldots, H-1$ and $i \in F$.

One observes that the preprocessing rules explained for $LG_{x,z}$ also apply to LG_x and we can show the following

Lemma 4. *Given the graph transformation from G to LG_x described above, there always exists an* optimal *solution of the directed ConFL on LG_x that can be transformed into a ConFL solution on G with at most H hops and the same cost. Conversely, every* feasible *HC ConFL solution on G corresponds to a directed ConFL solution on LG_x.*

Figure 3 illustrates the transformation of an original HC ConFL instance given in Figure 1a) into an instance for directed ConFL on LG_x, before and after preprocessing.

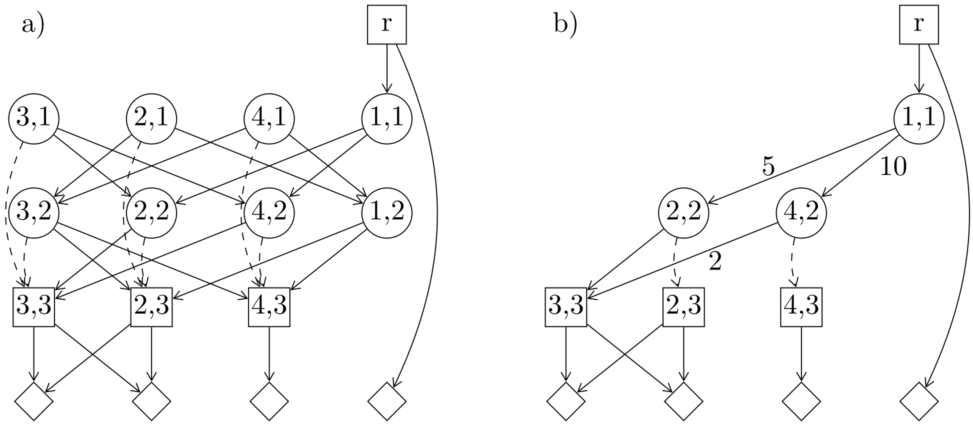


Figure 3: Layered graph LG_x for the instance given in Figure 1a) obtained a) before and b) after preprocessing.

We will associate binary variables to the arcs in A_x as follows: X^1_{rj} corresponds to $(r,(j,1)) \in A_1$, X^p_{ij} to $((i,p-1),(j,p)) \in A_2$, X^H_{ij} to $((i,H-1),(j,H)) \in A_3$, X^p_{ii} to $((i,p-1),(i,H)) \in A_6$. Again, for notational convenience, we will also introduce the following binary variables:

- X^p_{ri}, for $ij \in A_S$, $p = 2,\dots,H$, and
- X^1_{ij}, for $ij \in A_S$, $i \neq r$

and fix them to zero. Since the assignment graph is left unchanged, we will associate the corresponding $\mathbf{x}$ variables to the assignment graph in LG_x, i.e.: x_{jk} to $((j,H),k) \in A_7$ and x_{rk} to $rk \in A_4$. For the same reason, we link binary variables z_i to each (i,H) in F_x. The corresponding projection of a feasible solution $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$ into the space of $(\mathbf{x}, \mathbf{z})$ variables is given as follows: $x_{ij} := \sum_{p=1}^{H} X'^p_{ij}$ for all $ij \in A_S$, $x_{jk} := x'_{jk}$ for all $jk \in A_R$ and $z_i := z'_i$ for all $i \in F$.

Connectivity Cuts Between Root and Facilities/Customers. Let $X_x[\delta^-(W)]$ denote the sum of all $\mathbf{X}$ and $\mathbf{x}$ variables in the cut $\delta^-(W)$ in LG_x defined by $W \subseteq V_x \setminus \{r\}$. We now develop the MIP model for directed ConFL on LG_x with connectivity cuts involving node-variables as follows:

$$(CUT^F_x)\ \min \sum_{ij \in A_S} c_{ij} \sum_{p=1}^{H} X^p_{ij} + \sum_{jk \in A_R} c_{jk} x_{jk} + \sum_{i \in F} f_i z_i$$

$$X_x[\delta^-(W)] \geq z_i \qquad \forall W \subseteq S_x \setminus \{r\},\ W \cap F_x \neq \emptyset \tag{10}$$

$$\sum_{jk \in A_R} x_{jk} = 1 \qquad \forall k \in R \tag{11}$$

$$x_{jk} \leq z_j \qquad \forall jk \in A_R \tag{12}$$

$$X^p_{ij} = 0 \qquad ij \in A_S, \begin{cases} i = r,\ p = 2,\dots,H \\ i \neq r,\ p = 1 \end{cases} \tag{13}$$

$$z_r = 1 \tag{14}$$

$$X^p_{ij} \in \{0,1\} \qquad ij \in A_S, p = 1,\dots,H \tag{15}$$

$$z_i \in \{0,1\} \qquad \forall i \in F \setminus \{r\} \tag{16}$$

$$x_{jk} \in \{0,1\} \qquad \forall jk \in A_R \tag{17}$$

Constraints (10) are connectivity cuts on LG_x between sets containing the root and a facility i respectively. Equations (11) are the assignment constraints, and inequalities (12) are the coupling constraints.

Similarly, if we now replace constraints (10) and (11) by the following ones, we obtain a stronger formulation that we denote by CUT_x^R:

$$X_x[\delta^-(W)] \geq 1 \quad \forall W \subseteq V_x \setminus \{r\}, W \cap R \neq \emptyset \tag{18}$$

One observes that, if constraints (17) are relaxed to $x_{jk} \geq 0$, for all $jk \in A_R$, the optimal solution remains integral. Although constraints (11) are redundant (provided that the vectors $\mathbf{c}$ and $\mathbf{f}$ in the objective function are non-negative), we will explicitly use them in the computational study given in Section 5.

3.2. Modeling HC ConFL as Steiner Arborescence on Layered Graphs

In general, every (directed) ConFL problem can be modeled as the Steiner arborescence problem (see [11]). The transformation works as follows: Each potential facility node i is split into i and i' and replaced by a directed arc from i to i' of cost f_i. Assignment arcs $ik \in A_R$ are then replaced by $i'k$. That way, by solving the Steiner arborescence problem on the transformed graph, we distinguish between the following two situations:

1. arc ii' is taken into a Steiner arborescence, i.e., the potential facility node i is used as an open facility in a ConFL solution, or

2. only node i is taken into a Steiner arborescence, i.e., i is used only as a Steiner node in the corresponding ConFL solution.

Hence, by applying this transformation to both LG_x and $LG_{x,z}$ we can reformulate the HC ConFL as the Steiner arborescence problem on even larger layered graphs. This transformation increases the number of nodes by $|F|$, but does not provide stronger lower bounds for the corresponding cut-set formulation (see [11]).

Steiner Arborescence Model on LG_x. We now show an alternative and simpler way of modeling HC ConFL as the Steiner arborescence problem on the layered graph LG_x. The main difference between ConFL and the (node-weighted) Steiner tree problem is that it is not known in advance whether the opening costs of a potential facility node are going to be paid or whether it will be used only as a Steiner node. However, looking at LG_x, one observes that in any optimal solution of the directed ConFL on LG_x, the only Steiner nodes that are taken into an optimal solution are at levels $1, \ldots, H-1$. In other words, if a facility node (i, H) belongs to an optimal solution, it serves only to connect the root with a customer, i.e., every node (i, H) that belongs to an optimal solution is an open facility. Because the in-degree of every (facility) node in an optimal solution is at most one, facility opening costs can now be integrated into ingoing arcs as follows:

- for each arc from A_{S_x} connecting a node $(j, H-1)$ to (i, H) we set its cost to $c_{ji} + f_i$
- for each arc from A_{S_x} connecting a node (i, p) $(1 \leq p \leq H-1)$ to (i, H) we set its cost to f_i.

We will denote the layered graph LG_x with the new cost structure as LG_{sa}.

Lemma 5. *Every optimal solution of the Steiner arborescence problem on LG_{sa}, with R being the set of terminals, can be transformed into a ConFL solution on G with at most H hops that incurs the same cost. Conversely, every feasible HC ConFL solution on G corresponds to a Steiner arborescence solution on LG_{sa}.*

The corresponding MIP model reads then as follows:

$$(CUT_{sa}) \ \min \sum_{ij \in A_S} c_{ij} \sum_{p=1}^{H-1} X_{ij}^p + \sum_{jk \in A_R} c_{jk} x_{jk} + \sum_{i \in F} f_i \sum_{p=1}^{H-1} X_{ii}^p + \sum_{ij \in A_S, j \in F} (c_{ij} + f_j) X_{ij}^H + f_r$$
$$(11), (13), (15), (17), (18)$$

One observes that the given transformation works only for the graph LG_x, but not for $LG_{x,z}$. In Section 5, we will provide computational results for the given cut-set formulation CUT_{sa}.

3.3. Hop-indexed Tree Formulations

The following three-index model can be seen as a compact MIP formulation for HC ConFL on LG_x. A hop-indexed tree model has been originally proposed by [14] for solving the Hop Constrained STP. [31] has observed that this formulation is a disaggregation of a formulation based on Miller-Tucker-Zemlin constraints. Costa et al. [8] have extended this model with valid inequalities to solve the hop constrained STP with profits. We will now extend the ideas of using the hop-indexed tree variables to model HC ConFL. We model constraints for core and assignment graphs separately. Variables X_{ij}^p indicate whether an arc $ij \in A_S$ is used at the p-th position from the root node. Variables x_{jk} indicate whether customer $k \in R$ is assigned to facility $j \in F$. We link core and assignment graphs by variables z_j, indicating whether a facility is installed on node $j \in F$. Using the variables described above we can formulate the HC ConFL as follows:

$$(HOP) \quad \min \sum_{p=1}^{H} \sum_{ij \in A_S} c_{ij} X_{ij}^p + \sum_{jk \in A_R} c_{jk} x_{jk} + \sum_{i \in F} f_i z_i$$
$$\sum_{\substack{i \in S \setminus \{k\}: \\ ij \in A_S}} X_{ij}^{p-1} \geq X_{jk}^p \qquad \forall jk \in A_S, \ j \neq r, \ p = 2, \dots, H \tag{19}$$
$$\sum_{ij \in A_S} \sum_{p=1}^{H} X_{ij}^p \geq z_j \qquad \forall j \in F \setminus \{r\} \tag{20}$$
$$(11) - (17)$$

Constraints (19) are connectivity constraints given in a compact way – comparing HOP with the model CUT_x^F, we observe that the former one is obtained by replacing constraints (10) by (19) and (20). Constraints (19) ensure that for every arc on level p leaving out a node j, there is at least one arc at the level $p-1$ entering j. Similarly, inequalities (20) link opening facilities to their in-degree, i.e., if facility j is open, at least one of the arcs on levels $p \in \{1, \dots, H\}$ needs to enter it. Using the same arguments as for the construction of the graph LG_{sa}, one could replace inequalities in (20) by equations, and consequently eliminate $\mathbf{z}$ variables.

To model HC ConFL, there are actually two options for the hop-indexed variables. We propose to separate core and assignment graph and link them by the $\mathbf{z}$-variables indicating the use of facilities. Alternatively, we can define hop-indexed variables on the whole graph G, modeling connectivity between the root and each customer node. It can be shown that the latter model in which hop-indexed variables are introduced for both, the core and assignment graph, provides the same lower bounds as the model *HOP*, while exhibiting a much larger number of variables and constraints. Hence, this alternative approach will not be considered throughout this paper.

4. Polyhedral Comparison

In this section we provide a theoretical comparison of the MIP models described above with respect to the optimal values of their LP-relaxations. Denote by $\mathcal{P}_{.}$ the polytope and by $v_{LP}(.)$ the value of the LP-relaxation of any of the MIP models described above. We call a formulation R_1 stronger than a formulation R_2 if the optimal value of the LP-relaxation of R_1 is no less than that of R_2 for all instances of the problem. If R_2 is also stronger than R_1, we call them equivalent, otherwise we say that R_1 is strictly stronger than R_2. If neither is stronger than the other one, they are incomparable.

Lemma 6. *Formulation $CUT^{F+}_{x,z}$ is strictly stronger than formulation $CUT^{F}_{x,z}$. Furthermore, there exist HC ConFL instances for which $\frac{v_{LP}(CUT^{F+}_{x,z})}{v_{LP}(CUT^{F}_{x,z})} \approx H - 1$.*

Lemma 7. *Formulation CUT^{F}_{x} is strictly stronger than formulation HOP.*

Lemma 8. *The following results hold:*

1. *The formulation CUT^{R}_{x} is strictly stronger than CUT^{F}_{x}. Furthermore, there exist HC ConFL instances such that $\frac{v_{LP}(CUT^{R}_{x})}{v_{LP}(CUT^{F}_{x})} \approx |F| - 1$.*

2. *The formulation $CUT^{R}_{x,z}$ is strictly stronger than $CUT^{F}_{x,z}$. Furthermore, there exist HC ConFL instances such that $\frac{v_{LP}(CUT^{R}_{x,z})}{v_{LP}(CUT^{F}_{x,z})} \approx (|F| - 1)|H|$.*

Lemma 9. *Formulations $CUT^{R}_{x,z}$ and CUT^{R}_{x} are equivalent.*

The table in Figure 4 gives an overview of the models described and the chart illustrates their relationships as shown in this section. In the chart an arrow denotes that the LP bound of its apex is greater or equal than the LP bound of its origin. A grey background indicates the formulations tested in the computational experiments whose results are provided in the next section.

HOP	Compact formulation on graph LG_x
CUT^{F}_{x}	Facility-based cuts on graph LG_x
CUT^{R}_{x}	Customer-based cuts on graph LG_x
$CUT^{F}_{x,z}$	Facility-based cuts on graph $LG_{x,z}$
$CUT^{F+}_{x,z}$	Stronger facility-based cuts on graph $LG_{x,z}$
$CUT^{R}_{x,z}$	Customer-based cuts on graph $LG_{x,z}$
CUT_{sa}	Customer-based cuts on graph LG_{sa}

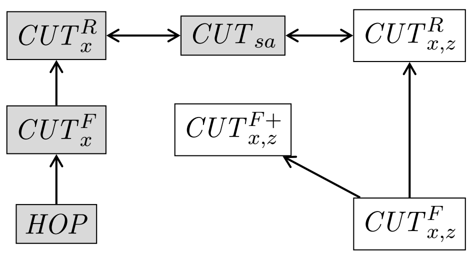


Figure 4: Summary and relationships between the LP lower bounds of the presented formulations.

5. Computations

In this section we present a computational comparison of the MIP models for solving HC ConFL given above. According to Lemma 9 and the theoretical analysis given in the previous section, transformations of G into $LG_{x,z}$ and LG_x provide the two strongest MIP formulations known until know. These formulations have the same quality of lower bounds. Therefore, we concentrate on models derived from the layered graph LG_x, which comprises a smaller number of edges and facilities. The computational comparison is conducted on three branch-and-cut (B&C) algorithms derived for MIP models with an exponential number of variables, and on one compact model, *HOP* (cf. Section 3.3).

5.1. Branch-and-Cut: Implementation Details

We implemented B&C algorithms for solving HC ConFL using the following MIP models: CUT_x^F, CUT_x^R and CUT_{sa}. The ingredients of our branch-and-cut schema are outlined below. We used the commercial package IBM CPLEX (version 11.2) and IBM Concert Technology (version 2.7), for solving the LP-relaxations, as well as a generic implementation of the branch-and-cut approach. All experiments were performed on a Intel Core2 Quad 2.33 GHz machine with 3.25 GB RAM, where each run was performed on a single processor. Separated cut-set inequalities are treated globally. The separation routine is called at every node of the B&C tree.

Initialization. Each branch-and-cut algorithm is initialized with the assignment and coupling constraints, (11) and (12), respectively. In addition, the following *flow-balance* inequalities are used. Let $X_x[\delta^+(W)]$ denote the sum of all variables X_{ij}^p in the cut $\delta^+(W)$ in LG_x defined by $W \subseteq S_x \backslash \{r\}$. The flow-balance inequalities ensure that Steiner nodes $i \in S_x$ cannot be leaves in the core graph:

$$X_x[\delta^-(\{i\})] \leq X_x[\delta^+(\{i\})] \quad \forall i \in S_x.$$

These inequalities are also known to strengthen the quality of lower bounds of cut-based models in general (see, e.g., [16]).

Separation. Separation of cut-set inequalities (10) and (18) is done in polynomial time by running the maximum-flow algorithm of Cherkassky and Goldberg [6] on the corresponding support graphs. In the case of inequalities (10), the maximum flow is calculated between the root node and any facility i such that $z_i > 0$. Inequalities (18) are separated by calculating the flow between the root and any customer $j \in R$. Separation is performed at each node of the branch-and-cut tree.

Since the computation of an LP-relaxation may be a time-consuming task, and the maximum-flow computation can be performed relatively efficiently, we would like to detect more than one violated inequality each time the separation routine is executed. To this end, we use the techniques of *nested* and *backward cuts* which are described below.

Nested cuts: Each time a violated cut-set is detected, we update the capacities on the links of that set and re-run the maximum flow algorithm in order to find the next violated inequality with a disjoint set of variables. This process is repeated until a maximal allowed number of cuts (M_{cut}) is inserted or until no more violated cuts are found. At the end of this process, the LP-relaxation of the problem with the newly added set of inequalities is resolved.

Backward cuts: Once the maximum flow on a graph is calculated, we are able to detect up to two different minimum cuts induced by the flow. More precisely, the maximum flow algorithm of Cherkassky and Goldberg labels the nodes with three labels: "l_r" - reachable from the root node,

"l_t" - reachable from the target node, and "l_0" - not reachable. All nodes labeled by l_r form a cut set L_r such that outgoing arcs are saturated by the flow, while all arcs into L_r are completely unused. Similarly, the nodes labeled by l_t form a cut set L_t such that all ingoing arcs are saturated by the flow and all outgoing arcs are completely unused. Hence, the first minimum cut is obtained by running the breath-first search (BFS) starting from the root and visiting all nodes labeled by l_r, the other one is obtained by running the BFS starting from the target and visiting all nodes labeled by l_t. Those two cut-sets are identical only in the case that the minimum cut in the graph is unique (in which case none of the nodes is labeled l_0).

The two features, nested and backward cuts are combined with each other, i.e., we are "nesting" both, forward and backward cuts.

Finally, in order to favor sparse cuts, we add a small ϵ value to the capacity of each arc, before running the maximum-flow algorithm. Hence, in the case of several minimum cuts of the same weight, the ones with the least number of variables will be detected earlier.

Figures 5 and 6 show the pseudo-codes of the separation algorithms for detecting violated inequalities of type (10) and (18), respectively. Thereby, *Pool* represents a pool of valid inequalities that are added to the LP-model at the end of the separation procedure. For a directed graph G with non-negative arc capacities c, the procedure MAXFLOW(G, c, r, i) returns the value of the maximum r-i flow. For a directed graph G with the flow f, procedures FORWARD() and BACKWARD() return the set of arcs composing the forward and backward minimum cuts as described above.

Algorithm 5.1: FACILITYCUTS(LG_x^{core}, X, z)

for each $a \in A_{S_x}$
  do $c_a \leftarrow X_a + \epsilon$
$Pool \leftarrow \emptyset$
for each $i \in F_x$ s.t. $z_i > 0$
  do {
    $f \leftarrow$ MAXFLOW(LG_x^{core}, c, r, i)
    while $f < z_i$ and $|Pool| < M_{cut}$
      do {
        $A_{fw} \leftarrow$ FORWARD(f)
        $A_{bw} \leftarrow$ BACKWARD(f)
        $Pool \leftarrow Pool \cup \{X_x[A_{fw}] \geq z_i\} \cup \{X_x[A_{bw}] \geq z_i\}$
        for each $a \in A_{fw} \cup A_{bw}$
          do $c_a \leftarrow \infty$
        $f \leftarrow$ MAXFLOW(LG_x^{core}, c, r, i)
      }
  }
Add *Pool* to the LP and resolve it.

Figure 5: Pseudo-code for separating inequalities (10) on the graph LG_x^{core}.

Observe that the separation of (10) is done using only the core of the layered graph $LG_x^{core} = (S_x, A_x)$ while the separation of (18) is conducted on the whole layered graph LG_x.

Branching and Enumeration. Among all binary variables, the biggest influence on the structure of the solution is due to facility location variables z_i. Therefore, in our default branch-and-bound implementation, the highest branching priority is assigned to these variables. The default enumeration strategy of CPLEX is used.

Algorithm 5.2: CustomerCuts(LG_x, X)

$$
\begin{array}{l}
\textbf{for each } a \in A_x \\
\quad \textbf{do } c_a \leftarrow X_a + \epsilon \\
Pool \leftarrow \emptyset \\
\textbf{for each } j \in R \\
\quad \textbf{do } \left\{
\begin{array}{l}
f \leftarrow \text{MAXFLOW}(LG_x, c, r, j) \\
\textbf{while } f < 1 \text{ and } |Pool| < M_{cut} \\
\quad \textbf{do } \left\{
\begin{array}{l}
A_{fw} \leftarrow \text{FORWARD}(f) \\
A_{bw} \leftarrow \text{BACKWARD}(f) \\
Pool \leftarrow Pool \cup \{X_x[A_{fw}] \geq 1\} \cup \{X_x[A_{bw}] \geq 1\} \\
\textbf{for each } a \in A_{fw} \cup A_{bw} \\
\quad \textbf{do } c_a \leftarrow \infty \\
f \leftarrow \text{MAXFLOW}(G, c, r, j)
\end{array}
\right.
\end{array}
\right. \\
\text{Add } Pool \text{ to the LP and resolve it.}
\end{array}
$$

Figure 6: Pseudo-code for separating inequalities (18) on the graph LG_x^{core}.

5.2. Data Set

We consider a class of benchmark instances, originally introduced in Ljubić [21], and also used by Tomazic and Ljubić [30] and Bardossy and Raghavan [4]. The ConFL instances are obtained by merging data from two public sources. In general, one combines an instance for the Uncapacitated Facility Location problem (UFLP) with an STP instance, to generate ConFL input graphs in the following way: Nodes indexed by $1, \ldots, |F|$ in the STP instance are selected as potential facility locations, and the node with index 1 is selected as the root. The number of facilities, the number of customers, opening costs and assignment costs are provided in UFLP files. STP files provide edge costs and Steiner nodes.

- As UFLP data we chose a set of non-trivial instances from UflLib (see [27]): Instances `mp{1,2}` and `mq{1,2}` have been proposed by [18]. They are designed to be similar to UFLP real-world problems and have a large number of near-optimal solutions. There are 6 classes of problems, and for each problem $|F| = |R|$. We took 2 representatives per each of the 2 classes `mp` and `mq`. The instances from `mp` are of size 200×200 and the ones from `mq` are of size 300×300.
- As STP instances we chose a set from the OR-library (see [5]): Instances `{c,d}n`, for $n \in \{5, 10, 15, 20\}$ were chosen as representatives of medium size instances for STP. These instances define the core networks with between 500 and 1000 nodes and with up to 25 000 edges.

For the instances described above Table 1 shows: the name of the original STP and UFLP instance it is derived from; the number of customers ($|R|$); the number of facilities ($|F|$), the number of nodes in the core graph ($|V \setminus R|$); the number of edges in the core graph ($|E_S|$) and the number of assignment edges ($|E_R|$). Combined with assignment graphs, the largest instances of this data set contain 1300 nodes and 115 000 edges.

STP	UFLP	$\lvert R\rvert$	$\lvert F\rvert$	$\lvert V \setminus R\rvert$	$\lvert E_S\rvert$	$\lvert E_R\rvert$
c5	mp{1,2}	200	200	500	625	40000
c5	mq{1,2}	300	300	500	625	90000
c10	mp{1,2}	200	200	500	1000	40000
c10	mq{1,2}	300	300	500	1000	90000
c15	mp{1,2}	200	200	500	2500	40000
c15	mq{1,2}	300	300	500	2500	90000
c20	mp{1,2}	200	200	500	12500	40000
c20	mq{1,2}	300	300	500	12500	90000
d5	mp{1,2}	200	200	1000	1250	40000
d5	mq{1,2}	300	300	1000	1250	90000
d10	mp{1,2}	200	200	1000	2000	40000
d10	mq{1,2}	300	300	1000	2000	90000
d15	mp{1,2}	200	200	1000	5000	40000
d15	mq{1,2}	300	300	1000	5000	90000
d20	mp{1,2}	200	200	1000	25000	40000
d20	mq{1,2}	300	300	1000	25000	90000

Table 1: Basic properties of benchmark instances.

5.3. Comparison of Formulations

In the first step of our computational study we compare the performance of four proposed formulations, the compact formulation HOP and three cut-set based formulations CUT_x^F, CUT_x^R and CUT_{sa}. More detailed computational results are provided in the Appendix (Tables 4 - 7).

5.3.1. Overall Performance

In Table 2 we show the number of instances that were solved to optimality by each of the tested approaches. We did not impose a time limit. For the instances not solved to optimality the memory requirements of the LP exceeded the 3.25 GB of memory available. In the leftmost column we show the value of H, in the second column the group of instances is specified. We combine every instance of this group with every instance in the set `m{p,q}{1,2}`, thus each line corresponds to 16 instances.

H		CUT_x^F	CUT_x^R	CUT_{sa}	HOP
3	c{5,10,15,20}	16	16	16	16
	d{5,10,15,20}	16	16	16	16
5	c{5,10,15,20}	16	16	16	14
	d{5,10,15,20}	16	16	16	12
7	c{5,10,15,20}	16	16	16	12
	d{5,10,15,20}	16	16	16	10
10	c{5,10,15,20}	16	16	16	10
	d{5,10,15,20}	12	12	12	10

Table 2: Number of instances solved to optimality per group of 16.

In Figure 7 we compare the relative running times of the tested approaches. We chose the running time of model CUT_x^F as reference and display the speedup or slowdown factors for the other models obtained as $\frac{t_M}{t_{CUT_x^F}}$ where $M \in \{CUT_x^R, HOP, CUT_{sa}\}$. Observe that in Figure 7 some entries are missing, in particular when considering model *HOP* for $H \geq 5$ and the other two models for $H = 10$. This is because even the LPs of the corresponding formulations could not be solved due to the memory limitations.

Table 3 compares the four models with respect to the following key figures: Number of enumerated Branch-and-bound nodes (*BB*), number of cuts added (*Cuts*) and running time (t [*s*]). The number of instances that could be solved by all four approaches is given in column $Inst^{OPT}$. The numbers shown are averages over the set of these instances, calculated separately for each value of H. In addition to the calculated arithmetic means (μ_a), in order to avoid dominance of either the harder or the easier instances in the results, we also provide shifted geometric means μ_s (cf. Achterberg [1]) with $s = 1$. For non-negative values $v_i, i \in \{1, \dots, k\}$ the shifted geometric mean for $s \geq 1$ is defined as

$$\mu_s(v_1, \dots, v_k) = \prod_{i=1}^{k} (v_i + s)^{1/k} - s.$$

Instances for which at least one of the models ran out of memory were not considered in the calculation. In the last rows (#*Best*) we count the number of instances (out of 32) for which each approach performed best with respect to the corresponding key figure. Note that two or more approaches can perform equally well on one instance, thus the values for #*Best* do not necessarily add up to 32. In each row we mark the entry of the best approach in bold.

Table 3 clearly indicates that regarding the overall running time, CUT_x^F dominates the other approaches. While for $H = 10$ model *HOP* appears to be faster on the instance set $Inst^{OPT}$, CUT_x^F (*HOP*) solves 8 (0) of the remaining 12 instances to optimality. Table 7 in the Appendix shows that for $H = 10$, four instances remain unsolved by our approaches. Regarding the number of separated cuts, CUT_x^F also dominates CUT_x^R and CUT_{sa}. The number of branch-and-bound nodes varies between the models and does not show any pattern. We recall that we used the default branching strategy of CPLEX enhanced by higher priorities associated to facilities.

			CUT_x^F			CUT_x^R			CUT_{sa}			*HOP*	
H	$Inst^{OPT}$		*BB*	*Cuts*	t [*s*]	*BB*	*Cuts*	t [*s*]	*BB*	*Cuts*	t [*s*]	*BB*	*t* [s]
3	32	μ_a	18.0	**6**	**36.0**	16.0	138	114.6	17.0	111	99.9	**15.0**	54.5
3	32	μ_s, $s=1$	8.3	**2**	**13.4**	**7.5**	37	33.8	8.0	32	31.5	7.7	33.3
5	26	μ_a	19.0	**55**	**57.5**	19.0	371	291.1	**18.0**	317	241.0	19.0	90.0
5	26	μ_s, $s=1$	**14.9**	**18**	**30.0**	15.4	243	112.4	15.3	215	102.1	15.0	61.0
7	22	μ_a	30.0	**216**	164.7	24.0	722	992.9	**21.0**	584	726.8	26.0	**121.2**
7	22	μ_s, $s=1$	24.4	**115**	**82.5**	21.2	642	450.9	**17.6**	517	354.9	22.8	92.8
10	20	μ_a	37.0	**509**	580.9	**26.0**	1220	4128.4	28.0	1064	3730.8	55.0	**176.4**
10	20	μ_s, $s=1$	31.3	**469**	367.2	**21.7**	1081	2321.4	23.7	970	2228.8	39.5	**148.5**
3	32	#*Best*	14	**32**	**28**	19	7	2	**21**	7	2	19	0
5	26	#*Best*	**16**	**32**	**30**	12	1	0	11	1	0	10	2
7	22	#*Best*	5	**26**	**21**	**12**	4	1	11	2	0	7	10
10	20	#*Best*	6	**27**	12	**12**	0	0	11	1	0	1	**16**

Table 3: Comparison chart for models CUT_x^F, CUT_x^R, CUT_{sa} and *HOP*.

The maximum LP gaps over all models, instances and values for the hop limit were less than 5%. For all but 8 instances model CUT_x^R gave a strictly better LP bound than CUT_x^F. We observe

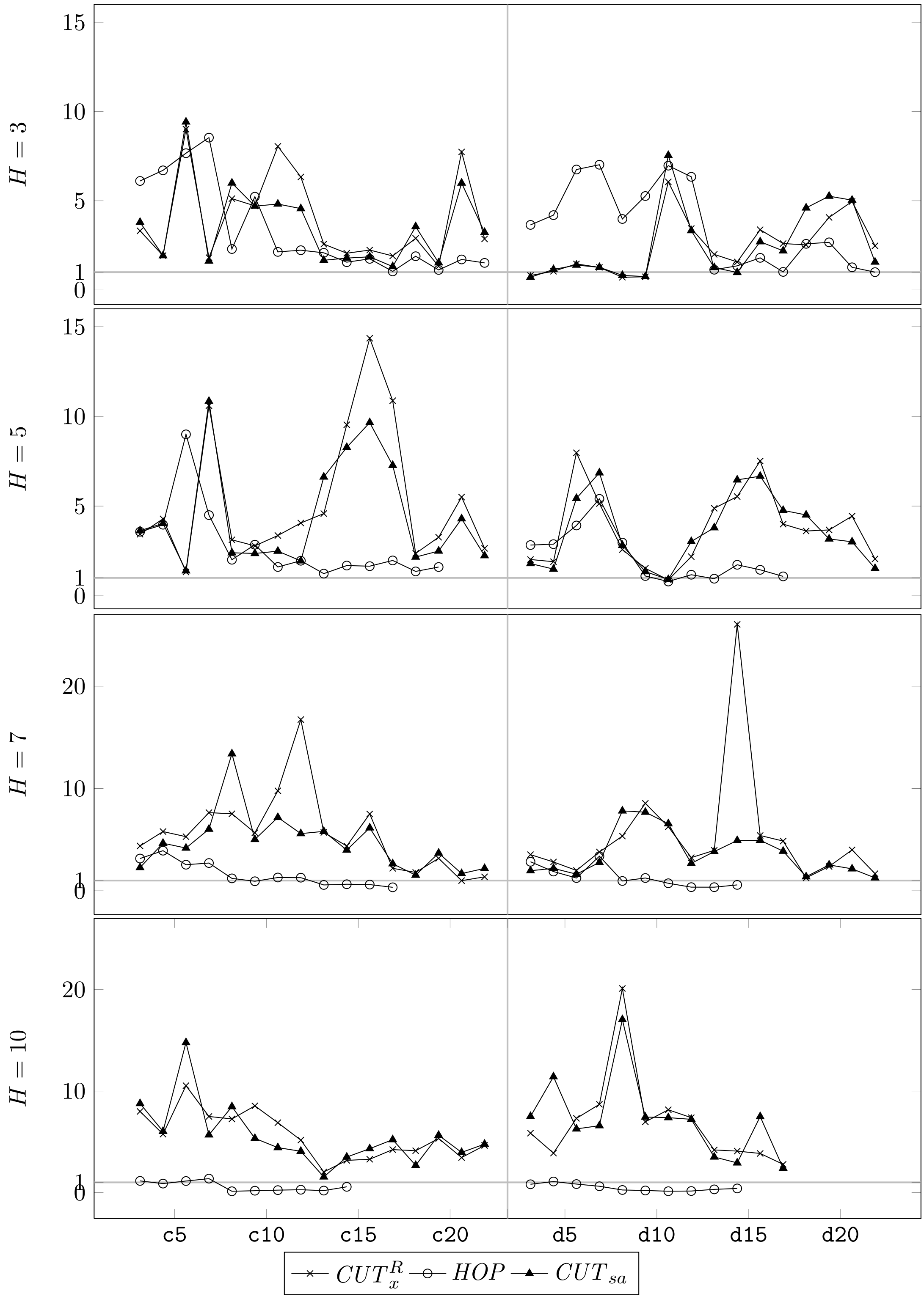


Figure 7: Speedup/slowdown factors of total runtime of models CUT_x^R, HOP and CUT_{sa} compared to CUT_x^F

that the obtained gaps are far below the gaps of the worst-case examples provided in Section 4.

5.3.2. Separation algorithms

In Figure 8 we compare the time spent for separation compared to the total running time of models CUT_x^F and CUT_x^R. The lower value shown indicates the time spent for separation and the upper value indicates the total running time.

Figure 8 shows that typically the amount of time needed for the separation of facility based connectivity cuts is by 1 to 2 orders of magnitude smaller than the corresponding time needed to separate customer based cuts. This can be explained by two factors. One is the size of the core graph (S_x, A_{S_x}) versus the size of the complete layered graph (V_x, A_x). The other factor is the number of maximum flow calculations, that are carried out in each iteration. While this number in case of model CUT_x^F corresponds to the number of non-zero variables $z_i, i \in F$, it is always equal to the number of customers in case of CUT_x^R or CUT_{sa}. The difference between these two values is up to 2 orders of magnitude as indicated by the values of $|F_0|$ given in Tables 4 - 7 in the Appendix. F_0 is the set of non-zero facility variables after solving the LP relaxation at the root node.

The instances in Figure 8 were chosen randomly. The models show a behaviour similar to the one described above on the remaining instances as well.

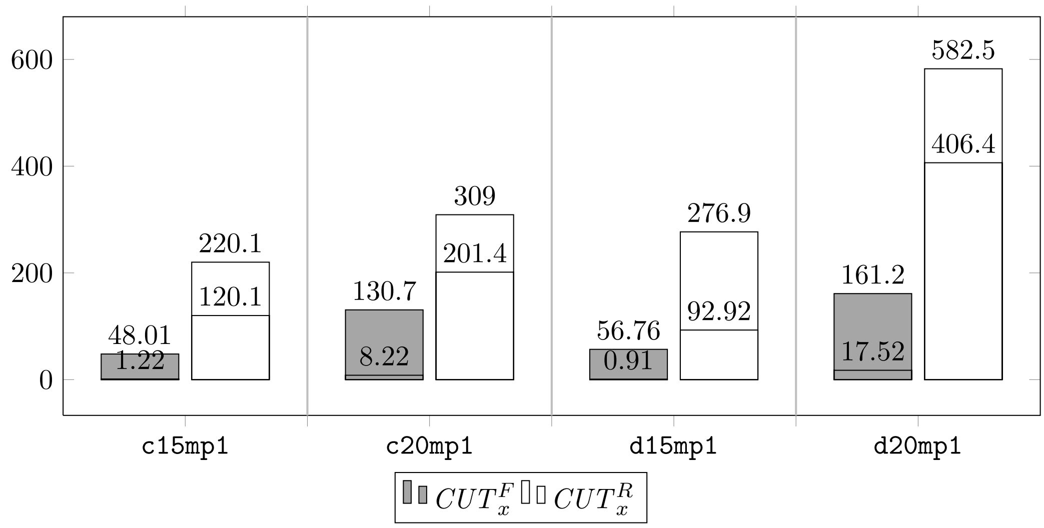


Figure 8: Comparison of the time spent for separation and the total running time for selected instances and $H = 5$.

5.4. Size of the Layered Graph

One of the potential drawbacks of layered graph models might be the size of the underlying graph LG_x. We now study the growth of the size of the layered graph in relation with the number of allowed hops H and in relation with the density of the core graph. Figure 9 shows the relative size of the layered graph, dependent on the value of H, for 4 different instances. We chose one UFLP instance (`mp1`) and combine it with four STP instances of different densities: `c5, c10, c15, c20`. For each of the four instances, we report the following two quotients: $|V_x|/|V|$ and $|A_{S_x}|/|A_S|$, for $H = 3, \ldots, 10$. Note that all reported values for $|V_x|$ and $|A_{S_x}|$ are obtained after the preprocessing described in Section 3.1.1.

One observes that for sparse graphs (c5, c10) and smaller values of H, the graph LG_x is significantly smaller than G. This explains the efficacy of models on LG_x in these cases. The reason why the layered graph is sometimes much smaller than G is the sparsity of the core graph. Many facilities and Steiner nodes might be removed during the preprocessing steps because for small values of H they are not reachable within the given hop limit.

Solving HC ConFL for $H \in \{3, 5\}$ is in most cases even faster than solving the ConFL problem without any hop constraints (cf. the running times for ConFL given in Gollowitzer and Ljubić [11]). As the density of the graph or the value of H increases, the layered graph may become ten times as large as the original graph G (for example, for c20mp1 and $H = 10$). This suggests that layered graph models are better suited for sparse core graphs and smaller values of H. We recall that the density of the assignment graph does not influence the size of the layered graph LG_x.

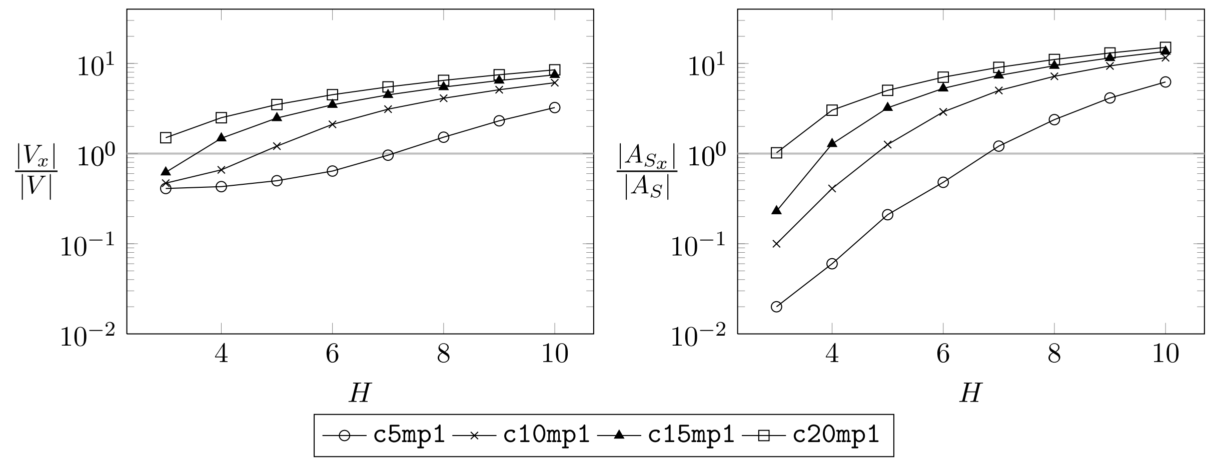


Figure 9: Size of V_x and A_{S_x} compared to size of V and A, respectively.

6. Diameter and Delay Constrained ConFL

Throughout this paper we make the assumption that one node of the solution is known in advance. However, this assumption is not valid for all applications. For example, in the information distribution networks considered by Krick et al. [19] no root node is given. Thus, to ensure reliability, the hop distance between each pair of installed facilities is limited, leading to a diameter constraint (DiaC). Another critical aspect of modern telecommunication networks is signal delay (e.g., for video conferences) or signal attenuation (e.g., in long distance fiber-optic cables). Such applications lead to models with a delay constraint (DelC), i.e., a limit on the delay along the longest path in the network (cf. [17]).

There are recent contributions on layered graph approaches to both of these variants of the Minimum Spanning and Steiner tree problems. Ruthmair and Raidl [29] extend known layered graph models for the Delay constrained Minimum Spanning and Steiner tree problems. They improve this approach by developing an Adaptive Layers Framework that is adjusted continuously during the solution process of a Mixed Integer Program. Gouveia et al. [15] describe how the layered graph for the HCMST can be adapted to model the Diameter constrained MST with either odd or even diameter. This adapted layered graph involves an artificial root node (even diameter) and an additional artificial layer (odd diameter).

To solve the diameter constrained ConFL, we implemented the ideas presented in [15] and ran some computations on the smallest (700 nodes and 40625 edges) instances of the benchmark set described in Section 5. These instances were much larger than those considered by [15] (at most 161 nodes and 12880 edges) and turned out to increase the problem complexity significantly. Model CUT_x^F with the diameter constraint set to 6 took more than 45 minutes of running time and 100 Branch-and-Bound nodes (compared to less than 3 seconds and less than 12 nodes for $H = 3$) for instances `c5mp1`, `c5mp2` and `c5mq1` respectively. Not even within 1 hour instance `c5mq2` was solved to optimality. Considering the difference in size of the used benchmark instances and the fact that HC ConFL generalizes the HCMST these results are not too surprising.

There are two explanations for such a performance:

- The layered graph obtained by the transformation suggested in [15] is much larger than the graph LG_x. In particular, due to insertion of additional level(s), the preprocessing of nodes with in-degree equal to zero has no effect at all.
- Cut set models on layered graphs for DiaC and DelC problems contain a lot of symmetries. The presented approach is not developed to cope with these.

Therefore, we conclude that, in addition to the MIP models considered throughout this paper, efficient approaches for the DiaC or DelC ConFL on larger instances require further investigation on symmetry breaking techniques, preprocessing or adaptive layered graph frameworks.

7. Conclusions

Presently the strongest MIP models for the HCMST are obtained on layered graphs (see Gouveia, Simonetti, and Uchoa [15]). Following this concept, we described two possibilities to develop the strongest MIP models for the HC ConFL so far by modeling it as the directed ConFL problem on layered graphs. In the first transformation, a disaggregation of both the core and the assignment graphs leads towards the corresponding strong MIP models. In the second transformation, we disaggregate only the core graph, and then show that the best MIP formulation on that graph provides the same strong lower bounds to the optimal integer solution value, while saving a significant number of variables. We finally propose a simpler way of modeling HC ConFL as the Steiner arborescence problem on the latter layered graph.

In the computational study, we show that the proposed layered graph models are computationally tractable. The model based on connectivity cuts between the root and open facilities computationally outperforms its stronger counterpart with connectivity cuts between the root and each customer. Surprisingly, the compact three-index model performs comparatively well but shows certain limitations due to its memory requirements. The size of the layered graph may drastically increase with the density of the core graph and with the number of allowed hops.

Acknowledgements

We would like to thank the Area and Associate Editors as well as the two referees for their useful comments and suggestions that significantly improved the quality of this paper. The authors further thank Luis Gouveia for fruitful discussions on topics related to HCMST and HCSTP. Ivana Ljubić is supported by the APART Fellowship of the Austrian Academy of Sciences (OEAW).

Appendix

Proof of Lemma 1. The result can be obtained by applying an error-preserving polynomial reduction from SET COVER. Any SET COVER instance can be reduced into a HC ConFL instance in polynomial time, as follows. We first reduce the SET COVER instance into a hop constrained Steiner tree instance in which all edge weights are set to 1 (see Manyem and Stallmann [26] or Manyem [25]). We then reduce such obtained hop constrained Steiner tree instance into a HC ConFL instance as follows: For each terminal i in the hop constrained Steiner tree we define a potential facility in HC ConFL. Then, for each such facility i, we add a customer node c_i. Each customer c_i is connected only to facility i with an edge of weight 1. The result follows immediately from the fact that SET COVER cannot be approximated in polynomial time within any factor smaller than $c \ln n$ (c is a constant given by Alon et al. [2] and n is the number of items to be covered) unless $P = NP$. □

For the proof of Lemma 2 we need the following

Lemma 10. *There always exists an optimal solution $(V^0_{x,z}, A^0_{x,z})$ of directed ConFL on the layered graph $LG_{x,z}$ such that*

$$\sum_{p=1}^{H} |\delta^-\{(i,p)\} \cap A^0_{x,z}| \leq 1 \quad \forall i \in F \setminus \{r\} \tag{21}$$

and

$$\sum_{p=1}^{H-1} |\delta^-\{(i,p)\} \cap A^0_{x,z}| \leq 1 \quad \forall i \in S \setminus F. \tag{22}$$

Proof. Assume that, w.l.o.g., there exists a node $j \in S$, whose in-degree over all levels is equal to 2, i.e., there exist p and q ($1 \leq p < q \leq H$) such that in-degree of (j,p) and (j,q) is equal to one. Denote by T^q_j the optimal sub-tree rooted at (j,q). We transform the solution as follows: a) We move the core arcs in T^q_j up by $q-p$ levels, such that the obtained tree is then rooted in (j,p). We then refer to it as T^p_j. b) For customers assigned to open facilities (i,l), $q \leq l \leq H$ in T^q_j, we assign them to facility $(i, l-q+p)$ instead. c) Finally, starting from (j,q) towards r, we recursively remove nodes with out-degree 0 from the solution.

By repeating this procedure for all nodes whose respective in-degree is greater than 1, we obtain a solution with the desired property. As we remove arcs with non-negative cost and reassign customers without incurring additional cost, the obtained solution is at most as expensive as the original one. □

Proof of Lemma 2. We associate binary variables to the arcs in $A_{x,z}$ as follows: X^1_{rj} corresponds to $(r,(j,1)) \in A_1$, X^p_{ij} to $((i,p-1),(j,p)) \in A_2$, X^H_{ij} to $((i,H-1),(j,H)) \in A_3$, X^1_{rk} to $rk \in A_4$ and X^p_{ik} corresponds to $((i,p),k) \in A_5$.

We prove the lemma as follows: We first show that every feasible solution $(\mathbf{x}, \mathbf{z})$ of HC ConFL on G can be mapped onto a feasible solution $(\mathbf{X}, \mathbf{Z})$ of the directed ConFL problem on the according layered graph $LG_{x,z}$. Then we show that an optimal solution of the directed ConFL problem on $LG_{x,z}$ with the property of Lemma 10 can be mapped onto a feasible solution of HC ConFL on G. We prove optimality of this solution by contradiction.

Consider a solution $(\mathbf{x}, \mathbf{z})$. We label the nodes in S in this solution according to their respective hop distance from the root. For core arcs $ij \in A_S$ such that $x_{ij} = 1$ and such that i and j are labelled $p-1$ and p respectively, we set $X_{ij}^p = 1$. For nodes i such that $z_i = 1$ we set $Z_i^p = 1$ for p equal to the label of node i. For assignment arcs $jk \in A_R$ such that $x_{jk} = 1$ and node j has label p we set $X_{jk}^p = 1$. This mapping preserves a feasible assignment of customers to open facilities as well as connectivity among those chosen facilities. Thus, the solution corresponding to $(\mathbf{X}, \mathbf{Z})$ is feasible for the directed ConFL problem on $LG_{x,z}$. By the cost structure of $LG_{x,z}$ it also incurs the same cost as $(\mathbf{x}, \mathbf{z})$.

Consider now a cost-optimal solution $(\mathbf{X}, \mathbf{Z})$ on $LG_{x,z}$ with the property of Lemma 10. Ignoring the second index on the nodes of that solution, we obtain a feasible HC ConFL solution $(\mathbf{x}, \mathbf{z})$ in G (i.e., a ConFL solution with at most H hops). By the cost structure of $LG_{x,z}$ $(\mathbf{x}, \mathbf{z})$ has the same objective function value as $(\mathbf{X}, \mathbf{Z})$. Assume now, that $(\mathbf{x}, \mathbf{z})$ is not optimal on G, i.e., there exists a solution $(\mathbf{x}', \mathbf{z}')$ with a strictly lower cost. We can project this solution onto a solution $(\mathbf{X}', \mathbf{Z}')$ on $LG_{x,z}$ as described above. $(\mathbf{X}', \mathbf{Z}')$ then has a lower cost than $(\mathbf{X}, \mathbf{Z})$ which is a contradiction to $(\mathbf{X}, \mathbf{Z})$ being optimal.

Figures 2d) and 1b) illustrate this mapping for one instance. □

Proof of Lemma 4. We will associate binary variables to the arcs in A_x as follows: X_{rj}^1 corresponds to $(r, (j, 1)) \in A_1$, X_{ij}^p to $((i, p-1), (j, p)) \in A_2$, X_{ij}^H to $((i, H-1), (j, H)) \in A_3$, X_{ii}^p to $((i, p-1), (i, H)) \in A_6$.

The proof follows the same idea as the proof of Lemma 2.

A mapping of a solution $(\mathbf{x}, \mathbf{z})$ in G onto a solution $(\bar{\mathbf{X}}, \bar{\mathbf{x}}, \bar{\mathbf{z}})$ in LG_x is the following: We label the nodes in S in this solution according to their respective hop distance from the root. For core arcs $ij \in A_S$ such that $x_{ij} = 1$ and such that i and j are labelled $p-1$ and p respectively, we set $\bar{X}_{ij}^p = 1$. For nodes i such that $z_i = 1$ we set $\bar{z}_i = 1$. If the label of such a node i is $p < H$ we set $\bar{X}_{ii}^{p+1} = 1$ in addition. For assignment arcs $jk \in A_R$ such that $x_{jk} = 1$ we set $\bar{x}_{jk} = 1$. This mapping preserves the assignment of customers to open facilities and provides connectivity among those chosen facilities, possibly using additional arcs in A_6. Thus, the solution corresponding to $(\bar{\mathbf{X}}, \bar{\mathbf{x}}, \bar{\mathbf{z}})$ is feasible for the directed ConFL problem on $LG_{x,z}$. By the cost structure of LG_x (arcs in A_6 have a cost of 0) it also incurs the same cost as $(\mathbf{x}, \mathbf{z})$.

Consider now a cost-optimal solution $(\bar{\mathbf{X}}, \bar{\mathbf{x}}, \bar{\mathbf{z}})$ on LG_x with the property of Lemma 10 but where arcs in A_6 are ignored in the summation terms. Removing the arcs in A_6 and ignoring the second index on the nodes of that solution, we obtain a feasible HC ConFL solution $(\mathbf{x}, \mathbf{z})$ in G (i.e., a ConFL solution with at most H hops). By the cost structure of LG_x $(\mathbf{x}, \mathbf{z})$ has the same objective function value as $(\bar{\mathbf{X}}, \bar{\mathbf{x}}, \bar{\mathbf{z}})$. Assume now, that $(\mathbf{x}, \mathbf{z})$ is not optimal on G, i.e., there exists a solution $(\mathbf{x}', \mathbf{z}')$ with a strictly lower cost. We can project this solution onto a solution $(\tilde{\mathbf{X}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ on $LG_{x,z}$ as described above. $(\tilde{\mathbf{X}}, \tilde{\mathbf{x}}, \tilde{\mathbf{z}})$ then has a lower cost than $(\bar{\mathbf{X}}, \bar{\mathbf{x}}, \bar{\mathbf{z}})$ which is a contradiction to $(\bar{\mathbf{X}}, \bar{\mathbf{x}}, \bar{\mathbf{z}})$ being optimal. □

Proof of Lemma 6. Constraints (8) dominate constraints (1). Thus, formulation $CUT_{x,z}^{F+}$ is at least as strong as $CUT_{x,z}^{F}$. The strict relation holds because of the example in Figure 10. To show an instance for which $\frac{v_{LP}(CUT_{x,z}^{F+})}{v_{LP}(CUT_{x,z}^{F})} \approx H-1$ holds, we generalize the above example. The subgraph induced by nodes $\{1, 2, 3\}$ is replaced by the subgraph containing nodes $\{1, \ldots, H-1\}$ being the Steiner nodes and a node H, being the facility node. This subgraph is connected as follows: Node H is connected to all $i = 1, \ldots, H-1$ with an edge of cost $c_{iH} = H-i$. For each $i = 1, \ldots, H-1$, node

i is connected to $i+1$ with an edge of cost $c_{i,i+1} = 1$. In the LP-relaxation of the model $CUT^F_{x,z}$, all facilities (H,p) at levels $p = 2, \dots, H$ will be open with $Z^p_H = 1/(H-1)$, and consequently, $X^1_{r1} = 1/(H-1)$, so that $\upsilon_{LP}(CUT^F_{x,z}) \approx L/(H-1)$. In contrast, the optimal LP-value of the model $CUT^{F+}_{x,z}$ is $\upsilon_{LP}(CUT^{F+}_{x,z}) \approx L$, which proves the claim. $\square$

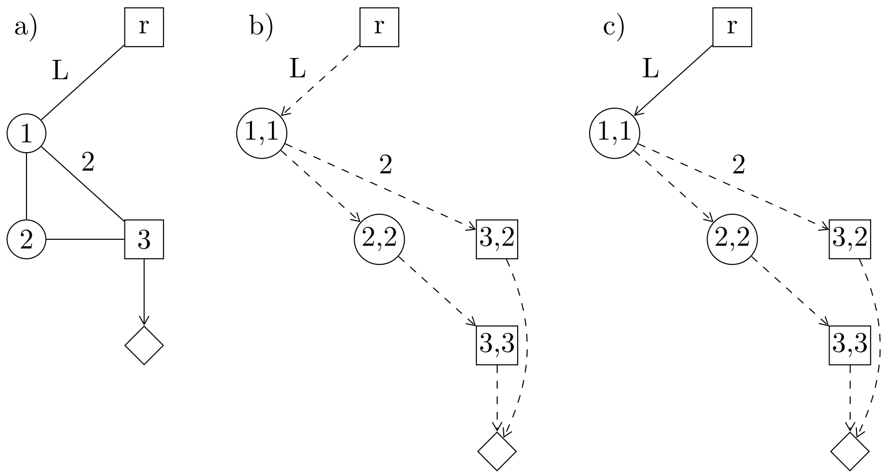


Figure 10: a) Instance on G with $H = 3$; b) LP optimal solution for $CUT^F_{x,z}$. Dashed and solid arcs take LP-values equal to 1/2 and 1, respectively. $\upsilon_{LP}(CUT^F_{x,z}) = L/2 + 4$; c) LP optimal solution for $CUT^{F+}_{x,z}$ with cost $L + 4$.

Proof of Lemma 7. We first show that $\upsilon_{LP}(CUT^F_x) \geq \upsilon_{LP}(HOP)$ and then give an example for which the strict inequality holds:

$\upsilon_{LP}(CUT^F_x) \geq \upsilon_{LP}(HOP)$: It is enough to show that an optimal LP-solution of the formulation CUT^F_x is also feasible for the model *HOP*. For that purpose we will use the max-flow min-cut theorem. A flow formulation on the graph G which is equivalent to the CUT^F_x formulation is given below. It comprises additional flow variables f^{kp}_{ij}, for all $ij \in A_S$, and $k \in F \setminus \{r\}$, $p = 1, \dots, H$, representing the flow of commodity k on arc ij at the p-th position from the root node. We denote this formulation by MCF_F:

$$\sum_{ji \in A_S} f^{k,p-1}_{ji} - \sum_{ij \in A_S} f^{kp}_{ij} = 0 \qquad \forall k \in F \setminus \{r\},\ i \in S \setminus \{r,k\},\ p = 2, \dots, H \tag{23}$$

$$\sum_{rj \in A_S} f^{k1}_{rj} = z_k \qquad \forall k \in F \setminus \{r\} \tag{24}$$

$$\sum_{p=1}^{H} \sum_{jk \in A_S} f^{kp}_{jk} = z_k \qquad \forall k \in F \setminus \{r\} \tag{25}$$

$$0 \leq f^{kp}_{ij} \leq X^p_{ij} \qquad \forall ij \in A_S,\ k \in F \setminus \{r\},\ p = 1, \dots, H \tag{26}$$

$$(11) - (17)$$

Let $(\mathbf{X}', \mathbf{x}', \mathbf{z}', \mathbf{f}')$ be an optimal LP-solution for MCF_F and $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$ its projection into the space of $(\mathbf{X}, \mathbf{x}, \mathbf{z})$ variables. We will show that $(\mathbf{X}', \mathbf{x}', \mathbf{z}') \in \mathcal{P}_{HOP}$. Constraints (20) are directly implied by inequalities (23)-(26). To show that constraints (19) are also satisfied, we first observe that, for every X^p_{jl}, $jl \in A_S$, $p = 1, \dots, H$, there exists a commodity $k \in F \setminus \{r\}$ such that constraint (26) is tight, i.e., $X^p_{jl} = f^{kp}_{jl}$. From the flow conservation constraints (23)-(25), it follows:

$$X'^p_{jl} = f'^{kp}_{jl} \leq \sum_{\substack{i \in S\setminus\{k\}:\\ ij \in A_S}} f'^{k,p-1}_{ij} \leq \sum_{\substack{i \in S\setminus\{k\}:\\ ij \in A_S}} X'^{p-1}_{ij}$$

and thus, inequalities (19) hold for $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$.

$v_{LP}(CUT^F_x) > v_{LP}(HOP)$**:** Consider an example given in Figure 11. LP-solution for HOP shown in Figure 11b) is not feasible for LG_xCUT_F and the strict inequality regarding the LP-values holds.

□

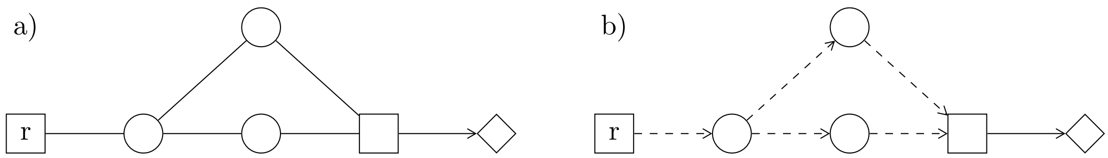


Figure 11: a) Instance G with $H = 3$. b) An optimal LP-solution for HOP_F in which dashed arcs take value 1/2.

Proof of Lemma 8. The result given in Gollowitzer and Ljubić [11] shows that the relative gap between the LP-values of models CUT_F and CUT_R can be as large as $|F| - 1$, where $|F|$ is the number of facilities of a ConFL instance. Since the number of facilities in LG_x is $|F|$ and the number of facilities in $LG_{x,z}$ is $(|F| - 1)|H| + 1$, the result follows immediately. □

Proof of Lemma 9. To prove this claim, we describe mappings between corresponding LP-solutions as follows.

$v_{LP}(CUT^R_{x,z}) \geq v_{LP}(CUT^R_x)$**:** Let $(\mathbf{X}, \mathbf{Z})$ be an optimal LP-solution of the model $CUT^R_{x,z}$. We project $(\mathbf{X}, \mathbf{Z})$ into a solution $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$ and show that it is feasible for the model CUT^R_x. We set $X'^p_{ij} := X^p_{ij}$ for all arcs in A_1, A_2 and A_3; $X'^p_{jj} := Z^p_j (= \max_{k \in R} X^p_{jk})$ for all arcs in A_6; $x'_{jk} := \sum_{p=1}^H X^p_{jk}$ for all arcs in A_7; $x'_{rk} := X^1_{rk}$ for all arcs in A_4; $z_i := \sum_{p=1}^H Z^p_i$. All the remaining $\mathbf{X}'$ values are set to zero. Obviously, constraints (12)-(13) are satisfied, it only remains to show that $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$ satisfies (18). Denote by $\delta^-(W)_{|D} = \{ij \in \delta^-(W) \mid ij \in D\}$. Then, $X_x[\delta^-(W)] = X_x[\delta^-(W)_{|\cup_{i=1}^4 A_i}] + X_x[\delta^-(W)_{|A_6 \cup A_7}] = X[\delta^-(W)_{|\cup_{i=1}^4 A_i}] + X_x[\delta^-(W)_{|A_6 \cup A_7}] \geq X[\delta^-(W)_{|\cup_{i=1}^4 A_i}] + X[\delta^-(W)_{|A_5}] = X[\delta^-(W)] \geq 1$.

$v_{LP}(CUT^R_x) \geq v_{LP}(CUT^R_{x,z})$**:** Let $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$ be an optimal LP-solution of the model CUT^R_x. We project this vector into $(\mathbf{X}, \mathbf{Z})$ as follows: $X^p_{ij} := X'^p_{ij}$ for all arcs in A_1, A_2 and A_3; $X^1_{rk} := x'_{rk}$ for all arcs in A_4. Furthermore, we set $Z^p_j := X'^p_{jj}$, for all arcs from A_6, for $p = 1, \dots, H - 1$, and $Z^H_j := z'_j - \sum_{p=1}^{H-1} Z^p_j$, for all $j \in F \setminus \{r\}$. We then recursively define $X^p_{jk} := \min(Z^p_j, x'_{jk} - \sum_{q=p+1}^H X^q_{jk})$ starting from $p = H, \dots, 1$. By definition, $(\mathbf{X}, \mathbf{Z})$ satisfies constraints (2)-(6). To show that constraints (9) are satisfied as well, observe that arc capacities defined as $(\mathbf{X}', \mathbf{x}')$

enable for each commodity $k \in R$ one unit of flow to be sent from r to k in LG_x. By using the above mapping of arcs and their capacities from LG_x to $LG_{x,z}$, we also ensure that one unit of flow can be sent from the root to each commodity $k \in R$ in the graph $LG_{x,z}$ which concludes the proof.

□

In the following 4 tables we show detailed computational results. For each of the approaches CUT_x^F, CUT_x^R and CUT_{sa} and HOP and for each instance we show: the number of nodes in the branch and bound tree (BB) and the total running time ($t[s]$). In addition, for the first three approaches, we also provide: the number of user cuts added (Cuts), and the time needed to separate them (t_{sep}).

Bibliography

[1] Achterberg, Tobias. 2009. SCIP: solving constraint integer programs. *Math. Programming Comput.* **1** 1–41.

[2] Alon, N., D. Moshkovitz, S. Safra. 2006. Algorithmic construction of sets for k-restrictions. *ACM Trans. Algorithms* **2**(2) 153–177.

[3] Balakrishnan, A., K. Altinkemer. 1992. Using a hop-constrained model to generate alternative communication network design. *INFORMS J. Comput.* **4**(2) 192–205.

[4] Bardossy, M. G., S. Raghavan. 2010. Dual-based local search for the connected facility location and related problems. *INFORMS J. Comput.* **22**(4) 584–602.

[5] Beasley, J. E. 1990. OR-Library: distributing test problems by electronic mail. *J. Oper. Res. Soc.* **41**(11) 1069–1072. URL `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/steininfo.html`.

[6] Cherkassky, B. V., A. V. Goldberg. 1994. On implementing push-relabel method for the maximum flow problem. *Algorithmica* **19** 390–410.

[7] Costa, A. M., J.-F. Cordeau, G. Laporte. 2008. Fast heuristics for the Steiner tree problem with revenues, budget and hop constraints. *Eur. J. Oper. Res.* **190**(1) 68–78.

[8] Costa, A. M., J.-F. Cordeau, G. Laporte. 2009. Models and branch-and-cut algorithms for the Steiner tree problem with revenues, budget and hop constraints. *Networks* **53**(2) 141–159.

[9] Dahl, G., L. Gouveia, C. Requejo. 2006. On formulations and methods for the hop-constrained minimum spanning tree problem. P. M. Pardalos, M. Resende, eds., *Handbook of Optimization in Telecommunications*. Springer, 493–515.

[10] Eisenbrand, F., F. Grandoni, T. Rothvoí, G Schäfer. 2010. Connected facility location via random facility sampling and core detouring. *J. Comput. System Sci.* **76** 709–726.

[11] Gollowitzer, S., I. Ljubić. 2011. MIP models for connected facility location: A theoretical and computational study. *Comput. Oper. Res.* **38**(2) 435–449.

			CUT_x^F				CUT_x^R				CUT_{sa}				HOP	
Inst.	OPT	$\lvert F_0 \rvert$	BB	$Cuts$	t_{sep}	t $[s]$	BB	$Cuts$	t_{sep}	t $[s]$	BB	$Cuts$	t_{sep}	t $[s]$	BB	t [s]
`c5mp1`	2907.96	6	**5**	**0**	0.0	**1.4**	**5**	7	2.8	4.8	**5**	10	3.5	5.5	**5**	8.8
`c5mp2`	2912.63	4	**3**	**0**	0.0	**1.3**	**3**	**0**	1.2	2.4	**3**	**0**	1.2	2.4	**3**	8.5
`c5mq1`	4505.04	7	11	**0**	0.0	**2.5**	11	18	20.3	22.9	**9**	23	21.1	24.0	11	19.5
`c5mq2`	4082.42	0	**0**	**0**	0.0	**2.3**	**0**	**0**	1.1	4.1	**0**	**0**	1.1	3.7	**0**	19.5
`c10mp1`	2861.05	12	39	**3**	0.1	**6.8**	38	261	25.3	34.7	62	272	30.3	40.7	**34**	15.6
`c10mp2`	2760.27	10	**3**	**0**	0.0	**2.1**	**3**	108	7.5	10.0	**3**	92	7.0	10.0	**3**	11.1
`c10mq1`	4092.95	12	17	**0**	0.0	**13.1**	23	190	91.6	105.7	16	133	52.5	63.3	**13**	28.1
`c10mq2`	3946.52	13	29	**3**	0.1	**15.3**	29	169	78.8	96.8	**21**	108	57.3	69.7	29	34.1
`c15mp1`	2668.48	20	9	**2**	0.1	**13.5**	9	224	24.7	34.6	**7**	118	12.4	22.4	9	28.1
`c15mp2`	2679.63	14	**9**	**0**	0.0	**20.9**	13	271	28.3	43.1	**9**	177	18.9	37.4	**9**	32.6
`c15mq1`	3861.57	7	25	**1**	1.4	**82.1**	**17**	254	132.1	183.7	21	241	119.0	152.6	21	143.3
`c15mq2`	3694.56	16	63	**6**	1.2	**130.9**	27	315	183.8	250.0	27	214	126.2	172.1	**23**	136.6
`c20mp1`	2618.66	17	11	**14**	0.3	**24.7**	**9**	130	50.5	71.6	11	131	56.5	87.8	13	46.8
`c20mp2`	2630.46	14	7	**36**	0.4	**33.2**	**5**	69	28.4	44.3	9	86	34.5	50.7	9	37.4
`c20mq1`	3828.50	15	45	**36**	1.0	**80.5**	53	305	452.4	622.3	**39**	204	302.0	482.2	43	137.8
`c20mq2`	3687.49	20	37	**36**	1.1	**149.3**	**27**	190	293.4	427.4	35	181	296.3	482.7	**27**	226.9
`d5mp1`	2846.01	0	**0**	**0**	0.0	2.5	**0**	**0**	0.2	2.1	**0**	**0**	0.3	**1.8**	**0**	9.2
`d5mp2`	2847.68	5	**3**	**0**	0.0	**2.2**	**3**	1	1.0	2.3	**3**	1	1.2	2.5	5	9.0
`d5mq1`	4190.20	0	**0**	**0**	0.0	**2.9**	**0**	**0**	1.1	4.2	**0**	**0**	1.1	4.0	**0**	19.4
`d5mq2`	3978.17	0	**0**	**0**	0.0	**2.8**	**0**	**0**	1.1	3.5	**0**	**0**	1.1	3.6	**0**	19.6
`d10mp1`	2970.53	0	**0**	**0**	0.0	2.4	**0**	**0**	0.3	**1.7**	**0**	**0**	0.3	2.0	**0**	9.5
`d10mp2`	2941.59	0	**0**	**0**	0.0	1.9	**0**	**0**	0.2	**1.4**	**0**	**0**	0.2	1.4	**0**	9.9
`d10mq1`	4212.81	9	7	**0**	0.0	**3.1**	**3**	43	15.8	18.7	**3**	48	20.3	23.2	**3**	21.4
`d10mq2`	3979.59	5	**3**	**0**	0.0	**3.3**	**3**	2	8.3	11.4	**3**	1	7.9	11.0	**3**	20.9
`d15mp1`	2805.22	21	123	**0**	0.4	**62.4**	76	540	68.3	124.6	**61**	330	37.4	78.4	75	71.0
`d15mp2`	2692.85	10	**11**	**0**	0.1	12.6	**11**	116	14.0	19.8	**11**	56	7.8	**12.4**	**11**	17.1
`d15mq1`	3890.39	12	19	**0**	0.0	**34.5**	15	216	94.8	116.6	**11**	156	67.6	93.4	17	62.2
`d15mq2`	3788.07	20	25	**0**	0.6	**82.2**	**17**	386	177.8	214.1	51	236	132.7	180.5	27	83.5
`d20mp1`	2621.66	17	11	**12**	0.3	**23.9**	**9**	82	34.3	60.4	31	148	74.0	109.9	13	61.8
`d20mp2`	2632.46	13	**5**	**22**	0.3	**18.7**	6	130	54.2	75.9	11	198	81.7	98.2	9	49.9
`d20mq1`	3830.50	14	49	**18**	0.8	**107.0**	51	224	362.9	531.0	55	282	428.4	538.7	**39**	136.2
`d20mq2`	3687.49	19	38	**12**	1.6	**210.1**	49	182	350.3	520.0	**31**	114	204.3	329.2	**31**	210.2

Table 4: Comparison of models CUT_x^F, CUT_x^R, CUT_{sa} and HOP for $H = 3$. The best running times are shown in bold.

			CUT_x^F				CUT_x^R				CUT_{sa}				HOP	
Inst.	OPT	$\lvert F_0\rvert$	BB	$Cuts$	t_{sep}	t $[s]$	BB	$Cuts$	t_{sep}	t $[s]$	BB	$Cuts$	t_{sep}	t $[s]$	BB	t [s]
`c5mp1`	2839.80	15	33	**3**	0.1	**4.8**	27	137	11.6	16.4	**25**	156	12.8	17.4	27	17.1
`c5mp2`	2839.05	14	**15**	**0**	0.1	**3.8**	19	154	13.0	16.4	17	133	11.9	15.4	**15**	15.2
`c5mq1`	3986.08	0	**0**	**0**	0.0	**3.5**	**0**	**0**	1.2	4.7	**0**	**0**	1.1	4.9	**0**	31.5
`c5mq2`	3928.49	12	23	**4**	0.0	**8.4**	29	251	77.3	89.4	17	273	79.6	91.5	**15**	38.0
`c10mp1`	2683.48	18	**11**	**24**	0.2	**28.6**	17	547	49.1	89.3	17	388	37.9	68.6	13	57.5
`c10mp2`	2663.46	12	7	**25**	0.1	**14.0**	**5**	256	18.6	39.1	8	193	16.7	33.1	7	40.0
`c10mq1`	3867.57	15	**27**	**15**	1.2	**114.0**	31	526	272.4	383.2	31	368	178.2	283.8	29	183.2
`c10mq2`	3733.85	20	57	**30**	1.1	**170.4**	37	850	432.2	691.6	**33**	453	220.0	332.0	**33**	333.1
`c15mp1`	2637.66	18	**17**	**166**	1.2	**48.0**	19	486	120.1	220.1	21	611	159.2	318.0	31	59.3
`c15mp2`	2644.46	14	10	**136**	0.8	**22.7**	11	597	151.7	216.7	**9**	467	104.8	188.1	15	38.3
`c15mq1`	3846.50	15	39	**249**	2.3	**107.4**	43	1038	923.5	1542.0	**34**	810	626.5	1037.0	53	177.4
`c15mq2`	3692.56	20	25	**203**	1.8	**111.0**	29	879	761.7	1207.0	**21**	717	566.7	807.4	30	218.8
`c20mp1`	2618.66	17	**11**	**83**	8.2	**130.7**	**11**	126	201.4	309.0	19	96	163.0	282.2	19	177.8
`c20mp2`	2626.46	14	**6**	**37**	3.3	**71.0**	13	85	170.4	232.0	9	75	112.3	177.3	9	114.0
`c20mq1`	3826.50	14	44	**193**	20.7	**324.5**	**42**	307	1098.0	1786.0	71	279	1107.0	1394.0	-	-
`c20mq2`	3686.49	20	**31**	**150**	16.8	**475.5**	54	219	926.4	1258.0	41	198	701.1	1064.0	-	-
`d5mp1`	2766.52	11	**9**	**6**	0.1	**5.5**	**9**	126	7.3	11.2	**9**	121	6.5	10.0	**9**	15.7
`d5mp2`	2795.15	10	11	**6**	0.0	**5.3**	9	82	6.8	10.2	**5**	70	4.4	8.0	9	15.4
`d5mq1`	4124.65	15	**13**	**5**	0.1	**10.9**	15	292	70.4	87.0	17	199	47.2	59.3	15	42.8
`d5mq2`	3826.77	11	9	**4**	0.1	**7.1**	**7**	112	27.0	36.5	9	155	37.8	48.6	11	38.3
`d10mp1`	2759.67	22	13	**8**	0.1	**11.6**	**11**	325	17.9	30.0	13	393	20.9	32.7	**11**	34.5
`d10mp2`	2782.68	18	37	**0**	0.1	**29.3**	23	382	27.6	44.9	29	334	26.6	39.5	**15**	32.4
`d10mq1`	3892.51	14	9	**5**	0.5	102.4	**7**	210	63.8	90.9	21	160	59.5	93.4	**7**	**81.6**
`d10mq2`	3760.49	20	**17**	**12**	2.1	**93.9**	31	446	147.2	204.8	35	514	194.0	284.3	23	110.3
`d15mp1`	2643.66	18	21	**80**	0.9	56.8	**15**	368	92.9	276.9	17	286	68.4	215.4	21	**54.1**
`d15mp2`	2647.46	13	9	**44**	0.5	**23.1**	9	253	53.8	127.6	**7**	251	56.5	149.0	**7**	39.9
`d15mq1`	3850.06	14	53	**210**	2.0	**99.0**	45	558	506.4	744.2	**37**	430	361.2	660.5	51	143.5
`d15mq2`	3702.56	20	**23**	**100**	2.5	**211.9**	43	576	499.9	846.8	31	601	541.5	1009.0	**23**	229.5
`d20mp1`	2619.66	17	**21**	**59**	17.5	**161.2**	**21**	128	406.4	582.5	23	182	485.3	726.7	-	-
`d20mp2`	2628.46	14	**7**	**40**	12.6	**102.3**	**7**	85	273.7	375.6	**7**	64	204.5	324.3	-	-
`d20mq1`	3828.50	14	**46**	**246**	81.1	**828.9**	71	500	2762.0	3681.0	60	383	2077.0	2496.0	-	-
`d20mq2`	3685.49	20	**35**	**36**	18.5	**733.1**	**35**	170	966.5	1507.0	39	126	735.6	1120.0	-	-

Table 5: Comparison of models CUT_x^F, CUT_x^R, CUT_{sa} and HOP for $H = 5$. The best running times are shown in bold.

			CUT_x^F				CUT_x^R				CUT_{sa}				HOP	
Inst.	OPT	$\|F_0\|$	BB	$Cuts$	t_{sep}	t $[s]$	BB	$Cuts$	t_{sep}	t $[s]$	BB	$Cuts$	t_{sep}	t $[s]$	BB	t [s]
c5mp1	2703.97	18	13	**14**	0.1	**9.9**	21	331	25.8	43.2	**9**	210	11.5	22.4	13	31.1
c5mp2	2736.55	17	25	**17**	0.1	**9.1**	21	481	35.7	52.5	**15**	417	26.2	42.2	**15**	35.6
c5mq1	3906.98	14	29	**17**	0.2	**36.8**	21	452	119.7	194.5	**19**	366	101.5	153.6	23	93.7
c5mq2	3842.99	24	49	**42**	0.3	**73.2**	59	1037	345.6	559.2	**41**	963	309.8	441.0	45	198.5
c10mp1	2661.66	19	25	**287**	1.0	**54.6**	17	801	115.1	411.1	15	732	83.1	731.7	**12**	66.2
c10mp2	2663.46	13	9	**187**	0.4	42.8	**7**	476	52.5	242.9	11	430	49.4	214.9	25	**39.8**
c10mq1	3867.57	16	51	**317**	1.6	**132.2**	**31**	1154	616.6	1292.0	45	715	416.5	948.1	**31**	171.5
c10mq2	3733.85	19	47	**499**	2.8	**250.5**	49	1888	1176.0	4192.0	53	1146	686.8	1400.0	**41**	320.8
c15mp1	2634.66	17	**17**	**473**	8.9	162.7	19	744	345.1	927.6	19	822	363.2	944.7	68	**92.8**
c15mp2	2640.46	14	12	**336**	4.7	88.2	11	396	172.5	385.9	**9**	381	168.0	351.0	51	**56.3**
c15mq1	3844.50	15	53	**617**	15.3	358.9	**41**	1134	1434.0	2695.0	47	954	1095.0	2207.0	52	**217.5**
c15mq2	3689.56	20	70	**718**	23.8	747.8	42	762	1013.0	1642.0	33	742	939.0	1978.0	**29**	**253.1**
c20mp1	2618.66	17	**35**	186	146.7	**552.1**	45	217	803.1	990.5	59	**174**	654.9	842.3	-	-
c20mp2	2626.46	14	**12**	**93**	49.2	**206.8**	41	137	536.7	655.9	28	184	622.9	762.3	-	-
c20mq1	3826.50	14	37	230	149.7	1352.0	**27**	**186**	1042.0	**1338.0**	77	255	1490.0	2276.0	-	-
c20mq2	3686.49	20	269	335	195.5	**1677.0**	**53**	**278**	1728.0	2283.0	64	456	2417.0	3665.0	-	-
d5mp1	2685.94	10	10	**25**	0.1	**9.3**	7	346	19.6	32.7	**3**	183	9.0	18.2	11	26.3
d5mp2	2761.15	8	22	**55**	0.3	**17.1**	19	351	27.2	48.0	**13**	272	20.4	37.3	21	32.1
d5mq1	3903.51	11	21	**13**	0.8	**107.0**	33	336	128.0	212.7	19	328	100.3	173.7	**15**	133.3
d5mq2	3744.49	20	17	**33**	0.2	**45.6**	**11**	424	117.2	173.0	15	278	79.7	126.6	13	154.3
d10mp1	2685.54	19	17	**120**	0.5	78.4	**13**	625	65.9	419.1	19	813	93.3	612.2	21	**75.0**
d10mp2	2693.46	14	13	**102**	0.5	**46.8**	**11**	690	79.8	400.3	13	648	81.5	359.9	13	58.4
d10mq1	3873.06	16	33	**157**	0.7	280.6	**27**	861	385.8	1757.0	35	783	374.9	1840.0	33	**206.1**
d10mq2	3724.49	21	90	**266**	4.4	727.4	31	962	434.3	2358.0	**19**	820	313.7	1949.0	23	**257.6**
d15mp1	2639.66	17	41	**276**	7.3	233.3	**15**	527	295.3	922.0	17	491	265.3	893.9	32	**81.9**
d15mp2	2647.46	14	7	**189**	3.3	110.7	42	1112	821.1	2883.0	**5**	369	204.1	544.1	**5**	**63.6**
d15mq1	3847.06	14	**49**	**434**	17.9	**877.6**	63	1303	1969.0	4742.0	56	1288	1777.0	4331.0	-	-
d15mq2	3698.49	20	45	**528**	19.6	**821.9**	**44**	966	1552.0	3994.0	89	818	1430.0	3181.0	-	-
d20mp1	2619.66	16	38	216	237.3	**1171.0**	**37**	**203**	1097.0	1475.0	47	212	1311.0	1608.0	-	-
d20mp2	2628.46	13	**18**	184	117.5	**354.0**	28	**118**	688.7	841.3	26	129	748.0	895.4	-	-
d20mq1	3828.50	14	118	365	320.1	**1419.0**	136	417	4641.0	5671.0	**48**	**291**	2371.0	3044.0	-	-
d20mq2	3685.49	20	59	**84**	102.6	**1375.0**	61	109	1535.0	2301.0	**25**	104	1030.0	1733.0	-	-

Table 6: Comparison of models CUT_x^F, CUT_x^R, CUT_{sa} and HOP for $H = 7$. The best running times are shown in bold.

			CUT_x^F				CUT_x^R				CUT_{sa}				HOP	
Inst.	OPT	$\|F_0\|$	BB	$Cuts$	t_{sep}	t $[s]$	BB	$Cuts$	t_{sep}	t $[s]$	BB	$Cuts$	t_{sep}	t $[s]$	BB	t [s]
c5mp1	2692.66	19	23	**275**	0.8	**89.0**	**13**	907	75.7	711.5	33	913	87.6	780.9	33	102.0
c5mp2	2692.46	16	27	**287**	0.7	68.4	**7**	687	53.8	393.3	15	638	63.0	412.2	17	**60.2**
c5mq1	3906.98	12	62	**337**	1.7	**196.0**	70	1569	655.1	2064.0	59	1271	539.9	2897.0	**54**	222.4
c5mq2	3769.56	23	95	**353**	2.0	**309.0**	**35**	1837	731.3	2316.0	**35**	1601	607.8	1756.0	59	422.9
c10mp1	2661.66	18	41	**731**	8.0	778.0	**19**	1503	419.2	5641.0	26	1555	417.9	6585.0	31	**97.5**
c10mp2	2661.46	15	21	**567**	4.3	337.0	15	1160	321.5	2878.0	**7**	909	222.4	1799.0	84	**60.7**
c10mq1	3867.57	14	**35**	**678**	10.2	836.8	47	2063	1728.0	5759.0	47	1395	1023.0	3704.0	47	**197.8**
c10mq2	3732.56	20	91	**926**	18.8	1201.0	57	2530	2445.0	6202.0	**37**	1850	1643.0	4884.0	62	**331.3**
c15mp1	2634.66	17	35	798	35.4	892.2	18	905	666.8	1808.0	**15**	**714**	491.2	1374.0	215	**170.4**
c15mp2	2640.46	14	**7**	**320**	9.7	190.0	11	433	299.2	601.8	17	472	356.4	664.5	160	**107.1**
c15mq1	3842.50	14	**37**	**937**	46.4	**1005.0**	53	1040	1650.0	3298.0	51	1250	1956.0	4342.0	-	-
c15mq2	3689.56	20	61	**370**	23.1	**616.1**	**45**	820	1515.0	2607.0	**45**	890	1679.0	3210.0	-	-
c20mp1	2618.66	16	**46**	**127**	67.9	**550.5**	67	291	1940.0	2268.0	54	215	1216.0	1475.0	-	-
c20mp2	2626.46	14	**23**	**122**	53.0	**220.0**	40	159	966.9	1172.0	44	169	1077.0	1242.0	-	-
c20mq1	3826.50	14	55	**225**	134.5	**987.0**	**39**	264	2689.0	3404.0	63	293	2818.0	3904.0	-	-
c20mq2	3686.49	20	128	**287**	226.2	**1143.0**	136	352	4305.0	5300.0	**98**	403	3951.0	5454.0	-	-
d5mp1	2677.94	20	23	**344**	1.2	98.2	11	743	79.5	574.8	**9**	801	90.3	736.8	11	**79.9**
d5mp2	2713.63	15	23	**349**	1.3	**72.2**	**13**	625	79.8	280.5	29	724	114.9	823.7	15	78.5
d5mq1	3878.98	17	33	**326**	1.5	278.6	29	679	272.9	2037.0	**25**	656	249.0	1747.0	33	**232.2**
d5mq2	3741.49	20	25	**372**	1.9	503.2	**23**	1214	504.8	4373.0	31	982	432.1	3314.0	27	**311.3**
d10mp1	2678.94	19	**21**	**659**	7.0	450.3	33	1756	575.0	9055.0	45	1511	458.6	7673.0	55	**113.2**
d10mp2	2682.46	15	21	**494**	4.7	364.8	**11**	1055	310.8	2542.0	15	1012	307.6	2712.0	15	**74.0**
d10mq1	3869.06	16	69	**850**	12.9	1908.0	**45**	1846	1399.0	15560.0	67	1789	1475.0	14080.0	77	**246.6**
d10mq2	3724.49	22	65	**794**	13.3	2211.0	35	1769	1333.0	16310.0	**33**	1514	1201.0	15950.0	77	**327.7**
d15mp1	2635.66	17	23	**367**	28.8	487.8	19	586	849.1	2046.0	**15**	512	638.8	1709.0	19	**151.7**
d15mp2	2647.46	14	15	**367**	20.6	346.8	13	549	691.9	1416.0	**7**	467	467.2	1013.0	11	**139.6**
d15mq1	3844.50	14	43	**872**	66.6	**1223.0**	**39**	1040	2519.0	4710.0	67	1227	2816.0	9154.0	-	-
d15mq2	3698.49	20	32	**659**	57.8	**1395.0**	**29**	799	1816.0	3873.0	53	717	1787.0	3338.0	-	-
d20mp1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
d20mp2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
d20mq1	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-
d20mq2	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-

Table 7: Comparison of models CUT_x^F, CUT_x^R, CUT_{sa} and HOP for $H = 10$. The best running times are shown in bold.

[12] Gouveia, L. 1996. Multicommodity flow models for spanning trees with hop constraints. *Eur. J. Oper. Res.* **95**(1) 178–190.

[13] Gouveia, L. 1998. Using variable redefinition for computing lower bounds for minimum spanning and Steiner trees with hop constraints. *INFORMS J. Comput.* **10**(2) 180–188.

[14] Gouveia, L. 1999. Using hop-indexed models for constrained spanning and Steiner tree models. Brunhilde Sanso, Patrick Soriano, eds., *Telecommunications network planning*. Kluwer, 21–32.

[15] Gouveia, L., L. Simonetti, E. Uchoa. 2011. Modeling hop-constrained and diameter-constrained minimum spanning tree problems as steiner tree problems over layered graphs. *Math. Programming* **128**(1-2) 123–148.

[16] Koch, T., A. Martin. 1998. Solving Steiner tree problems in graphs to optimality. *Networks* **32** 207–232.

[17] Kompella, V. P., G. C. Polyzos, J. C. Pasquale. 1993. Multicast routing for multimedia communication. *IEEE/ACM Trans. Networking* **1**(3) 286–292.

[18] Kratica, J., D. Tošić, V. Filipović, I. Ljubić. 2001. Solving the simple plant location problem by genetic algorithms. *RAIRO - Oper. Res.* **35**(1) 127–142.

[19] Krick, C., H. Räcke, M. Westermann. 2003. Approximation algorithms for data management in networks. *Theory Comput. Systems* **36**(5) 497–519.

[20] Leitner, M., G. R. Raidl. 2011. Branch-and-cut-and-price for capacitated connected facility location. *J. Math. Model. Algorithms* **10**(3) 245–267.

[21] Ljubić, I. 2007. A hybrid VNS for connected facility location. T. Bartz-Beielstein, M. J. Blesa Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, M. Sampels, eds., *Hybrid Metaheuristics*, *Lecture Notes in Comput. Sci.*, vol. 4771. Springer, 157–169.

[22] Ljubić, I., S. Gollowitzer. 2010. Modeling the hop constrained connected facility location problem on layered graphs. M. Haouari, A. R. Mahjoub, eds., *Proc. of the Internat. Sympos. on Combin. Optim. (ISCO)*, *Electronic Notes in Discrete Math.*, vol. 36. Elsevier, 207–214.

[23] Magnanti, T., L. Wolsey. 1995. Optimal trees. *Handbook in Oper. Res. and Management Sci.* 503–615.

[24] Mahdian, M., Y. Ye, J. Zhang. 2006. Approximation algorithms for metric facility location problems. *SIAM J. Comput.* **36**(2) 411–432.

[25] Manyem, P. 2009. Constrained spanning, Steiner trees and the triangle inequality. Charles Pearce, Emma Hunt, eds., *Optimization*, *Springer Optimization and Its Applications*, vol. 32. Springer New York, 355–367.

[26] Manyem, P., M. F. M. Stallmann. 1996. Some approximation results in multicasting. Tech. rep., North Carolina State University at Raleigh, Raleigh, NC, USA.

[27] Max-Planck-Institut für Informatik. 2003. UflLib. URL `http://www.mpi-inf.mpg.de/departments/d1/projects/benchmarks/UflLib/`.

[28] Pathan, M., R. Buyya. 2008. A taxonomy of CDNs. Rajkumar Buyya, Mukaddim Pathan, Athena Vakali, eds., *Content Delivery Networks*. Lecture Notes in Electr. Engrg., Springer, 33–77.

[29] Ruthmair, M., G. R. Raidl. 2011. A layered graph model and an adaptive layers framework to solve delay-constrained minimum tree problems. O. Günlük, G. J. Woeginger, eds., *15th Conf. on Int. Prog. and Comb. Opt. (IPCO XV)*, *Lecture Notes in Comput. Sci.*, vol. 6655. Springer, 376–388.

[30] Tomazic, A., I. Ljubić. 2008. A GRASP algorithm for the connected facility location problem. *Internat. Sympos. on Applications and the Internet (SAINT)*. IEEE Computer Society, 257–260.

[31] Voß, S. 1999. The Steiner tree problem with hop constraints. *Ann. Oper. Res.* **86** 321–345.

Enhanced Formulations and Branch-and-Cut for the Two Level Network Design Problem with Transition Facilities

Stefan Gollowitzer[a], Luis Gouveia[b,1], Ivana Ljubić[a,2,*]

[a] *Department of Statistics and Operations Research, University of Vienna, Brünnerstraße 72, A-1210 Vienna, Austria*

[b] *DEIO/CIO, Faculdade de Ciências da Universidade de Lisboa, Bloco C2, Campo Grande, 1749-016 Lisboa, Portugal*

Abstract

We consider a new combinatorial optimization problem that combines network design and facility location aspects. Given a graph with two types of customers and two technologies that can be installed on the edges, the objective is to find a minimum cost subtree connecting all customers while the primary customers are served by a primary subtree that is embedded into the secondary subtree. In addition, besides fixed link installation costs, facility opening costs, associated to each node where primary and secondary subtree connect, have to be paid. The problem is called the *Two Level Network Design Problem with Transition Facilities* (TLNDF).

We first model the problem on an extended graph where an additional set of arcs corresponds to the installation of node facilities and propose a cut set based model for the TLNDF that is defined on this extended graph. We present several theoretical results relating families of cut set inequalities on the extended graph with subfamilies of cut set inequalities on the original graph. We then show how a standard multi-commodity flow model defined on the original graph can be strengthened using disaggregation "by technology". We prove that the disaggregated compact formulation on the original graph provides the same lower bound as the cut set formulation on the extended graph.

We develop a branch-and-cut algorithm for solving the TLNDF. The performance of this algorithm is improved by separating subfamilies of cut set inequalities on the original graph. Our computational study confirms the efficiency and applicability of the new approach.

Keywords: OR in telecommunications, Integer programming, Linear programming relaxations, Hierarchical network design, Tree-tree networks, Network design and facility location

---

1. Introduction

The *Multi-Level Network Design Problem* (MLND) has been originally defined by Balakrishnan et al. [1]: We are given an undirected graph with a set of nodes partitioned into L levels and a set of

*Corresponding author. Tel: +43-1-4277-38661 Fax: +43-1-4277-38699

Email addresses: `stefan.gollowitzer@univie.ac.at` (Stefan Gollowitzer), `legouveia@fc.ul.pt` (Luis Gouveia), `ivana.ljubic@univie.ac.at` (Ivana Ljubić)

[1]Supported by National Funding from FCT - Fundação para a Ciência e Tecnologia, under the project: PEst-OE/MAT/UI0152.

[2]Supported by the APART Fellowship of the Austrian Academy of Sciences (OEAW).

edges such that along each edge one of L different technologies can be installed, with higher grade technologies inducing higher fixed costs. The goal is to find a (spanning) subtree and decide which technology to install along each edge, so that all customers at level ℓ can communicate with each other along a path using technology of grade ℓ or higher. We extend the definition of the MLND by introducing the fixed costs for *transition nodes*, i.e., the nodes where a change of technology takes place, and by considering so-called potential Steiner nodes which are nodes that need not be included in the solution. In this problem, which we denote by *Multi-Level Network Design Problem with Transition Facilities*, the overall goal is to find a MLND subtree that minimizes the sum of fixed edge and facility installation costs. In this paper we study the case with $L = 2$, which we will denote by the *Two Level Network Design Problem with Transition Facilities* (TLNDF).

TLNDF arises in the topological design of hierarchical communication, transportation, and electric power distribution networks. One of the most important applications of TLNDF is in the context of telecommunication networks, where networks with two cable technologies, fiber optic and copper, are built. Telecommunication companies distinguish between *primary* and *secondary customers*. The switching centers, important infrastructure nodes and small businesses are considered as primary customers (i.e., those to be served by fiber optic connections). Single households are not considered as being consumers of a high potential and hence they only need to be supplied using copper cables. The secondary technology is much cheaper, but the guaranteed quality of the connections and bandwidth is significantly below the quality provided by the primary technology. The goal is to build a network (with tree topology) such that there is a fiber optic connection between each primary customer and a designated root node (e.g., a central office), and each secondary customer is connected to the root along a path using either of the two technologies. Typically, at transition nodes, expensive switching devices need to be installed to transmit the electrical into optical signal, and the respective purchasing and equipment operating costs are not negligible. This particular application involves two new features that have not been considered in the previous literature (see, e.g., Balakrishnan et al. [2], Duin and Volgenant [10]). First, the application considers additional *transition costs* due to the presence of two technologies on the network. Second, in graphs that represent telecommunication networks nodes like street intersections need to be considered as well, i.e., we need to allow that the set of primary and secondary customers is a proper subset of the set of nodes, and a subset of remaining nodes may be a part of the solution, if it helps in establishing a cheaper connection. Those remaining nodes will be referred to as *potential Steiner nodes*.

More formally, the problem can be defined as follows:

Definition 1 (TLNDF)**.** We are given an undirected graph $G = (V, E)$ with a set of customers $R \subseteq V$. To each edge $e \in E$ we associate two installation costs, $c_e^1 \geq c_e^2 \geq 0$. These correspond to the *primary* and *secondary technology*, respectively. The set of customers, R, is partitioned into the sets of *primary* and *secondary customers* P and S, respectively ($P \cap S = \emptyset$, $P \cup S = R$). We are also given a root node $r \in V$, otherwise we choose one of the primary customers as such. To each node $i \in V$ we associate *facility opening cost* $d_i \geq 0$ that needs to be paid if i is used as a transition node.

Our goal is to determine a *subtree* T (built of a set of primary and secondary edges, T_1 and T_2, respectively) with the set F of transition nodes (i.e., nodes that are adjacent to edges from both T_1 and T_2), satisfying the following properties:

(P) Each *primary node* in P is connected to the root node by a path that consists of T_1 edges only,

(S) each *secondary node* in S is connected to the root by a path consisting of edges from $T_1 \cup T_2$,

(F) facilities need to be open at each transition node $i \in F$ and

(M) the sum of fixed edge and facility installation costs

$$\sum_{e \in T_1} c_e^1 + \sum_{e \in T_2} c_e^2 + \sum_{i \in F} d_i$$

is minimized.

Figure 1 illustrates a solution of the TLNDF. It uses the following symbols: Squares represent primary customers, triangles represent secondary customers, dots represent potential Steiner nodes. A grey fill indicates a transition node. Solid lines indicate the installation of primary edges (e.g., fiber-optic technology) and grey dotted lines indicate the installation of secondary edges (e.g., copper wires).

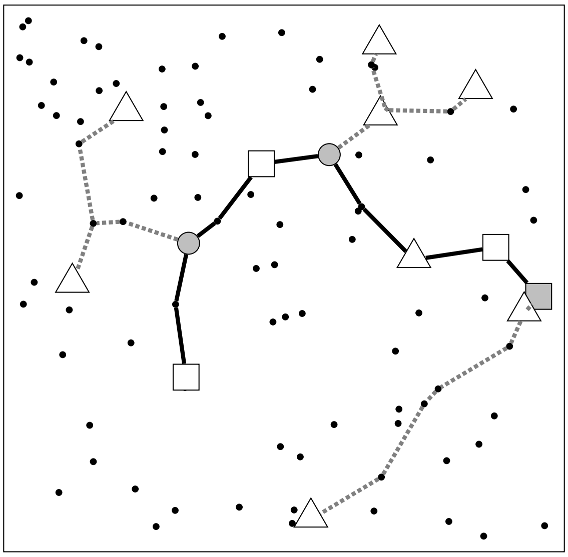


Figure 1: Example of a TLNDF solution

The following observation can be made about optimal solutions of the TLNDF: i) Since $c_e^1 \geq c_e^2$ for all $e \in E$, there always exist an optimal solution in which the subgraph induced by T_1 is a rooted subtree of T (*primary subtree*) and the subgraph induced by T_2 is a forest (a union of *secondary subtrees*) attached to it. ii) If facility opening costs are uniform for all nodes, leaves of the primary subtree are nodes from $P \cup S$. In addition, any leaf of the primary subtree that has a secondary subtree attached to it will be a primary node. iii) Otherwise, if facility opening costs are location-dependent, placing facilities at locations of Steiner nodes or secondary customers may provide cheaper solutions, i.e., a secondary subtree can be attached to any node from V, and henceforth, a leaf of the primary subtree can be any node from V.

Notice also that our general definition covers the case in which potential facility locations are a true subset of V (which can be modeled by setting $d_i := \infty$ for the non-facility locations).

This important problem generalizes problems with tree-star and star-tree topologies including connected facility location, hierarchical network design, Steiner trees and uncapacitated facility location.

Overview of the paper. In Section 2 we model the TLNDF in an extended graph, where an additional set of arcs corresponds to the installation of node facilities. We present several theoretical results relating families of cut set inequalities on the extended graph with special families of cut set inequalities on the original graph. Cut set inequalities on the extended graph can be separated in polynomial time using maximum flow algorithms. We also study special classes of cut set inequalities that are obtained by projecting subsets of constraints of the extended graph formulation on the original graph. In Section 4 we show how these can be separated efficiently on the original graph extended by a single node. Our computational study, reported in Section 5, confirms the efficiency and applicability of these separation procedures.

It is quite straightforward to model the TLNDF on the original graph with a standard multi-commodity flow model. The linear programming (LP) relaxation of the cut set model on the extended graph is easily shown to dominate the LP relaxation of this flow model. This dominance result is also known from similar problems (see, e.g., [4]). In this paper (cf. Section 3), we show that by disaggregating the previous multi-commodity flow formulation *by technology* we obtain a formulation on the original graph that provides the same lower bound as the cut set formulation on the extended graph. Our result also extends to the two level network design problem *without* transition nodes and, as far as we know, this is the first time a compact formulation on the original graph is given that provides the same LP bound as the cut set formulation on the extended graph. Preliminary results of this work appeared in Gollowitzer et al. [11].

1.1. Literature review

The concept of two level network design problems (more precisely, *two-level spanning trees*) has been developed in the 80's and early 90's.

Hierarchical network design (HNDP). The *hierarchical network design* problem, in which $R = V$ and $|P| = 2$, was the "initial" variant of the TLND introduced by Current et al. [8]. The authors proposed an integer programming model based on subtour elimination constraints and a heuristic for the problem. Later, Duin and Volgenant [9] proved structural properties of HNDP that enable reductions of the problem graph and elimination of variables from an integer programming model. Pirkul et al. [22] derived a heuristic based upon a Lagrangian relaxation of a flow-based formulation for the problem. A dynamic programming procedure that finds suboptimal solutions was then proposed by Sancho [24]. Recently, Obreque et al. [21] proposed a branch-and-cut algorithm for this problem.

Two level network design (TLND). This problem, a generalization of HNDP in which $|P| \geq 2$ and $R = V$, was introduced by Duin and Volgenant [10]. Balakrishnan et al. [2] proposed several network flow based models for TLND and compared the LP bounds of the proposed formulations. The same authors also proposed a composite heuristic that provides an approximation ratio of 4/3 if the embedded Steiner tree is solved to optimality and $c_e^1/c_e^2 = q > 1$ for all $e \in E$. The approximation ratio is $\frac{4}{4-\rho}$ if the Steiner tree problem is solved with an approximation ratio of $\rho < 2$. For non-proportional edge costs, this ratio becomes $\rho + 1$. Balakrishnan et al. [1] tested a dual ascent method derived on the strongest formulation proposed in [2] (which is a directed multi-commodity flow formulation on G, cf. Section 3). Gouveia and Telhada [15] proposed another formulation in which the primary subtree is modeled as a directed arborescence embedded into the secondary spanning arborescence. The authors proposed to solve the problem using a Lagrangian relaxation based method. In a later paper, Gouveia and Telhada [16] improved this formulation by using a "reformulation by intersection" concept to derive a new compact formulation whose lower bounds are at least as strong as the strongest ones proposed in [2]. In a recent work, Chopra and Tsai [4] developed a branch-and-cut approach for a generalization of the TLND with more than two levels.

Hierarchical network design with transshipment facilities (HNDF). The HNDP was introduced by Current [6]. In this problem additional transshipment costs need to be paid for each node of the primary path whenever a change of technology takes place. The main difference between the HNDF and the TLNDF, besides the restriction $|P| = 2$ and $R = V$, is that secondary nodes included in the primary path are not considered as "served", and therefore they also need to be connected by a (possibly empty) path to a transshipment facility. In addition, the union of primary and secondary edge sets may form a cycle, i.e., the optimal solution is not necessarily a tree.

Current [6] proposed a heuristic approach to HNDF in which, for a given root r and terminal node t, K shortest paths are calculated. For each of these paths an auxiliary problem is constructed, in which the nodes of the path are connected to a dummy root node by edges whose weights are set to their facility opening costs. The edges of the path are deleted and in the graph obtained by this procedure a minimum spanning tree using secondary edge costs is calculated. Later, Current and Pirkul [7] described a new formulation of the problem based on the introduction of the dummy root node (as above) and provided computational results for two Lagrangian based heuristics derived from this model.

Combined network design and facility location and other related problems. A large body of literature exists on problems that combine network design and facility location decisions. Contreras and Fernández [5] give a unifying framework for many well-studied problems including the p-median problem, hub location problems and the ring-star problem. They give an exhaustive overview of the existing literature and analyze modeling aspects and algorithmic ideas. Most problems considered there satisfy the assumption that nodes are customers, potential facility location or both. Among the problems combining network design and facility location, the *connected facility location problem* (ConFL) (see, e.g., [12]) is the closest to the TLNDF. More precisely, ConFL is a special case of the TLNDF where the secondary subtrees are stars. The TLNDF problem also belongs to a class of problems with a tree-tree topology. The reader is referred to a survey by Gourdin et al. [13], who describe several variants of related problems such as star-tree, tree-star and star-star problems as well as other variants of tree-tree problems.

1.2. Notation

It is known that for rooted spanning or Steiner tree problems, modeling the problem on a directed graph provides models whose LP bounds are stronger than the bounds of their undirected counterparts (see, e.g., [19]). Henceforth we will consider a directed graph $G = (V, A)$ that is obtained from the original undirected graph $G = (V, E)$ as follows: Instead of each edge $e = \{i, j\} \in E$ we use two arcs ij and ji in A, both of which are assigned the cost of the original edge. Since a solution on the undirected graph corresponds to an arborescence directed away from the root node, edges $\{r, j\}$ are replaced by a single arc rj.

In our models we will use the following binary variables:

$$x_{ij}^1 = \begin{cases} 1, & \text{if the primary technology is installed on arc } ij \\ 0, & \text{otherwise} \end{cases} \qquad \forall ij \in A$$

$$x_{ij}^2 = \begin{cases} 1, & \text{if the secondary technology is installed on arc } ij \\ 0, & \text{otherwise} \end{cases} \qquad \forall ij \in A, j \notin P$$

$$z_i = \begin{cases} 1, & \text{if a facility is installed on node } i \\ 0, & \text{otherwise} \end{cases} \qquad \forall i \in V$$

Observe that no feasible solution will contain secondary arcs pointing to a primary node (i.e., $x_{ij}^2 = 0$ for $j \in P$). We will ignore the variables corresponding to these arcs in our models but, to simplify the notation, we will allow them in the indexation of the summation terms.

For a set $W \subseteq V$, we will write $z(W) = \sum_{i \in W} z_i$. For any $W \subset V$ we denote its complement set by $W^c = V \setminus W$. For any $M, N \subset V$, $M \cap N = \emptyset$, we denote the induced cut set of arcs by $(M, N) = \{ij \in A \mid i \in M, j \in N\}$. In particular, let $\delta^-(W) = (W^c, W)$ and $\delta^-(i) = (V \setminus \{i\}, \{i\})$. For a set of arcs $\hat{A} \subseteq A$, we will write $x^\ell(\hat{A}) = \sum_{ij \in \hat{A}} x_{ij}^\ell$, for $\ell = 1, 2$, and $(x^1 + x^2)(\hat{A}) = \sum_{ij \in \hat{A}} (x_{ij}^1 + x_{ij}^2)$.

We will describe models based on these variables, but in the next sections several models using other variables will be described as well. In order to relate all of these models, for a mixed integer programming model M let $\mathcal{P}_{\mathbf{a^1},\dots,\mathbf{a^n}}(M)$ denote the orthogonal projection of the convex hull of LP solutions of M onto the space defined by variables $\mathbf{a^1}, \dots, \mathbf{a^n}$.

The illustrations in the next sections use the following symbols in addition to the ones previously described: $\boxed{\text{r}}$ represents the root node and, whenever we solve a problem as the Steiner tree problem, terminals are denoted by $\Diamond$.

2. Cut set-based formulations

In Gollowitzer et al. [11] we show that the TLNDF can be modeled as a Steiner arborescence problem on an extended graph with additional node degree constraints. In this section we first recall a cut set formulation on the original graph, then we provide the definition of the extended graph and state the most important results taken from [11]. Finally, we present a new result that characterizes the inequalities obtained by projecting cut set constraints from the extended graph into the natural space of variables $(\mathbf{x}^1, \mathbf{x}^2, \mathbf{z})$.

2.1. The cut set formulation on the original graph

We recall the following formulation of the TLNDF, that we first introduced in [11].

$$(TLNDF) \quad \min \sum_{ij \in A} (c_{ij}^1 x_{ij}^1 + c_{ij}^2 x_{ij}^2) + \sum_{i \in V} d_i z_i$$

$$x^1(\delta^-(W)) \geq 1 \qquad \forall W \subseteq V \setminus \{r\},\ W \cap P \neq \emptyset \tag{x1}$$

$$(x^1 + x^2)(\delta^-(W)) \geq 1 \qquad \forall W \subseteq V \setminus \{r\},\ W \cap S \neq \emptyset \tag{x12}$$

$$(x^1 + x^2)(\delta^-(i)) \leq 1 \qquad \forall i \in V \setminus \{r\} \tag{1}$$

$$z_j + \sum_{ij \in A, i \neq k} x_{ij}^2 \geq x_{jk}^2 \qquad \forall jk \in A, k \notin P \tag{2}$$

$$x^1_{ij}, x^2_{ij} \in \{0,1\} \quad \forall ij \in A \tag{3}$$

$$z_i \in \{0,1\} \quad \forall i \in V \tag{4}$$

The *primary connectivity constraints* (x1) ensure that for every primary node i, there is a path between r and i containing only primary arcs. The *secondary connectivity constraints* (x12) ensure that every secondary node is connected to the root by a path containing primary and/or secondary arcs. The in-degree constraints (1) ensure that the overall solution is a subtree. Together with connectivity constraints (x12), the *basic coupling constraints* (2) guarantee that if a facility is installed at node j, then j is the root of a secondary subtree.

2.2. The cut set formulation on the extended graph

The *extended graph* $G_{NS} = (V_{NS}, A_{NS})$, with the root r' and the set of terminals R_{NS}, is defined as follows:

$$\begin{aligned}
V_{NS} :=& V' \cup V'' \cup S \text{ where} & A_{NS} :=& A' \cup A'' \cup A_z \cup A_S \text{ where}\\
& V' := \{i' \mid i \in V\}, & & A' := \{i'j' \mid ij \in A\},\\
& V'' := \{i'' \mid i \in V\}, & & A'' := \{i''j'' \mid ij \in A\},\\
& S \text{ is the set of secondary nodes;} & & A_z := \{i'i'' \mid i \in V\},\\
R_{NS} :=& P' \cup S \text{ where} & & A_S := \{i'i \mid i' \in V', i \in S\}\\
& P' = \{i' \mid i' \in V', i \in P\}; & & \qquad \cup \{i''i \mid i'' \in V'', i \in S\}.
\end{aligned}$$

The graph G_{NS} consists of several components:

i) A subgraph $G' = (V', A')$ corresponds to the primary network. It contains nodes and arcs that may be included in the primary subtree.

ii) A subgraph $G'' = (V'', A'')$ corresponds to the secondary network. It contains nodes and arcs that may be contained in the secondary subtrees.

iii) Arcs linking nodes in G' to the corresponding copy in G'' represent potential facilities.

iv) An additional copy of the secondary nodes (with arcs pointing from their representatives in graphs G' and G'') represents *terminals* that will make sure that each secondary node is either a part of the primary or the secondary network.

Arc costs C_{uv} for $uv \in A_{NS}$ are defined as follows:

$$C_{uv} = \begin{cases} c^1_{ij}, & u = i', v = j', ij \in A,\\ c^2_{ij}, & u = i'', v = j'', ij \in A,\\ d_i, & u = i', v = i'', i \in V,\\ 0, & \text{otherwise,} \end{cases} \qquad uv \in A_{NS}$$

We observe that if, for a primary node $i \in P$, its copy $i'' \in V''$ belongs to the optimal solution on the extended graph, then no ingoing arc of i'', except for the facility arc $i'i''$, will be used. Thus, we reduce the size of G_{NS} by removing all the arcs, except $i'i''$, leading into primary customer nodes in V''. Notice that a third copy of secondary nodes in G_{NS}, namely the set S, is needed, since the secondary customers can either be part of the primary subtree or be part of one of the secondary subtrees. The copies of secondary customers in G' and G'' are considered as potential Steiner nodes, with their third copy being a terminal. To ensure the tree topology, we will impose a restriction that for each node $i \in V$ at most one of the copies i' and i'' is allowed to have its $\mathbf{x}^1$- and $\mathbf{x}^2$-in-degree equal to one. Figure 2b) illustrates G_{NS} corresponding to the original graph shown in Figure 2a). We have the following result:

Lemma 1 ([11])**.** *The TLNDF problem can be modeled as the Steiner arborescence problem with additional node in-degree constraints on some node pairs on the graph G_{NS} with the root r' and terminal set R_{NS}.*

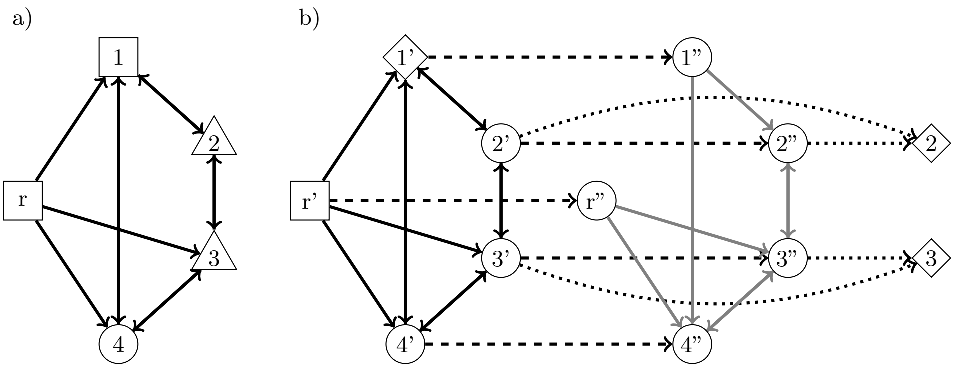


Figure 2: a) Instance of *TLNDF*; b) Transformed Steiner arborescence instance on G_{NS}.

To obtain an integer programming (IP) model, we assign binary variables X_{ij} to all arcs $ij \in A_{NS}$. Let $X(\delta^-(\tilde{W}))$ denote the sum of X variables that are in the directed cut set $(\tilde{W}^c, \tilde{W})$ in G_{NS}. Based on the classical cut set model for Steiner trees (cf. [3]) we derive the following IP formulation:

$$(SA) \quad \min \sum_{ij \in A_{NS}} C_{ij} X_{ij}$$

$$\text{s.t.} \quad X(\delta^-(\tilde{W})) \geq 1 \qquad \forall \tilde{W} \subseteq V_{NS} \setminus \{r'\}, \tilde{W} \cap R_{NS} \neq \emptyset \tag{5a}$$

$$\sum_{ij \in A} (X_{i'j'} + X_{i''j''}) \leq 1 \qquad \forall j \in V \setminus \{r\} \tag{5b}$$

$$X_{ij} \in \{0,1\} \quad \forall ij \in A_{NS} \tag{5c}$$

Constraints (5a) are connectivity cuts between the root node and each terminal. Inequalities (5b) ensure that any solution of SA does not contain the copies i' and i'' of $i \in V$ at the same time, unless the facility arc $i'i''$ is part of the solution.

Lemma 2 ([11])**.** *Cut set inequalities* (5a) *such that $\delta^-(\tilde{W}) \cap A_S \neq \emptyset$ are redundant in the model SA.*

A straightforward algorithmic approach based on the separation of inequalities (5a) might prove to be computationally expensive, since the separation of inequalities (5a) requires the solution of maximum flow problems on the graph G_{NS} with up to to $3|V|$ nodes and $2|A| + 3|V|$ arcs. This has motivated us to investigate and implement a two-phase separation method where we start by separating cut set inequalities on the original graph and only then move to separation on the extended graph. The reason for this two-phase approach is that the corresponding maximum flow algorithm for the cut set constraints on the original graph is applied to a much smaller graph.

To find these sets of inequalities on the original graph we add the following constraints to the model SA. They link the variables on the extended graph and the variables on the original graph.

$$x^1_{ij} = X_{i'j'} \qquad \forall ij \in A, i'j' \in A', \tag{6a}$$

$$x^2_{ij} = \begin{cases} X_{i''j''} \\ 0 \end{cases} \qquad \forall ij \in A, \begin{cases} i''j'' \in A'', \\ \text{otherwise} \end{cases} \tag{6b}$$

$$z_i = X_{i'i''} \qquad \forall i \in V, i'i'' \in A_z. \tag{6c}$$

Adding these equalities to the model SA will not alter its LP value but will allow us to characterize $\mathcal{P}_{\mathbf{x}^1,\mathbf{x}^2,\mathbf{z}}(SA)$.

Lemma 3. *$\mathcal{P}_{\mathbf{x}^1,\mathbf{x}^2,\mathbf{z}}(SA) = \mathcal{P}(CUT)$ where $\mathcal{P}(CUT)$ is given by the set of vectors $(\mathbf{x}^1, \mathbf{x}^2, \mathbf{z})$ satisfying*

$$(x^1 + x^2)(\delta^-(i)) \leq 1 \qquad \forall i \in V \setminus \{r\} \tag{7}$$

and

$$x^1(\delta^-(W')) + x^2(\delta^-(W'')) + z(W'' \setminus W') \geq 1 \qquad \begin{array}{l} r \notin W', W' \cap W'' \cap S \neq \emptyset \\ \text{or } W' \cap P \neq \emptyset. \end{array} \tag{x12-z}$$

where $W' = \{i \in V \mid i' \in \tilde{W}\}$ and $W'' = \{i \in V \mid i'' \in \tilde{W}\}$ for an arbitrary cut set $\tilde{W} \subseteq V_{NS} \setminus \{r'\}$ such that $\tilde{W} \cap R_{NS} \neq \emptyset$ and $\delta^-(\tilde{W}) \cap A_S = \emptyset$.

Constraints (x12-z) and (7) correspond to non redundant cut sets (5a) and constraints (5b) on the extended graph, respectively.

The following sets of inequalities are special cases of constraints (x12-z) (see Figure 3).

i) If $W' = W$ and $W'' = \emptyset$, we obtain *primary connectivity* constraints:

$$x^1(\delta^-(W)) \geq 1 \qquad \forall W \subseteq V \setminus \{r\},\ W \cap P \neq \emptyset \tag{x1}$$

ii) For $W'' = V$ and $W' = W$ we obtain constraints of the form

$$z(W^c) + x^1(\delta^-(W)) \geq 1 \qquad \forall W \subseteq V \setminus \{r\}, W \cap S \neq \emptyset \tag{x1-z}$$

iii) For $W' = W'' = W$, we obtain *secondary connectivity* cuts:

$$(x^1 + x^2)(\delta^-(W)) \geq 1 \qquad \forall W \subseteq V \setminus \{r\},\ W \cap S \neq \emptyset \tag{x12}$$

iv) For $W' = \{k\}$, $k \in S$, and $W'' = W \cup \{k\}$ we obtain constraints of the form

$$z(W) + x^1(\delta^-(k)) + x^2(\delta^-(W_k)) \geq 1 \qquad \forall k \in S, W \subseteq V \setminus \{k\}, W_k = W \cup \{k\} \tag{x2-z}$$

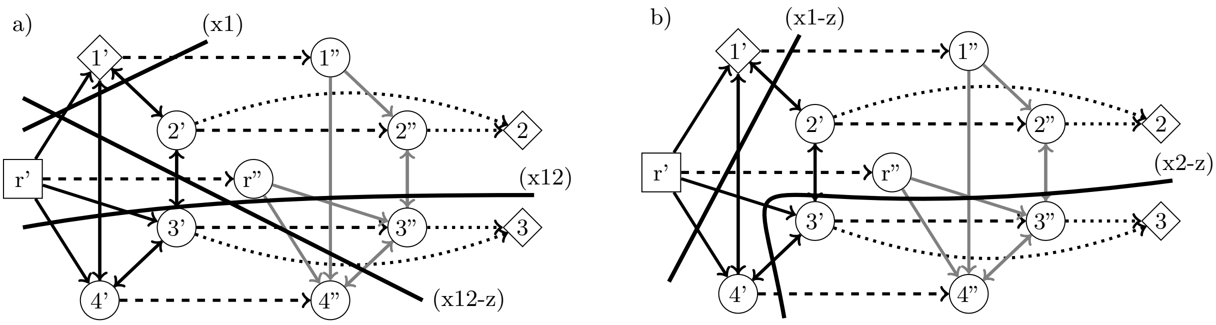


Figure 3: a) Illustration of inequalities (x1) for $W = \{1\}$, (x12) for $W = \{3, 4\}$, and (x12-z) for $W' = \{1, 2\}$, $W'' = \{r, 1, 2, 3\}$. b) Illustration of inequalities (x1-z) for $W = \{2, 3, 4\}$ and (x2-z) for $W = \{4\}$ and $k = 3$.

Constraints (x1) ((x12)) are connectivity cuts for primary (secondary) customers and ensure a path that consists of primary (primary and secondary) edges between the root node and each primary (secondary) customer. They have already been stated in [1] for the related problem without transition nodes. The remaining constraints involving $\mathbf{z}$ variables are new. Constraints (x1-z) state that for any subset of nodes that contains a secondary customer, there must either be an ingoing primary arc or an installed node facility in its complement. The interpretation of (x2-z) is not straightforward. However, these constraints were "found" in an indirect way. By subtracting $x^1(\delta^-(k))$ and $x^2(W^c, k)$ on both sides we obtain

$$z(W) + x^2(W_k^c, W) \geq x^2(W, \{k\}) \quad \forall k \in S,\ W \subseteq V \setminus \{k\}, W_k = W \cup \{k\}. \tag{x2-z'}$$

Constraints (x2-z') are a generalization of inequalities suggested in [14] for models with node transition variables. These inequalities state that if there is a secondary arc leading into node k from a node in a given node set W, then either there is a facility installed in a node in W or there is a secondary arc leading into W from the complement of $W \cup \{k\}$.

The four sets of constraints just described are the constraints that will be separated in the first phase of our branch-and-cut algorithm. Separation algorithms for these will be described in Section 4. Clearly, there are other constraints included in the general description given by (x12-z) that do not correspond to any constraint of these four sets. For these remaining constraints, it is not clear how to separate them in the original space. This is precisely the set of constraints that will be separated in the larger extended graph.

It is obvious that the model defined by (7) and (x12-z) gives a valid model for the TLNDF defined only on variables x^1, x^2 and z. Notice that another valid model is obtained by only considering the inequalities (x1), (x12) and the family (x2-z) for singleton sets W instead of the whole set (x12-z). In the next section we will present a multi-commodity flow formulation whose LP relaxation is equivalent to the LP relaxation of this latter model. However, our computational experiments (cf. Section 5) will show that this model provides much weaker bounds than the model with all general cut set inequalities (x12-z).

3. Flow-based formulations

In this section we will present a compact multi-commodity flow formulation that extends the strongest model proposed in [2] by introducing node transition variables. As we shall show later, the LP relaxation bound of this model is not as good as the one provided by the cut set model on the extended graph (presented in Lemma 3). A similar dominance result is known for the TLNDF problem without node transition costs and is given in Chopra and Tsai [4]. Therefore, we propose a new model based on multi-commodity flows that is obtained from the previous one by disaggregating the variables and constraints "by technology". We show that the LP relaxation of the new disaggregated model is equally strong as the LP relaxation of the cut set model on the extended graph.

3.1. Multi-commodity flow formulation

Let us define the following flow variables: $f_{ij}^k \geq 0$ corresponds to the flow from r to the commodity $k \in R$, using arc $ij \in A$. A multi-commodity formulation for the classical two level network design problem (without transition nodes) is the following (cf., e.g., [1]):

$$
\begin{aligned}
(MCF) \quad & \min \sum_{ij\in A} c_{ij}^1 x_{ij}^1 + \sum_{ij\in A} c_{ij}^2 x_{ij}^2 \\
& \sum_{ji\in A} f_{ji}^k - \sum_{ij\in A} f_{ij}^k = \begin{cases} 1 & i = k \\ -1 & i = r \\ 0 & i \neq k, r \end{cases} \qquad \forall i \in V,\ \forall k \in R \\
& 0 \leq f_{ij}^k \leq x_{ij}^1 \qquad \forall ij \in A, k \in P \\
& 0 \leq f_{ij}^k \leq x_{ij}^1 + x_{ij}^2 \qquad \forall ij \in A, k \in S \\
& x_{ij}^1, x_{ij}^2 \in \{0,1\} \qquad \forall ij \in A
\end{aligned}
$$

An extension to model transition node costs is obtained by changing the objective function to

$$\min \sum_{ij\in A} c_{ij}^1 x_{ij}^1 + \sum_{ij\in A} c_{ij}^2 x_{ij}^2 + \sum_{i\in V} d_i z_i$$

and adding the following, previously described compact sets of constraints to MCF:

$$
\begin{aligned}
(x^1 + x^2)(\delta^-(i)) \leq 1 & \qquad \forall i \in V \setminus \{r\} \\
z_j + \sum_{ij\in A, i\neq k} x_{ij}^2 \geq x_{jk}^2 & \qquad \forall jk \in A, k \notin P \\
z_i \in \{0,1\} & \qquad \forall i \in V
\end{aligned}
$$

For simplicity we maintain the same designation MCF for this model.

3.2. Disaggregated multi-commodity flow formulation

We now show how to strengthen the MCF model by disaggregating the variables f^k by technology. Consider the following two types of flow variables: $f_{ij}^{1k} \geq 0$ ($f_{ij}^{2k} \geq 0$) correspond to the primary (secondary) flow from r to the commodity $k \in R$, using arc $ij \in A$.

Consider then the following model, which extends a model described in Gouveia and Janssen [14] by node transition variables.

$$(dMCF) \quad \min \sum_{ij\in A} c^1_{ij}x^1_{ij} + \sum_{ij\in A} c^2_{ij}x^2_{ij} + \sum_{i\in V} d_i z_i$$

$$\text{s.t.} \quad \sum_{ji\in A} f^{1k}_{ji} - \sum_{ij\in A} f^{1k}_{ij} = \begin{cases} 1 & i=k \\ -1 & i=r \\ 0 & i\neq k,r \end{cases} \qquad \forall i\in V,\ \forall k\in P \tag{8a}$$

$$\sum_{ji\in A}(f^{1k}_{ji}+f^{2k}_{ji}) - \sum_{ij\in A}(f^{1k}_{ij}+f^{2k}_{ij}) = \begin{cases} 1 & i=k \\ -1 & i=r \\ 0 & i\neq k,r \end{cases} \qquad \forall i\in V,\ \forall k\in S \tag{8b}$$

$$z_i + \sum_{ji\in A} f^{2k}_{ji} \geq \sum_{ij\in A} f^{2k}_{ij} \qquad \forall k\in S, \forall i\in V, i\neq k \tag{8c}$$

$$(x^1+x^2)(\delta^-(i)) \leq 1 \qquad \forall i\in V \tag{8d}$$

$$0 \leq f^{1k}_{ij} \leq x^1_{ij} \qquad \forall ij\in A, k\in R \tag{8e}$$

$$0 \leq f^{2k}_{ij} \leq x^2_{ij} \qquad \forall ij\in A, k\in S \tag{8f}$$

$$z_i, x^1_{ij}, x^2_{ij} \in \{0,1\} \qquad \forall i\in V, ij\in A \tag{8g}$$

In the context of this model the flow variables f^{1k} and f^{2k} can be reinterpreted as indicating whether arc ij has technology 1 or 2 installed and whether it is in the path to node k. The new constraints (8c) state that a facility needs to be installed when the technology used on the arcs changes on the path to node k.

Equations (8a) and (8b) ensure one unit of primary (primary or secondary) flow to primary (secondary) customer nodes. Constraints (8d) limit the number of ingoing arcs for each node to one and inequalities (8e) and (8f) link the flow variables and design variables.

3.3. Polyhedral comparison

Next we will show that formulation *dMCF* on the original graph G provides the same LP bound as the cut set model on the extended graph given in Section 2.2. To prove this result we will introduce an auxiliary model and prove that the two models, the cut set model on the extended graph and the flow model *dMCF* provide the same LP bound as the auxiliary model.

The auxiliary model is a straightforward multi-commodity flow reformulation of the cut set model on the extended graph. To define this model we consider the following sets of variables. i) Flow variables f^{1k}_{ij} and f^{2k}_{ij}, that correspond to the flow from r to the commodity $k\in R$, using arcs $i'j'\in A'$ or $i''j''\in A''$, respectively. ii) Variables y^k_i correspond to the flow of commodity k sent through $i'i''\in A_z$. iii) Finally, for all $k\in S$, we also define f^{1k}_k and f^{2k}_k to be the flow values on arcs $k'k$ and $k''k$ in A_S, respectively. Obviously, $f^{1\ell}_k = f^{2\ell}_k = 0$ for all $k\neq \ell$.

$$(MCF_{NS}) \quad \min \sum_{ij\in A} c^1_{ij}x^1_{ij} + \sum_{ij\in A} c^2_{ij}x^2_{ij} + \sum_{i\in V} d_i z_i$$

s.t. (8a), (8d), (8e), (8f), (8g),

$$y^k_r + \sum_{rj\in A} f^{1k}_{rj} = 1 \qquad \forall k\in S \tag{9a}$$

$$y_r^k - \sum_{rj \in A} f_{rj}^{2k} = 0 \qquad \forall k \in S \tag{9b}$$

$$\sum_{ji \in A} f_{ji}^{1k} - \sum_{ij \in A} f_{ij}^{1k} - y_i^k = 0 \qquad \forall k \in S, i \in V \setminus \{r,k\} \tag{9c}$$

$$y_i^k + \sum_{ji \in A} f_{ji}^{2k} - \sum_{ij \in A} f_{ij}^{2k} = 0 \qquad \forall k \in S, i \in V \setminus \{r,k\} \tag{9d}$$

$$\sum_{jk \in A} (f_{jk}^{1k} + f_{jk}^{2k}) = 1 \qquad \forall k \in S \tag{9e}$$

$$0 \le y_i^k \le z_i \qquad \forall i \in V, k \in R \tag{9f}$$

Using the linking constraints (6) and $X_{i'i} = x^1(\delta^-(i))$ and $X_{i''i} = 1 - X_{i'i}$ for all $i \in S$, the max-flow min-cut theorem implies the following

Lemma 4. $\mathcal{P}(SA) = \mathcal{P}_X(MCF_{NS})$.

We can also show the following

Lemma 5. $\mathcal{P}_{\mathbf{x}^1,\mathbf{x}^2,\mathbf{z},\mathbf{f}^1,\mathbf{f}^2}(MCF_{NS}) = \mathcal{P}(dMCF)$.

Proof. We show the claim by mutual inclusion:

Let $(\mathbf{x}^1, \mathbf{x}^2, \mathbf{z}, \mathbf{f}^1, \mathbf{f}^2, \mathbf{y}) \in \mathcal{P}(MCF_{NS})$. Then $(\mathbf{x}^1, \mathbf{x}^2, \mathbf{z}, \mathbf{f}^1, \mathbf{f}^2) \in \mathcal{P}(dMCF)$: By eliminating y_i^k from (9a) and (9b) ((9c) and (9d)) we obtain constraints (8b) for the case $i = r$ ($i \neq k, r$). Constraints (8b) for $i = k$ are equivalent to (9e). Limiting y_i^k from above in (9b) and (9d) using (9f) we obtain (8c).

Let now $(\mathbf{x}^1, \mathbf{x}^2, \mathbf{z}, \mathbf{f}^1, \mathbf{f}^2) \in \mathcal{P}(dMCF)$ and variables y_i^k be defined as follows.

$$y_i^k := \sum_{ij \in A} f_{ij}^{2k} - \sum_{ji \in A} f_{ji}^{2k} \quad \forall k \in S, i \in V \setminus \{r,k\} \text{ and}$$

$$y_r^k := \sum_{j \in A} f_{rj}^{2k} \quad \forall k \in S.$$

Then one can easily verify that $(\mathbf{x}^1, \mathbf{x}^2, \mathbf{z}, \mathbf{f}^1, \mathbf{f}^2, \mathbf{y}) \in \mathcal{P}(MCF_{NS})$. □

The two preceding lemmata imply the following

Theorem 1. $\mathcal{P}_{\mathbf{x}^1,\mathbf{x}^2,\mathbf{z}}(dMCF) = \mathcal{P}_{\mathbf{x}^1,\mathbf{x}^2,\mathbf{z}}(SA)$.

4. Branch-and-cut framework

Since our models comprise an exponential number of constraints, we solve them using the cutting plane technique embedded into a branch-and-bound framework, commonly known as the *branch-and-cut* approach. Non-standard ingredients of our approach are described below. The primal heuristic that we use is described in [11].

4.1. Initialization

To reduce the number of separated cut set constraints and improve the general performance, in our computational experiments we initialize all models with degree constraints (1), coupling constraints (2) and the following sets of inequalities:

$$(x^1 + x^2)(\delta^-(j)) \leq (x^1 + x^2)(\delta^+(j)) \qquad \forall j \in V \setminus R \tag{10a}$$

$$\sum_{ij \in A, i \neq k} x^1_{ij} \geq x^1_{jk} \qquad \forall jk \in A, j \in V \setminus \{r\} \tag{10b}$$

Inequalities (10a) are strengthening *degree balance constraints* for potential Steiner nodes. Inequalities (10b) guarantee that for each outgoing primary arc of a node $j \in V \setminus \{r\}$ there is at least one primary arc entering this node.

4.2. Cut separation

In Section 2 we have proposed several classes of valid cut set inequalities for the TLNDF. In this section we will show that the separation of some of them can be done on the original graph G, extended by an extra node, while only the more general (x12-z) inequalities need to be separated on the extended graph.

Separation of constraints (x1) and (x12) is performed on the original graph G: we solve a maximum flow problem between the root and each $i \in P$ and $i \in S$, respectively, using the values of $\mathbf{x}^1$ and $\mathbf{x}^1 + \mathbf{x}^2$ as arc capacities. To separate the more general constraints (x12-z), we build the extended graph G_{NS}, set the arc capacities on G' and G'' to $\mathbf{x}^1$ and $\mathbf{x}^2$, respectively, and capacities of arcs $k'k''$ to z_k, for all $k \in F$. To make sure that only non-redundant cuts between the root and a secondary node $i \in S$ are separated, we set the capacities of arcs $i'i$ and $i''i$ to $M > 1$, for each $i \in S$. We will now describe how constraints (x1-z) and (x2-z) can be separated on graphs that are much smaller than the extended graph G_{NS}.

Lemma 6. *Inequalities* (x1-z) *can be separated by solving the maximum flow problem on a graph with* $|V| + 1$ *nodes and* $|A| + |V|$ *arcs.*

Proof. For each $k \in S$ we generate a graph $G_t = (V' \cup \{t\}, A' \cup A_t)$ with weights w_{uv} as follows:

1. V' and A' are defined as in Section 2.2
2. $A_t = \{(i', t) \mid i' \in V'\}$
3. $w_{i'j'} := x^1_{ij}$, $i'j' \in A'$ and $w_{i't} := z_i$, $i \in V \setminus \{k\}$ and $w_{k't} := 1$.

Then each (r', t)-cut in G_t with a weight of less than 1 corresponds to a violated (x1-z) inequality for $k \in W \cap S$. □

Lemma 7. *Inequalities* (x2-z) *can be separated by solving the maximum flow problem on a graph with* $|V| + 1$ *nodes and* $|A| + |V|$ *arcs.*

Proof. For each $k \in S$ we generate a graph $G_s = (V'' \cup \{s\}, A'' \cup A_s)$ with weights w_{uv} as follows:

1. V'' and A'' are defined as in Section 2.2
2. $A_s = \{(s, i'') \mid i'' \in V''\}$

3. $w_{i''j''} := x^2_{ij}$, $i''j'' \in A''$, $w_{si''} := z_i$, $i \in V \setminus \{k\}$ and $w_{sk''} := x^1(\delta^-(k))$.

Then each (s, k'')-cut in G_s with a weight of less than 1 corresponds to a violated (x2-z) inequality for $k \in W \cap S$. $\square$

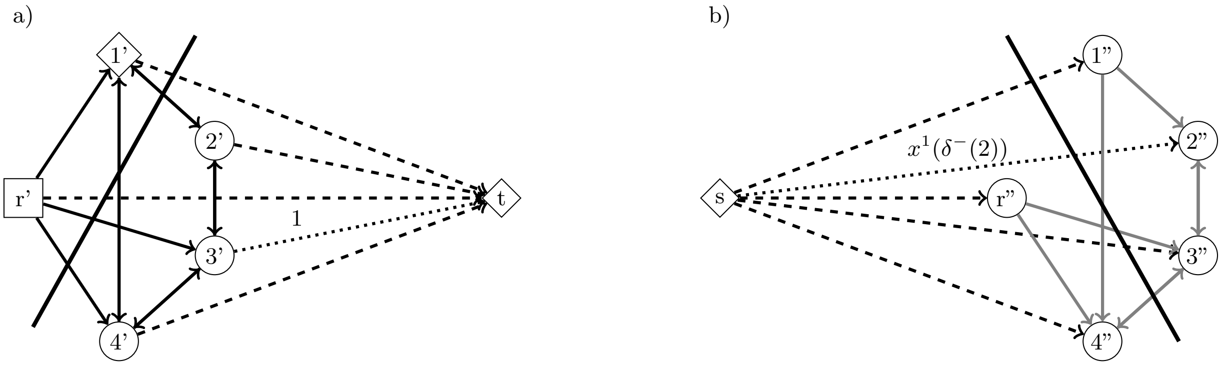


Figure 4: a) Illustration of G_t for $k = 3$ and inequality (x1-z) for $W = \{2, 3, 4\}$. b) Illustration of G_s for $k = 2$ and inequality (x2-z) for $W = \{1, 3\}$.

We separate violated cut set inequalities in every node of the the branch-and-bound tree (B&B). To improve the computational efficiency of the separation, for each family of cuts we search for nested minimum cardinality cuts. To do so, all capacities in the respective separation graph are increased by some $\epsilon > 0$. Thus, every detected violated cut contains the least possible number of arcs. We resolve the linear program after adding at most 50 violated inequalities of any class. Finally, we randomly choose the target nodes to search for violated cuts. To ensure comparability, we fix the seed value for the computational results reported.

As mentioned in Section 2, we propose a two-phase approach to separate the cut set inequalities. In the first phase we add violated inequalities that can be separated on G or on G extended by a single node. Only when no more violated inequalities of these types can be found, we resort to the separation on the extended graph.

The constraint sets (x1-z), (x12) and (x2-z) can be ordered with respect to pairwise inclusion of the corresponding sets W' and W'': Consider an inequality (x1-z). By removing the set $V \setminus W'$ from W'' we obtain an inequality of the form (x12). By removing all but a single node $k \in S$ from W' we obtain an inequality of the form (x2-z). Thus we will always detect violated inequalities in the following order: (x1) - (x1-z) - (x12) - (x2-z) - (x12-z).

In our computational study (see Section 5) we choose different subsets of the mentioned four families for the separation in phase one. The four different strategies we experiment with are given in Table 1.

Strategy EG^+ has the advantage of providing the strong lower bounds of the extended graph model but performs the computationally demanding separation of general cut sets (x12-z) only when it is needed. Strategy EG provides the same lower bounds as EG^+, but it is a more naive implementation of the previous strategy. The whole separation procedure (except for the (x1) cuts) is performed on the extended graph. For the last two strategies, denoted by OG and OG^+, separation is performed on the original graph G or G extended by an extra node, respectively.

	Phase 1				Phase 2
Model	(x1)	(x1-z)	(x12)	(x2-z)	(x12-z)
EG^+	✓	✓	✓	✓	✓
EG	✓				✓
OG^+	✓	✓	✓	✓	
OG	✓		✓		

Table 1: Constraint subsets in Phase 1 and 2

Strategy OG^+ derives slightly weaker lower bounds than EG^+ and EG because we refrain from the separation of the general family of cut sets (x12-z) and insert only four special subfamilies, namely (x1), (x1-z), (x12) and (x2-z). Finally, the weakest lower bounds are obtained using the strategy OG that separates only connectivity cuts for primary and secondary nodes, and uses a compact constraint set to model the transition nodes. Inequalities (2)used in the initialization phase of the branch-and-cut procedure guarantee the feasibility of this model.

5. Computational Study

In this section we report on our computational experience with the four MIP strategies described above. All experiments were performed on a desktop machine with an 8-core Intel Core i7 CPU at 2.80 GHz and 8 GB RAM. Each run was performed on a single processor. We used the CPLEX [17] branch-and-cut framework, version 12.2. All cutting plane and heuristic routines provided by CPLEX are turned off, the other parameters are set to their default values. We set the optimal solution value as global cutoff value in the first and second part of our computations. The primal heuristic was not used in that case.

5.1. Instances

For our computational study we transform instances of the Steiner tree problem (STP) using the following procedure: 30% of STP terminals are chosen as primary customers, the remaining 70% are selected as secondary customers. The primary customer with the lowest index is chosen as root node. The potential Steiner nodes in the STP instance are potential Steiner nodes in the TLNDF instance. We allow installation of a facility in every node of the graph. Primary edge costs equal edge costs of the STP instance. For each secondary edge e, the cost c_e^2 is defined as qc_e^1, where q is uniformly randomly chosen from $[0.25, 0.5]$. Facility opening costs are uniform and equal 0.5 times the average primary edge costs.

The parameters for generating instances have been chosen so that trivial solutions (e.g., optimal solutions that do not contain secondary subtrees) are avoided. In our computational study we also tested the effect of alterations of the above given parameters. We use sets `B`, `C` and `D` of the Steinlib library [18] with 50-100, 500 and 1000 nodes and up to 200, 12500 and 25000 edges, and the sets of random graphs named `K` and `P` proposed by Minkoff and Karger [20], with a street-like structure and up to 400 nodes and 1576 edges. The latter instances are available online at [23].

5.2. Results

Preliminary tests showed that all instances from groups `B`, `K100` and `P100` can be solved in less than two seconds by all of the four tested approaches. Of the instances in these three groups only b11 and b15 were not solved to optimality at the root node of the branch-and-bound tree, but required to examine two branch-and-bound nodes each.

To avoid possibly misleading conclusions from large relative but small absolute deviations in the running time, we do not consider these three instance groups in the following.

5.2.1. Comparing lower bounds and running times

We perform the first part of our computational study on the set of 18 instances with 200 and 400 nodes from test sets K and P. Our goal was to test whether the theoretical results presented in Section 3 are supported by computational experience. Since among the four MIP approaches presented above, EG and EG^+ provide the same LP bounds v_{LP} and differ only in the separation strategy, we performed this test using only three out of the four approaches, namely, EG, OG^+ and OG. We set the default time limit to 10 minutes. For each of these approaches, and for each of the 18 instances, Table 2 reports the following values: the optimal integer solution value (OPT), the running time (in seconds) needed to solve the LP relaxation (t_{LP} $[s]$), the LP gap (calculated as *Gap* [%]=$(OPT - v_{LP})/OPT$), the running time (in seconds) of the integer program (t_{IP} $[s]$), and the number of enumerated branch-and-bound nodes (#*BnB*). A dash denotes that the respective model could not solve the instance within the given time limit. In that case, column #*BnB* indicates the number of nodes enumerated until then.

		t_{LP} $[s]$			*Gap* [%]			t_{IP} $[s]$			#*BnB*		
Instance	OPT	EG	OG^+	OG	EG	OG^+	OG	EG	OG^+	OG	EG	OG^+	OG
K200	385.9	4.3	5.3	**2.9**	0.00	0.05	2.63	**4.6**	5.6	355.9	0	1	6832
K400	383.5	36.7	38.9	**22.5**	0.00	0.37	4.07	**38.3**	47.4	-	0	7	1033
K400-1	474.2	50.1	48.2	**33.5**	0.02	0.34	2.46	**51.8**	58.9	-	0	8	1151
K400-2	456.5	46.3	41.0	**26.8**	0.00	0.04	2.91	48.2	**45.1**	-	0	2	1007
K400-3	431.5	43.5	42.2	**28.8**	0.00	0.05	2.62	45.3	**43.2**	-	0	1	1075
K400-4	394.4	45.3	40.9	**21.8**	0.03	0.05	2.20	47.0	**42.9**	-	1	2	1070
K400-5	539.0	147.3	112.7	**98.5**	0.00	0.07	2.57	149.0	**123.2**	-	0	5	495
K400-6	449.5	84.9	82.1	**56.2**	0.13	0.16	2.25	97.0	**89.9**	-	5	5	755
K400-7	468.1	57.5	64.4	**29.7**	0.00	0.49	3.68	**59.5**	206.9	-	0	172	998
K400-8	459.4	**70.1**	75.5	73.0	0.00	0.00	1.61	**71.5**	76.6	-	0	0	879
K400-9	480.5	104.6	95.0	**75.2**	0.23	0.33	3.29	**128.4**	153.7	-	13	43	780
K400-10	330.7	20.3	21.2	**12.8**	0.00	0.12	2.29	**21.3**	22.9	-	0	3	1888
P200	1051.7	5.5	3.9	**2.8**	0.00	0.00	0.04	5.9	4.2	**2.9**	0	0	1
P400	2085.7	35.8	34.8	**11.7**	0.11	0.11	0.63	42.6	38.4	**30.3**	3	3	8
P400-1	2183.8	32.5	26.4	**24.3**	0.00	0.00	0.34	35.2	**28.1**	118.1	0	0	38
P400-2	2239.2	20.2	18.6	**9.3**	0.00	0.00	0.20	22.5	20.1	**9.7**	0	0	0
P400-3	2636.9	40.1	35.6	**22.2**	0.05	0.05	0.19	42.6	**37.3**	37.6	1	1	6
P400-4	2104.8	24.6	19.4	**18.2**	0.00	0.00	0.10	26.4	20.6	**18.5**	0	0	1

Table 2: Running times of IP and LP, LP gaps and number of enumerated branch-and-bound nodes for three approaches of different strength.

The values in Table 2 confirm the relations stated in Section 3. While the strongest model, EG, solves to optimality a majority of the instances already at the root node of the branch-and-bound tree, the LP relaxations of the two weaker models provide weaker lower bounds (see column *Gap* [%]) and thus require to enumerate a significantly larger number of nodes before possibly reaching optimality. The running times of the LP relaxations show that separating violated inequalities from the "small" constraint sets (x1) and (x12) (model OG) can indeed be accomplished faster than separating inequalities from a larger subset including (x1-z) and (x2-z) (model OG^+) or even the complete set (x12-z) (model EG). When comparing the running times for the complete integer programs we see that there is a tradeoff between faster separation and stronger bounds. Whenever there is a significant difference in the enumerated number of branch-and-bound nodes, the stronger model *EG* using the slower separation is faster overall.

In the second part of our computational study we assess whether the cut separation on graph G in the first phase (EG^+) speeds up the overall performance of the model containing all constraints in (x12-z) (EG). Since the approach OG has shown to be computationally inferior (cf. Table 2), this study compares only strategies EG^+, EG and OG^+. We group the instances in sets `C` and `D` according to the number of customers (first two blocks) and the number of edges in the graph (last two blocks). For each of these groups we calculate the geometric mean of the running time (t_{IP} $[s]$) and the number of cuts added (*#Cuts*). To take into account the values equal to 0 we resort to the shifted geometric mean for presenting the number of enumerated branch-and-bound nodes (*#BnB*) and LP gaps (*Gap* [%]). We use the arithmetic mean as shift.[3]

	t_{IP} $[s]$			*#BnB*			*#Cuts*			*Gap* [%]	
Inst.	EG	EG^+	OG^+	EG	EG^+	OG^+	EG	EG^+	OG^+	EG	OG^+
`c{01,06,11,16}`	3.7	2.8	**2.5**	**1.0**	**1.0**	1.5	133	125	118	0.73	0.84
`c{02,07,12,17}`	7.0	5.5	**4.6**	**0.0**	**0.0**	**0.0**	294	262	245	0.00	0.00
`c{03,08,13,18}`	63.3	62.8	**51.8**	4.1	3.7	**3.5**	2239	2041	1683	0.09	0.11
`c{04,09,14,19}`	65.4	**49.4**	50.4	0.4	**0.1**	**0.1**	2309	2160	1835	0.00	0.00
`c{05,10,15,20}`	**83.3**	99.0	102.6	**1.5**	1.8	2.3	2708	2816	2579	0.04	0.04
`d{01,06,11,16}`	7.6	7.2	**6.6**	**0.1**	**0.1**	1.0	167	163	159	0.00	2.25
`d{02,07,12,17}`	17.9	**13.0**	13.9	0.4	**0.0**	0.6	343	264	287	2.28	4.21
`d{03,08,13,18}`	381.0	**349.1**	362.4	5.1	**3.7**	6.9	4800	4880	4036	0.09	0.11
`d{04,09,14,19}`	**311.5**	332.7	312.1	**0.4**	0.5	**0.4**	4948	4977	4053	0.01	0.01
`d{05,10,15,20}`	879.3	**842.5**	925.3	14.0	**11.9**	13.5	8231	7443	7282	0.09	0.10
`c{01-05}`	4.8	4.5	**3.8**	**0.1**	**0.1**	**0.1**	737	738	632	0.04	0.07
`c{06-10}`	8.4	7.6	**6.8**	**0.4**	**0.4**	**0.4**	786	742	666	0.00	0.00
`c{11-15}`	36.8	30.0	**28.6**	1.6	1.6	**1.1**	935	780	730	0.04	0.04
`c{16-20}`	243.4	214.0	**211.5**	3.3	**2.8**	4.5	1136	1140	1004	0.51	0.54
`d{01-05}`	21.4	**17.3**	18.0	**0.5**	**0.5**	**0.5**	1487	1318	1243	0.01	0.01
`d{06-10}`	49.2	**45.5**	45.9	1.1	**0.7**	**0.7**	1863	1697	1569	1.62	1.69
`d{11-15}`	122.5	124.1	**99.9**	2.5	**1.3**	1.5	1363	1387	1168	0.02	0.02
`d{16-20}`	1032.0	**957.6**	1179.7	10.7	**8.9**	17.1	1830	1666	1700	0.10	3.23

Table 3: Comparison of the 3 selected approaches. The best running times and least number of enumerated branch-and-bound nodes are shown in bold.

The running times indicate that for most groups the separation on the smaller graph G is beneficial. While EG^+ is faster than EG on 15 groups, it's the other way around for only 3 groups. The performance of approach OG^+ is surprisingly good. Even though the gaps are slightly larger, omitting the costly separation on G_{NS} leads to better overall running times, especially on the instances of set `C` with only few customers (`c{1,6,11,16}`, `c{2,7,12,17}`). However, for the instance group with the largest sets of edges and customers (`d{16-20}`) the significantly larger gap requires the enumeration of a lot more branch-and-bound nodes and leads to a performance worse than the one of the stronger models EG and EG^+. We believe that approach OG^+ is suitable for small to medium instances (with a small number of customers), but does not scale well to dense

[3] For non-negative values $v_i, i \in \{1, \dots, k\}$ the *shifted geometric mean* for shift $s > 0$ is defined as $\mu_s(v_1, \dots, v_k) = (\prod_{i=1}^{k}(v_i + s)^{1/k}) - s$.

graphs with large customer sets.

The average values for the number of enumerated branch-and-bound nodes are unexpectedly uncorrelated with the strength of the underlying lower bounds. For some instance groups the "weaker" approach OG^+ enumerated less nodes before reaching optimality, even though it provides the weaker LP bounds. This can be explained by the fact that we use CPLEX' default branching strategy. Thus the variables selected for branching and the order in which the nodes are explored might differ.

An interesting aspect of the results in Table 3 are the number of constraints detected by the separation procedures. In approach OG^+ the least number of cuts is added to the LP and optimality is enforced by extensive branching. This is as expected, as in OG^+ inequalities of the general set (x12-z) are not separated. The number of cuts added by EG and EG^+ indicate another advantage of the latter approach. The inequalities that can be separated on the smaller graphs are more likely to be binding in the optimal LP solution of each branch-and-bound node than some of those detected on the extended graph. Thus, the linear programs in the branch-and-bound tree need less memory, allowing EG^+ better scalability when system memory becomes the limiting factor.

5.2.2. Influence of instance size on algorithmic performance

To assess the influence of the instance size on the running time we compare the running times of approach EG^+ on all instances in testsets `C` and `D` in Tables 4 and 5, respectively. We conclude that a larger number of nodes and edges in G as well as a larger number of customers lead to longer running times. This is not surprising as our approach spends most of the time solving linear programs or separating violated constraints, i.e. calculating maximum flows. The only instance that is not solved within one hour of running time is `d20` with 1000 nodes, 500 customers and 25000 edges. We conclude that for instances with a low density our approach scales well.

	$\lvert E\rvert$			
$\lvert P \cup S\rvert$	625	1000	2500	12500
5	0.9	0.6	2.5	46.6
10	1.4	2.1	6.5	48.4
83	9.2	16.5	207.8	494.3
125	10.2	25.4	53.3	431.1
250	15.5	49.5	134.3	933.2

Table 4: Average running times (in seconds) of approach EG^+ for the instances in testset `C` grouped by the number of edges and the number of customers

	$\lvert E\rvert$			
$\lvert P \cup S\rvert$	1250	2000	5000	25000
5	1.0	6.5	5.7	73.8
10	5.5	2.0	34.4	76.1
167	58.9	220.6	436.8	2616.6
250	49.4	195.7	600.4	2110.3
500	97.2	349.2	571.7	25963.8

Table 5: Average running times (in seconds) of approach EG^+ for the instances in testset `D` grouped by the number of edges and the number of customers

5.2.3. Influence of instance parameters on algorithmic performance

In the last part of our study we assess the effect of changes of the parameters used to generate the test instances. We compare the performance of EG^+ on instance sets generated using the default parameters to the performance when these parameters are altered.

The following choices of parameters were tested:

- Higher and lower facility opening costs (0.75 and 0.25 times the average primary edge costs), indicated by $d \uparrow$ and $d \downarrow$, respectively.

- Higher and lower secondary edge costs ($q \in [0.5, 1)$ and $q \in [0.125, 0.25]$), indicated by $c^2 \uparrow$ and $c^2 \downarrow$, respectively.
- Higher/lower facility opening costs combined with lower/higher secondary edge costs.
- Restricting potential facilities to the customer nodes $R = P \cup S$ (cf. [11]).

In Table 6 we report results aggregated over the 18 instances listed in Table 2. We report first, second and third quartile (Q_1, Q_2 and Q_3) of the running times (t_{IP} [s]) and number of cuts added (#*Cuts*), the number of instances solved to optimality at the root node of the branch-and-bound tree ($\#Opt_{LP}$) and the shifted geometric mean of the LP gaps (*Gap* [%], where we once again use the arithmetic mean as shift). For the default setting indicated by EG^+ we report absolute numbers. For all other parameter settings we report absolute numbers for $\#Opt_{LP}$ and *Gap* [%]. For the runtime t_{IP} [s] and the number of added cuts #*Cuts* we report the increase or decrease compared to the respective value for EG^+ in per cent.

	t_{IP} [s]			#*Cuts*				
Setting	Q_1	Q_2	Q_3	Q_1	Q_2	Q_3	$\#Opt_{LP}$	*Gap* [%]
EG^+	23.8	43.4	63.0	1817	2215	2897	13	0.04
$d \uparrow$	0.0	34.2	29.2	3.0	16.1	12.0	11	0.06
$d \downarrow$	-3.4	-31.0	-28.7	-14.0	-12.9	-12.0	16	0.01
$c^2 \uparrow$	31.9	-2.5	6.3	11.0	6.7	13.0	11	0.14
$c^2 \downarrow$	2.9	17.5	27.6	-8.0	10.9	1.0	7	0.08
$d \uparrow c^2 \downarrow$	24.8	71.1	70.3	-5.0	35.3	14.0	5	0.12
$d \downarrow c^2 \uparrow$	33.2	-11.2	-12.9	-2.0	-3.6	-5.0	13	0.02
$F = R$	-55.5	-34.8	-44.1	-58.0	-38.2	-42.0	12	0.05

Table 6: The influence of variations from the initial parameter settings for generating TLNDF instances.

We observe that increasing facility opening costs and lowering secondary edge costs leads to a significant increase of the running times. The effects add up when the two deviations are combined. Reducing the facility opening costs has the opposite effect of increasing facility opening costs and reduces the running times. Higher secondary edge costs do not show a similar opposite effect of lowering secondary edge costs.

Reducing the number of possible facility locations reduces the problem complexity and leads to shorter running times. This is not surprising as the variable space and the separation graphs for subsets of constraints involving variables $\mathbf{z}$ are much smaller in this case.

For all parameter settings (except for combined high facility opening and low secondary edge costs), the first, second and third quartile of the running time never increases by more than 35% compared to our default setting. We conclude that the cost structure of the instance has only little influence on the overall performance of our model.

The key values other than the running times reported in Table 6 confirm that increased solution time comes along with more detected violated inequalities, a larger LP gap and less instances for which the LP relaxation of our model provides the optimal integer solution.

To see the effect of different cost parameters on the solution structure consider Figure 5. Figure 5(a) shows the optimal solution for the default setting (which is the same as the optimal solution for $d \uparrow$, $c^2 \downarrow$ and $F = R$). Figure 5(b) shows the optimal solution for $d \uparrow c^2 \downarrow$. Even though the facility on the right hand side becomes more expensive in $d\uparrow$ the solution does not change. Lowering secondary edge costs makes it profitable to change the path to the secondary customer at the bottom and to open one facility less. Figure 5(c) shows the optimal solution for $c^2\uparrow$ and $d\downarrow c^2\uparrow$. It illustrates that less difference between primary and secondary edge costs will reduce the size and number of secondary customers in the secondary subforest. Finally, Figure 5(d) shows the optimal solution for $d\downarrow$. Lowering facility opening costs leads to an additional open facility and increases the number of subtrees in the secondary forest.

6. Conclusions

We introduced a new combinatorial optimization problem combining facility location and network design decisions. We considered several mixed integer programming formulations for the problem. Besides formulations derived on the space of original design variables, we also provided three extended formulations: two of them use a flow and a disaggregated flow concept, respectively, and the third one uses a reformulation of the problem on an extended graph in which facility nodes are modeled as arcs.

We provided a theoretical comparison of those models, with respect to the strength of their LP bounds, and in particular showed that the LP bound of the new model based on flows "disaggregated by technology" equals the LP bound of the cut set model on the extended graph. The extensive computational study compares and shows the applicability of the cutting-plane-based counterparts of these models.

Further interesting topics on TLNDF that have not been covered by this article include characterizations of facets of the TLNDF polytope, development of approximation algorithms and/or efficient (meta)heuristics. Furthermore, TLNDF can be extended for modeling networks in several stages. Multi-period or two-stage stochastic or recoverable robust approaches are natural extensions of this problem of great relevance in practice.

Bibliography

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

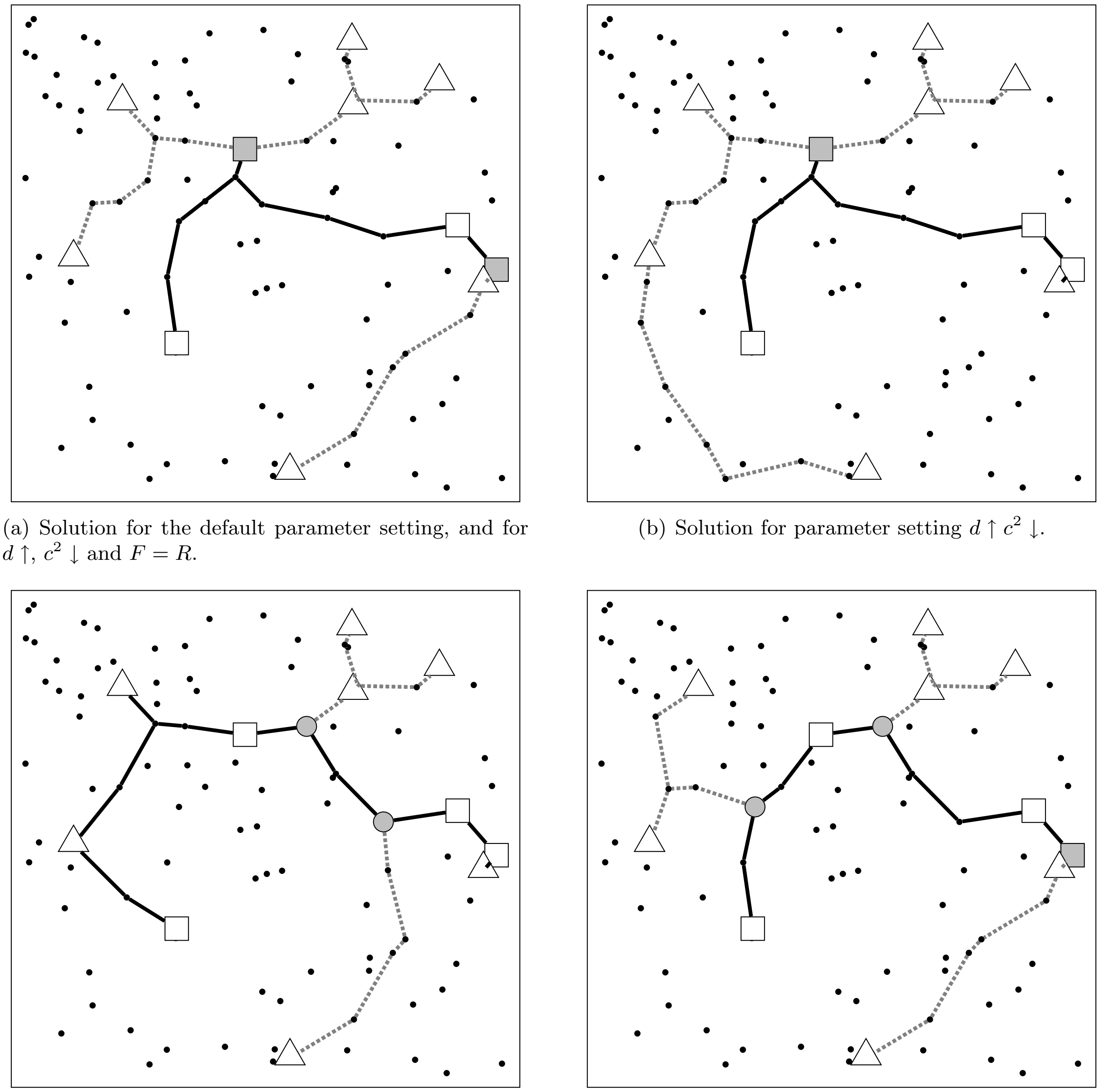
[2] Balakrishnan, A., Magnanti, T. L., and Mirchandani, P. (1994b). Modeling and heuristic worst-case performance analysis of the two-level network design problem. *Management Science*, 40(7):846–867.

[3] Chopra, S., Gorres, E. R., and Rao, M. R. (1992). Solving the Steiner Tree Problem on a Graph Using Branch and Cut. *INFORMS Journal on Computing*, 4(3):320–335.

[4] Chopra, S. and Tsai, C.-Y. (2002). A branch-and-cut approach for minimum cost multi-level network design. *Discrete Mathematics*, 242(1–3):65–92.

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

[6] Current, J. R. (1988). Design of a hierarchical transportation network with transshipment facilities. *Transportation Science*, 22(4):270–277.



(a) Solution for the default parameter setting, and for $d \uparrow$, $c^2 \downarrow$ and $F = R$.

(b) Solution for parameter setting $d \uparrow c^2 \downarrow$.

(c) Solution for parameter settings $c^2 \uparrow$ and $d \downarrow c^2 \uparrow$.

(d) Solution for parameter setting $d \downarrow$.

Figure 5: Optimal solution of instance `K100-1` for different parameter settings. Solid (dotted grey) lines indicate primary (secondary) edges, a grey fill indicates nodes where a facility is installed.

[7] Current, J. R. and Pirkul, H. (1991). The hierarchical network design problem with transshipment facilities. *European Journal of Operational Research*, 52(3):338–347.

[8] Current, J. R., ReVelle, C. S., and Cohon, J. L. (1986). The hierarchical network design problem. *European Journal of Operational Research*, 27(1):57–66.

[9] Duin, C. and Volgenant, A. (1989). Reducing the hierarchical network design problem. *European Journal of Operational Research*, 39(3):332–344.

[10] Duin, C. and Volgenant, A. (1991). The multi-weighted Steiner tree problem. *Annals of Operations Research*, 33(6):451–469.

[11] Gollowitzer, S., Gouveia, L., and Ljubić, I. (2011). A node splitting technique for two level network design problems with transition nodes. In Pahl, J., Reiners, T., and Voß, S., editors, *Network Optimization - 5th International Conference, INOC 2011, Hamburg, Germany, June 13-16, 2011. Proceedings*, volume 6701 of *Lecture Notes in Computer Science*, pages 57–70. Springer.

[12] Gollowitzer, S. and Ljubić, I. (2011). MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research*, 38(2):435–449.

[13] Gourdin, E., Labbé, M., and Yaman, H. (2001). Telecommunication and location - a survey. In Drezner, Z. and Hamacher, H., editors, *Facility Location: Applications and Theory*. Springer.

[14] Gouveia, L. and Janssen, E. (1998). Designing reliable tree networks with two cable technologies. *European Journal of Operational Research*, 105(3):552–568.

[15] Gouveia, L. and Telhada, J. (2001). An augmented arborescence formulation for the two-level network design problem. *Annals of Operations Research*, 106:47–61.

[16] Gouveia, L. and Telhada, J. (2008). The multi-weighted Steiner tree problem: A reformulation by intersection. *Computers & Operations Research*, 35(11):3599–3611.

[17] IBM (November 22nd 2011). CPLEX. `http://www.ilog.com/products/cplex/`;.

[18] Koch, T., Martin, A., and Voß, S. (2000). SteinLib: An updated library on Steiner tree problems in graphs. Technical report, Konrad-Zuse-Zentrum für Informationstechnik Berlin.

[19] Magnanti, T. and Wolsey, L. (1995). Optimal trees. *Handbook in Operations Research and Management Science*, pages 503–615.

[20] Minkoff, M. and Karger, D. R. (1998). The prize collecting Steiner tree problem. In *In Proceedings of the 11th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 760–769.

[21] Obreque, C., Donoso, M., Gutiérrez, G., and Marianov, V. (2010). A branch and cut algorithm for the hierarchical network design problem. *European Journal of Operational Research*, 200(1):28–35.

[22] Pirkul, H., Current, J., and Nagarajan, V. (1991). The hierarchical network design problem: A new formulation and solution procedures. *Transportation Science*, 25(3):175–182.

[23] Resende, M. G. C. (November 22nd 2011). Personal website. `http://www2.research.att.com/~mgcr/data/index.html`;.

[24] Sancho, N. (1995). A suboptimal solution to a hierarchial network design problem using dynamic programming. *European Journal of Operational Research*, 83(1):237–244.

Capacitated Network Design with Facility Location

Stefan Gollowitzer[a], Bernard Gendron[b], Ivana Ljubić[a]

[a] *Department of Statistics and Operations Research, Faculty of Business, Economics, and Statistics, University of Vienna, Austria*

[b] *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), and Department of Computer Science and Operations Research, Université de Montréal C.P. 6128, Succursale Centre-Ville, Montréal, Canada H3C 3J7*

Abstract

We consider a network design problem that arises in the design of last mile telecommunication networks. It combines the capacitated network design problem (CNDP) with the single-source capacitated facility location problem (SSCFLP). We will refer to it as the Capacitated Connected Facility Location Problem (CapConFL). We develop a basic integer programming model based on single-commodity flows. Based on valid inequalities for the subproblems, CNDP and SSCFLP, we derive several (new) classes of valid inequalities for the CapConFL. We use them in a branch-and-cut framework and show their applicability on a set of real-world instances.

Keywords: Capacitated Network Design, Facility Location, Connected Facility Location, Mixed Integer Programming Models, Telecommunications

---

1. Introduction

Given a set of customers, a set of potential facility locations and some inter-connection nodes, the goal of the *Connected Facility Location* problem (ConFL) is to find the minimum-cost way of assigning each customer to exactly one open facility, and connecting the open facilities via a Steiner tree. The sum of costs for the Steiner tree, the facility opening costs and the assignment costs needs to be minimized. This problem has been used to model a network design problem that arises in the design of last mile telecommunication networks when the *fiber to the curb* (FTTC) deployment strategy is applied (see, e.g., [17]). Contrary to the *fiber to the home* strategy, where each customer, i.e., household, has its own fiber-optic uplink, in the FTTC strategy some of the existing copper wire infrastructure is used. More precisely, in an FTTC network, fiber optic cables run from a central office to a cabinet serving a neighborhood. End users connect to this cabinet using the existing copper connections. Expensive switching devices are installed in these cabinets. The usage of the last d meters of copper wire between the customer and a switching device may significantly reduce deployment costs while still enabling broadband connections of reasonable quality.

In more detailed planning of FTTC networks, capacities of the links and of multiplexor devices are limited and this aspect was not captured by the ConFL variants studied in the literature so

Email addresses: `stefan.gollowitzer@univie.ac.at` (Stefan Gollowitzer), `gendron@iro.umontreal.ca` (Bernard Gendron), `ivana.ljubic@univie.ac.at` (Ivana Ljubić)

far. In this paper we consider a new capacitated variant of the ConFL problem, that we will refer to as the *Capacitated Connected Facility Location Problem* (CapConFL).

In a typical application from telecommunications, demands of customers are given as the number of twisted copper lines that are to be "served" at the respective customer location. Switching (or multiplexor) devices have both capacity and demand. Capacity is defined in terms of the number of twisted copper lines a device can serve. The demand of a switching device is defined as the number of fiber-optic uplinks required to connect the device to the central office (which is further connected to the backbone network). The number of uplinks is fixed for each device and independent of the number of customers that are finally assigned to it. The CapConFL consists of deciding on the location of switching devices, the assignment of customers to these devices and the routing of the uplinks from the switching devices to the central office, while minimizing the overall investment costs.

1.1. Problem definition

More formally, CapConFL can be defined as follows. The input is a graph $G = (V, E_S \cup A_R)$ with the set of nodes V partitioned into the set of *customers* (R), the set of *potential facility locations* (F) and the set of *potential Steiner nodes* ($V \setminus (F \cup R)$). A root node $r \in V \setminus (F \cup R)$ represents the connection to a higher order (e.g., backbone) network. The network $G_S = (V_S, E_S)$, where $V_S := V \setminus R$ and $E_S := \{e = \{i,j\} \in E \mid i,j \in V_S\}$ is called the *core network*. The *assignment network* $G_R = (F \cup R, A_R)$ consists of directed arcs between potential facilities and customers, i.e., $A_R = \{(i,k) \mid i \in F, k \in R\}$. The following input parameters are associated to the network:

- Facility opening cost $f_i \geq 0$, capacity $v_i > 0$ and demand $d_i > 0$ for each $i \in F$.
- Arc cost $c_e \geq 0$ and capacity $u_e > 0$ for each $e \in E_S$.
- Assignment cost $c_{ij} \geq 0$ for each $(i,j) \in A_R$.
- Customer demand $b_k > 0$ for each $k \in R$.

The goal is to find a subnetwork of G consisting of the set of open facilities F', the set of core edges E'_S and the set of assignment arcs A'_R such that:

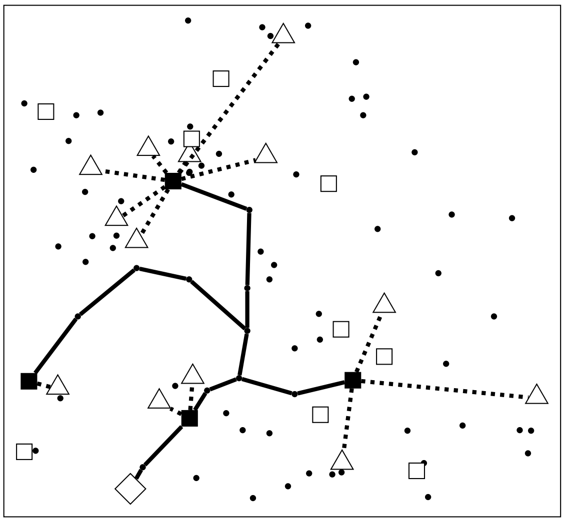
(P1) Each customer is assigned to exactly one open facility using arcs from A'_R.

(P2) The sum of customers' demands assigned to a facility i does not exceed its capacity v_i.

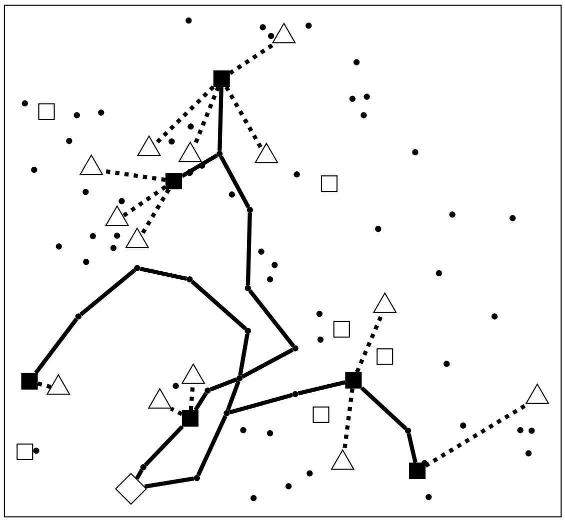
(P3) In the core subnetwork induced by E'_S, we can simultaneously route the flow from the root node to satisfy the demand of all open facilities, without violating the edge capacities.

(P4) The sum of assignment, facility opening and edge costs, given by $\sum_{e \in E'_S} c_e + \sum_{i \in F'} f_i + \sum_{(i,j) \in A'_R} c_{ij}$, is minimized.

Obviously, by setting capacities $u_e = \infty$, for all $e \in E_S$ and $v_i = \infty$, for all $i \in F$, we obtain the previously studied ConFL problem. Figure 1 illustrates solutions for ConFL and CapConFL. Squares and triangles denote facilities and customers, respectively. A black fill indicates that a facility is open. A diamond denotes the root node. Solid edges are in the core network, dotted edges represent the assignments. In the CapConFL the limited facility capacities require two



(a) ConFL solution



(b) CapConFL solution

Figure 1: Feasible solutions of the ConFL and CapConFL problem, respectively.

additional open facilities and a different assignment of customers to facilities. The limited edge capacities require additional edges in the core network.

Notice that ConFL combines the Steiner tree problem and the uncapacitated facility location problem. On the other hand, CapConFL combines the capacitated network design problem with the single-source capacitated facility location problem. To see this, consider a feasible CapConFL instance whose core graph has a star topology. One can easily transform this input graph into an instance of the single-source capacitated facility location problem: the facility opening costs for each $i \in F$ are now defined as $c_e + f_i$ where c_e is the corresponding adjacent edge, and the assignment graph remains unchanged. Similarly, a feasible CapConFL instance in which the assignment arcs are such that each customer is adjacent to exactly one facility can be reduced into an instance of the single-source capacitated network design problem.

1.2. Literature review

Since CapConFL has not been considered before, we provide a detailed literature overview of three closely related problems: connected facility location, capacitated network design and single-source capacitated facility location.

Connected Facility Location. Early work on ConFL mainly includes approximation algorithms. ConFL can be approximated within a constant ratio and the currently best-known approximation ratio is provided by Eisenbrand et al. [13]. Recently, heuristic approaches have been proposed by Ljubić [25] and Bardossy and Raghavan [3]. Gollowitzer and Ljubić [17] present and compare several formulations for ConFL, both theoretically and computationally. Some of these results will be discussed and related to the CapConFL later on. Arulselvan et al. [2] consider a time-dependent variant of the ConFL and present a branch-and-cut approach based on cover, cut set cover and degree balance inequalities. Leitner and Raidl [24] propose a branch-and-cut-and-price approach for a variant of ConFL with *capacities on facilities.* Cutting planes are used to ensure paths between the root and open facilities, while column generation is used for selecting open facilities and assigning customers to them.

(Single-Source) Capacitated Network Design Problems (CNDP). In a typical CNDP setting, a network is given with a limited capacity available on each edge. A subset of edges of minimum cost needs to be installed in the network such that commodities with multiple origins and multiple destinations can be routed through the network without violating installed edge capacities. There exists a large body of work on the CNDP and related problems.

It includes exact methods based on Lagrangian relaxation or decomposition [14, 19, 9, 23], heuristic methods based on tabu search, neighbourhood search, slope scaling and Lagrangian relaxation [8, 15, 16, 10]. Recent developments comprise a theoretical study and comparison of Benders, metric and cut set inequalities [7] and a hybrid method combining mathematical programming and neighbourhood search techniques [18]. Finally, Chouman et al. [5] present a branch-and-cut approach that compares several families of valid inequalities for the CNDP.

A generalization of the single-source CNDP is the *Local Access Network Design* problem (LAN). In this problem multiple copies of each edge are available. The Local Access Network Design problem was studied by Raghavan and Stanojević [27], Salman et al. [28] and Ljubić et al. [26].

The Single-Source Capacitated Facility Location Problem (SSCFLP). Aardal et al. [1] and Deng and Simchi-Levi [11] proposed MIP models and studied the corresponding polyhedra of the SSCFLP and related problems. Holmberg et al. [20] present a branch-and-bound method based on a Lagrangean heuristic, Diaz and Fernández [12] develop a branch-and-price approach based on a decomposition of the SSCFLP and Contreras and Díaz [6] propose a scatter search heuristic. Ceselli et al. [4] give an exhaustive computational evaluation of branch-and-cut and branch-and-price approaches for a general class of facility location problems that includes the SSCFLP.

1.3. Contribution and outline

In Section 2 we introduce a basic integer programming model for CapConFL and discuss the relation of CapConFL and the Connected Facility Location problem. In particular, we show that a domination result between two sets of valid inequalities for ConFL does not hold for CapConFL. In Section 3 we derive cover and extended cover inequalities for the various knapsack type constraints in our model. In addition, we provide two generalizations of recently proposed cut-set-cover inequalities and cover inequalities for single cut sets. Separation procedures for these valid inequalities are discussed in Section 4. In Section 5 we illustrate the effectiveness of the proposed model and the valid inequalities by computational experiments on a set of new, realistic benchmark instances based on real data. We conclude the paper in Section 6.

2. Mixed integer programming models

In this section we introduce a first basic model for the CapConFL. It is based on models familiar in the context of the SSCFLP and the CNDP. We then strengthen this model using concepts known from the Connected Facility Location problem [17].

Since all demands of open facilities have to be routed from a single source node, it can be shown (see, e.g., [26]) that without loss of generality we can replace the undirected core network G_S by a bidirected graph in which each edge $e \in E_S$ is replaced by two directed arcs, except for the edges adjacent to the root node, where it is sufficient to consider outgoing arcs from r. The set of arcs of the bidirected core network will be denoted by A_S. Since the flow routed through an edge will always be routed in one of the two opposite directions, we define cost and capacities as $c_{ij} = c_e$ and $u_{ij} = u_e$, respectively, for each $e = \{i, j\}$ in E_S. The union of core and assignment arcs is denoted by $A = A_S \cup A_R$. For a set of customers $J \subset R$ we denote the set of facilities that can serve these customers by $F(J) = \bigcup_{k \in J} F(k)$, where $F(k) := \{i \in F : (i, k) \in A_R\}$. Likewise, for $I \subset F$ we denote by $R(I) = \bigcup_{i \in I} R(i)$ where $R(i) := \{k \in R : (i, k) \in A_R\}$. For $W \subset V$ we denote the set of ingoing arcs by $\delta^-(W)$.

2.1. The basic MIP model

In our models we will use the following binary decision variables:

$$x_{ij} = \begin{cases} 1, & \text{if arc } (i,j) \text{ is installed} \\ 0, & \text{else} \end{cases} \qquad (i,j) \in A$$

$$z_i = \begin{cases} 1, & \text{if facility } i \text{ is installed} \\ 0, & \text{else} \end{cases} \qquad i \in F$$

In addition, continuous flow variables g_{ij} indicate the total amount of the flow between the root r and all open facilities in F routed through the arc $(i, j) \in A$.

The following model combines the single-commodity flow (SCF) formulation for the CNDP (see, e.g., [27, 28]) with a formulation for the SSCFLP (see, e.g., [20]):

$$(SCF) \quad \min \sum_{ij \in A} c_{ij} x_{ij} + \sum_{i \in F} f_i z_i$$

$$\text{s.t.} \quad \sum_{ji \in A_S} g_{ji} - \sum_{ij \in A_S} g_{ij} = \begin{cases} d_l z_l & i = l \\ -\sum_{l \in F} d_l z_l & i = r \\ 0 & \text{else} \end{cases} \quad i \in V_S \tag{1a}$$

$$0 \le g_{ij} \le u_{ij} x_{ij} \qquad (i,j) \in A_S \tag{1b}$$

$$\sum_{k \in R(i)} b_k x_{ik} \le v_i z_i \qquad i \in F \tag{1c}$$

$$x_{ik} \le z_i \qquad i \in F, k \in R(i) \tag{1d}$$

$$\sum_{i \in F(k)} x_{ik} = 1 \qquad k \in R \tag{1e}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{1f}$$

$$z_i \in \{0,1\} \qquad i \in F \tag{1g}$$

Constraints (1c)-(1e) are the strong relaxation of the SSCFLP. The assignment constraints (1e) model the property (P1) and constraints (1c)-(1d) ensure the property (P2). In constraints (1a)-(1b) we use the single-commodity flow variables to ensure the property (P3). This model is intuitive, but it provides weak lower bounds, due to the following facts: 1) big-M constraints (1b) are used to model the arc capacities, and 2) the connectivity between the root and the open facilities, rather than between the root and the customers, is required. The model is impractical to solve in a branch-and-bound framework, even for medium sized instances.

Using the following *capacitated cut set inequalities* we can project out the flow variables from the previous model and replace the constraints (1a)-(1b) by the following ones (see, e.g., Ljubić et al. [26]):

$$\sum_{ij \in \delta^-(W)} u_{ij} x_{ij} \geq \sum_{l \in F \cap W} d_l z_l \qquad W \subseteq V_S \setminus \{r\} \tag{Cut$_{SCF}$}$$

The obtained model contains an exponential number of inequalities and provides the same lower bounds as the corresponding flow model. However, inequalities (Cut$_{SCF}$) can be strengthened as follows:

$$\sum_{ij \in \delta^-(W)} \min(u_{ij}, \sum_{l \in F \cap W} d_l) x_{ij} \geq \sum_{l \in F \cap W} d_l z_l \qquad W \subseteq V_S \setminus \{r\}$$

2.2. Relations to Connected Facility Location and cut set inequalities

In [17] we studied MIP formulations for ConFL and provided a complete hierarchy of several MIP formulations with respect to the quality of their LP-bounds. Among others, we described two cut set-based formulations for ConFL. The models differ in the way they require connectivity.

In the first model, connectivity is ensured between the root and any open facility as follows:

$$\sum_{ij \in \delta^-(W)} x_{ij} \geq z_l \qquad W \subseteq V_S \setminus \{r\},\ l \in W \cap F \tag{Cut$_Z$}$$

These inequalities state that for each open facility the edges on at least one path between the root node and the respective facility need to be installed. Additional assignment constraints (1d) and (1e) are required between the facilities and customers.

The second model replaces constraints (Cut$_Z$) by the following cut set inequalities that ensure connectivity between the root and every customer:

$$\sum_{ij \in \delta^-(W)} x_{ij} \geq 1 \qquad W \subseteq V \setminus \{r\}, W \cap R \neq \emptyset \tag{Cut$_X$}$$

We showed that for ConFL the second model provides theoretically stronger lower bounds, but is computationally outperformed by the first model on the set of benchmark instances considered there.

Both sets of inequalities, (Cut$_Z$) and (Cut$_X$) are also valid for CapConFL. It is interesting to mention that, unlike for the ConFL, for which the inequalities (Cut$_Z$) are implied by the model with (Cut$_X$) constraints, the two families of inequalities can be used complementary to each other for CapConFL:

Lemma 1. *Inequalities* (Cut$_Z$) *and* (Cut$_X$) *both strengthen the LP-relaxation of the basic model (SCF). However, the MIP models (SCF)+*(Cut$_X$) *and (SCF)+*(Cut$_Z$) *are incomparable w.r.t. the quality of their LP-bounds.*

Proof. It is not difficult to see that inequalities (Cut$_Z$) and (Cut$_X$) both strengthen the LP-relaxation of (SCF). To see that (Cut$_Z$) inequalities are not implied by (SCF)+(Cut$_X$), consider the example shown in Figure 2. A vector $(\mathbf{x}, \mathbf{z})$ that satisfies (Cut$_{SCF}$) is $x_{12} = 0.75$, $x_{23} = x_{24} = 0.25$, $z_3 = z_4 = 0.75$, $x_{35} = x_{46} = 0.75$ and $x_{45} = x_{36} = 0.25$. This solution is cut off by the (Cut$_X$) constraints $x_{23}+x_{24} \geq 1$ and $x_{12} \geq 1$. Finally, inequalities (Cut$_Z$) are not redundant for (SCF)+(Cut$_X$)

since they ensure $x_{23} + x_{24} \geq 1.5$ which further strengthens the model.
Conversely, the model (SCF)+(Cut$_Z$) does not imply (Cut$_X$) constraints, which follows from the previous results for ConFL in [17], i.e., a CapConFL instance with sufficiently large capacities on arcs and facilities will have the desired property. □

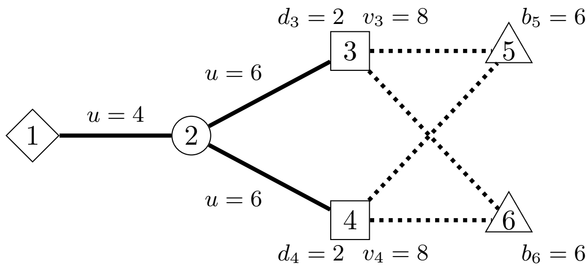


Figure 2: Example for comparison of cut set inequalities

3. Valid Inequalities

For the well-known subproblems of CapConFL, SSCFLP and CNDP, several sets of strengthening valid inequalities are known. We will review ideas that seem relevant in the context of the CapConFL and propose several sets of new valid inequalities based on the combination of the facility location and network design aspect.

3.1. Cover inequalities for single facilities

Deng and Simchi-Levi [11] proposed cover inequalities for the SSCFLP with uniform capacities. These inequalities are better known in the context of general mixed integer programming to strengthen knapsack-type constraints. We will use the concept of *extended cover inequalities* (see, e.g., the recent work of Kaparis and Letchford [22]).

Consider an arbitrary potential facility node $i \in F$. We call a set $R' \subseteq R(i)$ a cover for $i \in F$ if $\sum_{k \in R'} b_k > v_i$ and minimal if $\sum_{k \in R'} b_k - b_\ell \leq v_i$ for all $\ell \in R'$. We call it a minimal cover if it is minimal and a cover. For a minimal cover R', we define $E(R') = \{k \in R(i) \setminus R' : b_k \geq b^*\}$, where $b^* = \max_{k \in R'} b_k$.

Let the set of all minimal covers of $i \in F$ be denoted by $MC(i)$. Then the following *extended knapsack cover* inequalities are valid for the CapConFL:

$$\sum_{j \in R' \cup E(R')} x_{ij} \leq (|R'| - 1) z_i \qquad R' \in MC(i), i \in F \tag{EKS}$$

3.2. Inequalities involving multiple facilities

We derive two new families of inequalities that are implied by the limited capacities of facilities and the limited number of assignments edges in A_R.

Minimum cardinality inequalities on facilities. For a given set of customers $J \subset R$ and the corresponding subset of facilities $F(J)$, let $p(J)$ be the minimum number of facilities in $F(J)$ that is required to assign the customers in J in a feasible way, i.e., by respecting the allowed possible assignments and satisfying the capacity constraints on the facilities in $F(J)$. In other words, $p(J)$ is the optimum solution of a capacitated bin-packing problem with the set of bins $F(J)$, capacities v_i for $i \in F(J)$, the set of items J, demands b_j for $j \in J$ and such that each item $j \in J$ is only allowed to be assigned to bins in $F(j)$. W.l.o.g. we can assume that $b_k \leq v_i$ for all $(i,k) \in A_R$ and thus $p(\{k\}) = 1$ for all $k \in R$ and $p(J) \leq \min\{|F(J)|, |J|\}$ for all $J \subseteq R$.

Then the following *minimum cardinality* inequalities are valid for the CapConFL:

$$\sum_{i \in F(J)} z_i \geq p(J) \qquad J \subseteq R \tag{MC$_F$}$$

(Extended) Cover inequalities on facilities. Next we apply the idea of cover inequalities to the relation of facility capacities and customer demands. Let again $J \subseteq R$. We call a set $F' \subset F(J)$ a *capacity cover* with respect to J if $\sum_{i \in F(J) \setminus F'} v_i < b(J)$ and we call it minimal if $v_k + \sum_{i \in F(J) \setminus F'} v_i \geq b(J)$ for all $k \in F'$. Let $CC(F(J))$ denote the set of all such capacity covers of $F(J)$. We call the following set of constraints *cover inequalities on facilities*:

$$\sum_{i \in F'} z_i \geq 1 \qquad F' \in CC(F(J)), J \subseteq R \tag{2}$$

Similar to the cover inequalities for single facilities we can extend the covers and obtain stronger inequalities. Let $v^* = \max_{i \in F'} v_i$ and let $E(F') = \{i \in F(J) \setminus F' : v_i \geq v^*\}$ be the set of remaining facilities from $F(J)$ with a capacity of at least v^*. We refer to the following inequalities as *extended cover inequalities on facilities*:

$$\sum_{i \in F' \cup E(F')} z_i \geq 1 + |E(F')| \qquad F' \in CC(F(J)), J \subseteq R \tag{Cov$_F$}$$

To see that these inequalities are valid we can rewrite inequalities (2) as $\sum_{i \in F'}(1 - z_i) \leq |F'| - 1$. The corresponding extended cover inequality is then

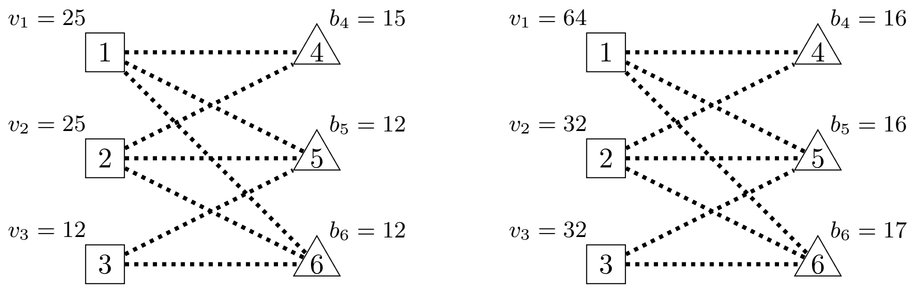
$$\sum_{i \in F' \cup E(F')} (1 - z_i) \leq |F'| - 1.$$

Rewriting this inequality gives (Cov$_F$).

The sets of inequalities (MC$_F$) and (Cov$_F$) do not contain each other as the following counterexamples show. In the example in Figure 3(a) a valid (Cov$_F$) inequality is $z_1 + z_2 \geq 2$, while the (MC$_F$) inequalities only ensure $z_1 + z_2 + z_3 \geq 2$. On the contrary, for the example given in Figure 3(b) the (Cov$_F$) inequalities are $z_1 + z_2 \geq 1$ and $z_1 + z_3 \geq 1$, but they are strictly dominated by the (MC$_F$) inequality $z_1 + z_2 + z_3 \geq 2$ that also implies $z_2 + z_3 \geq 1$.

General representation of cover inequalities on facilities. Consider now a general valid inequality of type

$$\sum_{i \in \hat{F}} z_i \geq p \tag{3}$$



(a) Example 1

(b) Example 2

Figure 3: Counterexamples for comparison of (MC_F) and (Cov_F)

defined for a set $\hat{F} \subseteq F$ and $p \geq 1$. For $p = 1$ we have the simple cover inequalities (2) (i.e., $\hat{F} \in CC(F(J))$) and, for $p \geq 2$, inequalities of type (MC_F) and (Cov_F) belong to this family, i.e., we have $\hat{F} \in F(J) \cup \{F' \cup E(F') \mid F' \in CC(F(J))\}$, for $J \subseteq R$. The following family of *general cover inequalities on facilities* is then also valid for our problem:

$$\sum_{i \in \tilde{F}} z_i \geq 1 \quad \tilde{F} \subseteq F, |\tilde{F} \cap \hat{F}| \geq |\hat{F}| - p + 1 \qquad (\mathrm{Cov}_{gen})$$

It is not difficult to see that the latter inequalities are implied by (3). However, they are of particular interest when combined with cut set inequalities, as explained below.

3.3. Cut-set-cover inequalities

This new family of valid inequalities combines cut set inequalities with the general cover inequalities for facilities of the form (Cov_{gen}). Inequalities (Cov_{gen}) state that at least one facility in $\tilde{F}$ needs to be opened in a feasible solution. Consequently, for every subset of nodes $W \subset V$ containing all nodes in $\tilde{F}$, at least one ingoing arc needs to be installed. Let $\mathcal{F}$ denote the family of all subsets of facilities for which (Cov_{gen}) is valid, i.e.:

$$\mathcal{F} = \bigcup_{J \subseteq R} F(J) \cup \{F' \cup E(F') \mid F' \in CC(F(J))\}$$

and let

$$p(\hat{F}) = \begin{cases} 1 + |E(F')|, & \hat{F} = F' \cup E(F'),\ F' \in CC(F(J)) \\ p(J), & \hat{F} = F(J) \end{cases}$$

for all $\hat{F} \in \mathcal{F}$. The following *cut-set-cover* inequalities are valid for CapConFL and not implied by any of the previously described sets of constraints:

$$\sum_{ij \in \delta^-(W)} x_{ij} \geq 1 \qquad \tilde{F} \subseteq W \cap F, |\tilde{F} \cap \hat{F}| \geq |\hat{F}| - p(\hat{F}) + 1,\ \hat{F} \in \mathcal{F} \qquad (\mathrm{Cut}_{Cov})$$

Inequalities (Cut_{Cov}) are a generalization of the previously introduced cut-set-cover inequalities for the incremental ConFL studied in Arulselvan et al. [2]. Figure 4 illustrates inequalities (Cut_{Cov}) for two different subsets W and a cover inequality $z_1 + z_2 \geq 1$ of type (Cov_F). Figure 5 illustrates inequalities (Cut_{Cov}) for the minimum cardinality inequality $z_1 + z_2 + z_3 \geq 2$.

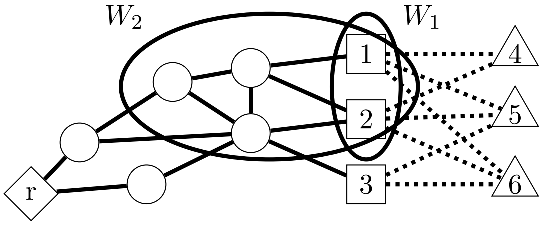


Figure 4: Example for cut-set-cover inequalities (Cut_{Cov}) derived from an inequality (Cov_F).

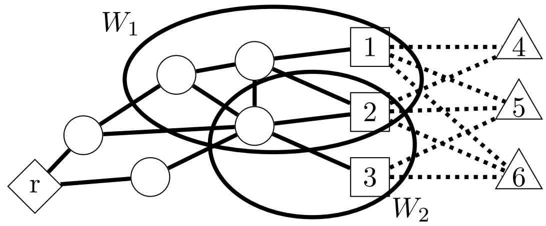


Figure 5: Example for cut-set-cover inequalities (Cut_{Cov}) derived from an inequality (MC_F).

3.4. Cover inequalities for single cut sets

The following set of valid inequalities generalizes the cover inequalities known for the capacitated network design problem studied in Chouman et al. [5]. Consider a (Cut_{SCF}) cut set inequality $\sum_{ij\in\delta^-(W)} u_{ij}x_{ij} \geq \sum_{l\in F\cap W} d_l z_l$ defined by a cut set $\delta^-(W)$ for $W \subseteq V \setminus \{r\}$. Let $F' \subseteq F \cap W$ and $d(F') = \sum_{l\in F'} d_l$. A set $C \subset \delta^-(W)$ is called a *cover* with respect to $\delta^-(W)$ and F', if $\sum_{ij\in\delta^-(W)\setminus C} u_{ij} < d(F')$ and a *minimal cover* if, in addition,

$$\sum_{ij\in\delta^-(W)\setminus C} u_{ij} + u_{lk} \geq d(F') \text{ for all } lk \in C.$$

Let $MC(W, F')$ denote the set of all minimal covers with respect to $\delta^-(W)$ and F'. Then the following *cover inequalities on single cut sets* are valid for the CapConFL:

$$\sum_{ij\in C} x_{ij} \geq 1 + \sum_{l\in F'} (z_l - 1) \qquad \forall C \in MC(W, F'). \qquad (\text{Cov}_{\delta^-(W)})$$

Figure 6 illustrates inequalities ($\text{Cov}_{\delta^-(W)}$). Edge (b, d) is a cover with respect to $W = \{1, 2, 3, d, e, f\}$ and $F' = \{2, 3\}$.

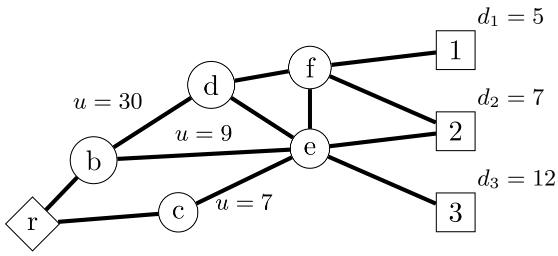


Figure 6: Illustration of cut set cover inequalities

4. Separation procedures

In this section we describe the separation procedures used in our branch-and-cut algorithm. We refer to the variable values of the current fractional solution by $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$.

4.1. Separation of inequalities (Cut_{SCF}) *and* (Cut_Z)

Inequalities (Cut_{SCF}) can be separated in polynomial time (see also Ljubić et al. [26]). We define the support graph $G' = (V', A')$ where $V' := V_S \cup t$ with an additional sink node t, $A' := A_S \cup A_t$ and $A_t := \{(i,t) \mid i \in F, \bar{z}_i > 0\}$. We define capacities on arcs as $u_{ij}\bar{x}_{ij}$ for each arc $ij \in A_S$ and $d_i\bar{z}_i$ for each arc $it \in A_t$. We calculate the minimum cut between r and t in G'. Let $\delta^-(W)$ denote the arcs of this cut. If $\delta^-(W) \cap A_S \neq \emptyset$ and $\sum_{ij\in\delta^-(W)\cap A_S} u_{ij}\bar{x}_{ij} < \sum_{i\in W\cap F} d_i\bar{z}_i$ we have detected a violated inequality (Cut_{SCF}).

Inequalities (Cut_Z) can be separated in similar fashion (see also Gollowitzer and Ljubić [17]). The support graph in this case is the bidirected core network (V_S, A_S) with arc capacities set to $\bar{x}_{ij}$ for each arc $ij \in A_S$. A minimum cut in A_S between r and $l \in F$ with a weight of less than $\bar{z}_l$ corresponds to a violated inequality (Cut_Z).

4.2. Separation of inequalities (Cut_X)

For the separation of (Cut_X) inequalities we define a support graph G_j for each $j \in R$. Thereby, $G_j = (V \cup \{j\}, A_S \cup A_j)$ where $A_j = \{(i,j) \mid i \in F(j)\}$. Capacities on the arcs from $A_S \cup A_j$ are set to $\bar{x}_{ij}$. Each minimum cut in G_j between r and $j \in R$ whose weight is less than 1 corresponds to a violated inequality (Cut_X).

If the number of customers is large, complete separation of inequalities (Cut_X) is very time-consuming. We therefore reduce the set of customers considered in the separation to a subset that still ensures all violated inequalities are identified. A customer $c_1 \in C$ is ignored if there exists another customer $c_2 \in C$ such that $F(c_2) \subset F(c_1)$. If $F(c_i)$ are identical for all $c_i \in \bar{C} \subseteq C$ only one customer in $\bar{C}$ is considered.

4.3. Separation of inequalities (EKS)

For a fractional point $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ and for each $i \in \bar{F} = \{i \in F \mid \bar{z}_i > 0\}$, the separation of (simple, non-extended) inequalities (EKS) is equivalent to solving a knapsack problem which is described by the following integer program:

$$\min\ z = \sum_{j\in R} (\bar{z}_i - \bar{x}_{ij}) s_j$$
$$\text{s.t.} \quad \sum_{j\in R} b_j s_j > v_i$$
$$s_j \in \{0,1\} \qquad \forall j \in R$$

If $z < \bar{z}_i$ a simple, non-extended cover inequality is violated.

Instead of solving the separation problem exactly (e.g., using a dynamic programming procedure), we restrain to a heuristic separation which was proposed by Kaparis and Letchford [22]. Their efficient and fast separation heuristic detects extended knapsack cover inequalities. Adapted to our (EKS) inequalities, this procedure consists of the following steps, executed for each $i \in F$:

1. Sort the items in $R(i)$ in non-decreasing order of $(\bar{z}_i - \bar{x}_{ij})/b_j$, and store them in a list L. Initialize the cover R' as the empty set and initialize $b^* = v_i$.
2. Remove an item from the head of the sorted list L. If its weight is larger than b^*, ignore it, otherwise insert it into R'. If R' is now a cover, go to step 4.
3. If L is empty, stop. Otherwise, return to step 2.

4. If the extended cover inequality corresponding to R' is violated by $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$, output it.
5. Let $k^* = \arg\max_{j \in R'} b_j$ be the customer in R' with the highest demand. Set $b^* = b_{k^*}$ and delete k^* from R'. Return to step 2.

In fact, we perform two variants of this algorithm. The one stated above and one where the customers are sorted in non-increasing order of $\bar{x}_{ij}$.

4.4. Separation of inequalities (MC_F) *and* (Cov_F)

We consider subsets of facilities $F' \in F_C := F_1 \cup F_2$, where $F_1 := \{F(k) \mid k \in R\}$ and $F_2 := \{F(k_1) \cup F(k_2) \mid |F(k_1) \cap F(k_2)| / \min(|F(k_1)|, |F(k_2)|) \geq 0.5, k_1, k_2 \in R\}$, i.e., F_2 contains unions of $F(k_1)$ and $F(k_2)$ such that at least half the facilities of either $F(k_1)$ or $F(k_2)$ are common to both these sets. For each F' we define the subset of customers to be considered in the separation of (MC_F) and (Cov_F) inequalities as $J(F') := \{k' \in R \mid F(k') \subseteq F'\}$.

Separation of inequalities (MC_F). We calculate $p(J)$ for J and $F(J)$ by solving a bin-packing problem with assignment restrictions and non-uniform bin capacities:

$$
\begin{aligned}
p(J) = \min & \sum_{i \in F(J)} t_i & \\
\text{s.t.} & \sum_{k \in R(i)} b_k s_{ik} \leq v_i t_i & i \in F(J) \\
& s_{ik} \leq t_i & k \in J, i \in F(k) \\
& \sum_{i \in F(k)} s_{ik} = 1 & k \in J \\
& s_{ik} \in \{0,1\} & k \in J, i \in F(k) \\
& t_i \in \{0,1\} & i \in F(J)
\end{aligned}
$$

We consider $F' \in F_C$ as candidate sets for $F(J)$ and determine $J = J(F')$ as described in the previous paragraph. The values of $p(J)$ are calculated for all such J during preprocessing. In the separation procedure we repeatedly check whether the current fractional solution violates any of the stored inequalities (MC_F). By doing so we consider at most $|F_C| \leq |R| + |R|^2$ inequalities of type (MC_F).

Separation of inequalities (Cov_F). Given $J \subseteq R$ and $F(J)$, the separation of (simple, non-extended) covers on facilities (2) is equivalent to solving the following knapsack problem:

$$
\begin{aligned}
\min\ z = & \sum_{i \in F(J)} \bar{z}_i s_i & \\
\text{s.t.} & \sum_{i \in F(J)} v_i s_i > \sum_{i \in F(J)} v_i - b(J) & \\
& s_i \in \{0,1\} & i \in F(J)
\end{aligned}
$$

If $z < 1$, an inequality (2) is violated.

We consider $F(J)$ for $J = J(F')$ and $F' \in F_C$. To find covers in $CC(F(J))$ we use the separation procedure described in Section 4.3 with the following modifications: The facilities in $F(J)$ are ordered according to $\bar{z}_i/v_i$ in non-decreasing fashion and b^* is initialized with the maximum capacity of the facilities in $F(J)$.

4.5. Separation of inequalities (Cut_{Cov})

In the separation of (Cut_{Cov}) we consider all inequalities of the form (3) that were found by the separation procedure for (Cov_F) and (MC_F). For the corresponding set of facilities $\hat{F}$ and right-hand side p we randomly generate up to p sets $\bar{F} \subseteq \hat{F}$ such that $|\bar{F}| = |\hat{F}| - p + 1$. We separate inequalities (Cut_{Cov}) by running a maximum flow algorithm on graph G' defined as in Section 4.1, but with capacities of 1 on arcs it if $i \in \bar{F}$ and 0 if $i \notin \bar{F}$.

4.6. Separation of inequalities ($\text{Cov}_{\delta^-(W)}$)

Given a cut set $W \subseteq V \setminus \{r\}$ and the set of facilities contained in that cut set, $F' = W \cap F$, a violated cut set cover inequality is detected by solving the following integer program:

$$\begin{aligned} \min\ z = & \sum_{ij \in \delta(W)} \bar{x}_{ij} s_{ij} - \sum_{l \in F'} (\bar{z}_l - 1) t_l \\ \text{s.t.} \quad & \sum_{ij \in \delta^-(W)} u_{ij} s_{ij} + \sum_{l \in F'} d_l t_l > \sum_{ij \in \delta^-(W)} u_{ij} \\ & s_{ij} \in \{0,1\} \quad ij \in \delta^-(W) \\ & t_l \in \{0,1\} \quad l \in F' \end{aligned}$$

A ($\text{Cov}_{\delta^-(W)}$) inequality is violated if $z < 1$.

We separate inequalities ($\text{Cov}_{\delta^-(W)}$) as follows: All cut sets W that are obtained during the separation of inequalities (Cut_{SCF}) are kept in a pool. We choose $F' = \{l \in F \cap W \mid \bar{z}_l > 0.1\}$. Then we use the following heuristic procedure to find minimal covers $C \in MC(W, F'')$, where $F'' \subseteq F'$:

1. Sort the items in $\delta^-(W)$ and F' in non-decreasing order of $(1 - \bar{z}_i)/d_i$ and $\bar{x}_{ij}/u_{ij}$ and store them in a list L. Initialize the cover C and F'' as empty sets and initialize $b^* = \sum_{ij \in \delta^-(W)} u_{ij}$.
2. Remove an item from the head of the sorted list L.
   (a) If it is an arc and its weight is larger than b^*, ignore it, otherwise insert it into R'.
   (b) If it is a facility insert it into F''.
   If C is now a cover with respect to $\delta^-(W)$ and F'', go to step 4.
3. If L is empty, stop. Otherwise, return to step 2.
4. If the cover inequality corresponding to $C \in MC(W, F'')$ is violated by $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$, output it.
5. Let $ij^* = \arg\max_{ij \in C} u_{ij}$ be the arc in C with the highest capacity. Set $u^* = u_{ij^*}$ and delete ij^* from C. Return to step 2.

5. Computational results

In this section we report the results of our computational experiments. They were performed on a desktop machine with an 8-core Intel Core i7 CPU at 2.80 GHz and 8 GB RAM. Each run was performed on a single processor. We used the CPLEX [21] branch-and-cut framework, version 12.2. All cutting plane generation procedures provided by CPLEX are turned off unless stated explicitly. All heuristics provided by CPLEX are turned off. The other parameters are set to their default values.

5.1. Branch-and-cut framework

The settings described in this section are the result of our preliminary testing.

To reduce the number of constraints that need to be identified by our separation routines we add degree balance constraints and subtour elimination constraints for cycles of size two to our model:

$$x_{ji} \leq x(\delta^+(i)) + z_i \qquad (j,i) \in A_S, i \in F \tag{4a}$$
$$x_{ji} \leq x(\delta^+(i)) \qquad (j,i) \in A_S, i \in V_S \setminus (F \cup \{r\}) \tag{4b}$$
$$z_i \leq x(\delta^-(i)) \qquad i \in F \tag{4c}$$
$$x_{ij} \leq x(\delta^-(i)) \qquad (i,j) \in A_S, i \in V_S \setminus \{r\} \tag{4d}$$
$$x_{ij} + x_{ji} \leq 1 \qquad (i,j) \in A_S,\ i < j,\ i,j \neq r \tag{4e}$$

In order to reduce the size of the linear programs solved throughout the process we relax constraints (1d) and add them only if they are violated. Separation procedures are called in the following order: (EKS) - (Cov_F) - (MC_F) - (1d) - ($\text{Cov}_{\delta^-(W)}$) - (Cut_{Cov}) - (Cut_Z) - (Cut_X) - (Cut_{SCF}). To prevent a tailing off effect of the separation procedures we stop separating valid inequalities if the lower bound has improved by less than 0.05% for the last 10 calls of the separation procedures. We apply this rule in each node of the branch-and-bound tree.

Inequalities (Cov_F), (MC_F), ($\text{Cov}_{\delta^-(W)}$), (Cut_{Cov}) and (Cut_{SCF}) are only separated at the root node of the branch-and-bound tree. Inequalities (EKS) and (1d) are separated at every node, separation of (Cut_X) is done at every 10th node and separation of (Cut_Z) is done at every 100th node. In all nodes of the branch-and-bound tree we ensure the feasibility of our model by testing potential integer solutions for violation of inequalities (Cut_{SCF}).

To improve the computational efficiency of the separation procedures for cut set inequalities, we search for nested minimum cardinality cuts. To do so, all capacities in the respective separation graph are increased by some $\epsilon > 0$. Thus, every detected violated cut contains the least possible number of arcs. We resolve the linear program after adding at most 30 violated inequalities of any class. Finally, we randomly choose the target nodes to search for violated cuts.

5.2. Instances

We generated a set of realistic benchmark instances derived from real world data we were given. The real world data contain most of the information needed for complete CapConFL instances: The sets of facilities, Steiner nodes and edges of the core network; a set of customers with associated demands; a set of assignment arcs connecting customers and facilities, including their distance and an estimate of the bandwidth provided by the respective assignment arc; lengths of core edges and assignment arcs. These inputs define five graphs with different topologies that will be denoted by `A`, `B`, `C`, `D` and `E`. To complete the instances with respect to the input required by CapConFL we applied the following steps:

- For each instance a minimum customer bandwidth is selected, assignment arcs that provide less than this bandwidth are removed. We chose 20, 25 and 30 MBit/s and denote this by `20`, `25` and `30` in the instance label.
- At most 20 assignment arcs per customer are considered.
- Customers without assignment arcs are removed and facilities without assignment arcs are replaced by Steiner nodes.

- Steiner nodes with a degree of two and their adjacent edges are replaced by a single edge.
- A technology for each facility is randomly selected. For FTTB instances we consider the following combinations of capacity, demand and cost: $(32, 4, 4000)$, $(64, 5, 6000)$, $(128, 7, 8000)$. For FTTC instances we choose between $(64, 4, 13000)$, $(128, 4, 16000)$ and $(192, 4, 20000)$.
- Edge capacities are uniformly randomly selected from $[0.7\mu, 1.3\mu]$, where μ is equal to the demand of the smallest set of facilities needed to feasibly assign the customers, given the facility capacities chosen before [9].

The key figures for the instances we use are listed in Table 1.

5.3. Comparison against basic model and general purpose solver

In the first part of our computational study we assess the influence of the cutting plane generation procedures built into CPLEX compared to the influence of the valid inequalities proposed in this work. To this end we ran our model with the following different settings: *Basic* is the cut set based model corresponding to SCF, i.e., the model consisting of constraints (1c)-(1g), (Cut_{SCF}) and (4a)-(4e). *Basic+CPX* is the basic model with all CPLEX cuts turned on. *All VI* is the basic model with all valid inequalities from Section 3 added. *All VI+CPX* is the basic model with all valid inequalities and CPLEX cuts turned on.

In Table 1 we compare the LP gaps (g_{LP}) and time to solve the LP relaxation (t_{LP}) for these four models. We calculated the gaps as $(UB - LB)/UB$, where UB is the best known integer solution found in all our tests and LB is the solution value of the LP relaxation of the respective model. In the last two lines we show the mean and median of the values in the respective column. The best LP gap of the four models is shown in bold.

From the results in Table 1 we conclude that the model without valid inequalities provides a weak LP bound with an LP gap of 14.28% on average over the considered instance set. The cutting planes provided by CPLEX can reduce the LP gaps of the basic model by almost one half to an average of 7.23%. This average gap is still substantial compared to 1.31% obtained by the model that is strengthened by the valid inequalities proposed in this paper. Using CPLEX cuts in addition only improves the average gap to 1.13%.

5.4. Influence of different sets of valid inequalities

In the second part of our computational study we assess the influence of the different sets of valid inequalities proposed in this work. We compare five different settings that differ by the sets of valid inequalities considered. For each setting we add a subset of valid inequalities to the basic model described above. Settings (Cut_Z), (Cut_X) and (Cut_Z)+(Cut_X) are self-explaining. Setting *All VI* is defined as above and setting *Most VI* uses inequalities (Cut_Z), (Cut_X), (EKS), (Cov_F) and (MC_F).

For each of these settings, Table 2 shows the gap of the linear programming relaxation, g_{LP}, calculated as in Table 1, the time needed to solve linear programming relaxation, t_{LP}, and the number of cutting planes added, *Cuts*. In the last two lines we show the mean and median of the values in the respective column. The best LP gap of all models is shown in bold.

We would like to point out several interesting aspects. The LP gaps of setting (Cut_Z) are substantially larger than the ones of all other settings. Surprisingly the same does not hold for setting (Cut_X), which on average gives even stronger LP bounds than setting (Cut_Z)+(Cut_X). We trace the difference between the gaps of (Cut_X) and (Cut_Z)+(Cut_X) to the criteria we used to prevent

	Instance properties							Basic		Basic+CPX		All VI		All VI+CPX	
	$\|V\|$	$\|E\|$	$\|F\|$	$\|R\|$	$\|E_S\|$	$\|E_R\|$	UB	g_{LP}	t_{LP}	g_{LP}	t_{LP}	g_{LP}	t_{LP}	g_{LP}	t_{LP}
A20-FTTC	2405	12827	1504	805	1636	11191	3315257	12.98	10	7.67	45	**3.56**	33	**3.56**	36
A25-FTTC	2283	11349	1371	794	1525	9824	3717919	9.54	11	4.15	18	3.36	36	**3.11**	43
A30-FTTC	1949	7689	1053	694	1289	6400	3356273	8.62	5	4.74	12	3.45	29	**2.77**	38
B20-FTTC	2300	9695	1624	510	1824	7871	4601666	15.87	17	4.37	62	**0.10**	39	**0.10**	39
B25-FTTC	2265	9141	1580	507	1792	7349	5051559	14.42	13	5.37	29	0.15	36	**0.07**	41
B30-FTTC	1848	5610	1152	450	1432	4178	6030938	9.18	8	4.06	16	0.10	17	**0.02**	18
C20-FTTC	4742	21388	3237	1160	3720	17668	8897504	21.06	149	11.12	409	0.45	141	**0.36**	152
C25-FTTC	4498	17477	2961	1134	3498	13979	9734526	16.29	55	9.91	228	0.57	134	**0.53**	145
C30-FTTC	3269	9303	1748	840	2557	6746	9332930	11.14	47	4.73	75	0.82	35	**0.70**	58
D20-FTTC	4042	24923	2241	1463	2661	22262	7962207	13.18	50	8.85	156	1.93	63	**1.90**	79
D25-FTTC	3925	21403	2106	1449	2557	18846	9490801	8.86	38	4.31	76	1.32	62	**1.01**	74
D30-FTTC	3407	11539	1588	1213	2274	9265	9971466	4.03	12	2.66	30	0.72	61	**0.45**	73
E20-FTTC	3426	18045	2143	1038	2492	15553	4858219	23.95	101	15.81	302	1.06	61	**0.86**	93
E25-FTTC	3290	12574	1965	1023	2369	10205	7419660	13.21	58	5.13	114	0.54	51	**0.34**	74
E30-FTTC	2149	4429	750	497	1747	2682	6082850	6.93	7	2.54	9	**0.00**	21	**0.00**	21
A20-FTTB	2405	12827	1504	805	1636	11191	2506131	16.00	10	9.56	37	3.23	40	**2.42**	48
A25-FTTB	2267	11284	1370	778	1525	9759	2741939	12.64	8	7.27	33	3.26	38	**2.81**	47
A30-FTTB	1949	7689	1053	694	1289	6400	2406722	10.01	4	3.75	17	2.73	26	**2.14**	37
B20-FTTB	2300	9695	1624	510	1824	7871	3781915	19.99	13	7.83	66	0.26	40	**0.21**	49
B25-FTTB	2265	9141	1580	507	1792	7349	4146838	18.22	11	5.77	31	0.17	37	**0.07**	37
B30-FTTB	1848	5610	1152	450	1432	4178	4797642	12.12	10	3.75	18	0.06	19	**0.05**	20
C20-FTTB	4742	21388	3237	1160	3720	17668	7337282	26.20	176	14.11	397	0.62	158	**0.56**	171
C25-FTTB	4498	17477	2961	1134	3498	13979	7836463	20.75	66	10.93	229	0.78	120	**0.69**	159
C30-FTTB	3269	9303	1748	840	2557	6746	7332424	14.18	44	6.85	63	0.89	56	**0.79**	61
D20-FTTB	4042	24923	2241	1463	2661	22262	7056336	13.94	46	9.30	131	1.06	57	**1.02**	68
D25-FTTB	3925	21403	2106	1449	2557	18846	8315107	10.26	39	6.22	78	1.44	65	**1.20**	71
D30-FTTB	3407	11539	1588	1213	2274	9265	8210407	4.67	11	3.05	28	0.56	50	**0.41**	60
E20-FTTB	3426	18045	2143	1038	2492	15553	3956572	29.98	188	19.18	413	1.38	64	**1.23**	88
E25-FTTB	3290	12574	1965	1023	2369	10205	5885571	19.53	99	9.27	170	3.07	65	**2.89**	60
E30-FTTB	2149	4429	750	497	1747	2682	4651295	10.53	5	4.66	9	1.55	24	**1.48**	23
mean								14.28	44	7.23	110	1.31	56	**1.13**	66
median								13.19	15	6.00	62	0.86	45	**0.75**	59

Table 1: Comparison of LP relaxation lower bounds with and without CPLEX cuts

	(Cut$_Z$)			(Cut$_X$)			(Cut$_Z$)+(Cut$_X$)			Most VI			All VI		
	g_{LP}	t_{LP}	*Cuts*	g_{LP}	t_{LP}	*Cuts*	g_{LP}	t_{LP}	*Cuts*	g_{LP}	t_{LP}	*Cuts*	g_{LP}	t_{LP}	*Cuts*
A20-FTTC	8.56	32	5711	3.57	48	3879	3.58	27	4168	**3.56**	31	4776	**3.56**	33	4523
A25-FTTC	4.48	20	4623	3.50	94	2984	3.51	31	3101	3.38	32	3528	**3.36**	36	3446
A30-FTTC	7.12	17	3565	4.05	47	2987	4.05	24	3144	**3.45**	27	3855	**3.45**	29	3660
B20-FTTC	11.85	105	7323	0.15	39	4459	0.15	26	5100	**0.10**	39	5140	**0.10**	39	5140
B25-FTTC	10.40	26	5669	0.18	50	4442	0.18	33	4815	**0.15**	35	4908	**0.15**	36	4908
B30-FTTC	5.03	17	3447	0.12	24	2382	0.12	16	2661	**0.10**	17	2728	**0.10**	17	2728
C20-FTTC	12.98	351	14191	0.54	293	9042	0.54	144	9813	**0.45**	142	10114	**0.45**	141	9831
C25-FTTC	8.32	265	11612	0.62	215	6947	0.62	122	7298	**0.56**	122	8107	0.57	134	7710
C30-FTTC	4.91	62	4514	0.98	79	3290	0.98	32	3809	**0.82**	35	3873	**0.82**	35	3799
D20-FTTC	7.79	232	13921	2.00	97	7561	2.00	51	8193	**1.93**	62	8751	**1.93**	63	8124
D25-FTTC	3.68	62	8784	1.33	68	5901	1.34	39	6056	**1.32**	62	6454	**1.32**	62	6529
D30-FTTC	1.28	50	6150	0.82	109	4578	0.82	55	4737	**0.72**	58	5001	**0.72**	61	4974
E20-FTTC	13.69	142	12891	1.10	140	8633	1.10	58	9566	**1.06**	61	10021	**1.06**	61	10021
E25-FTTC	4.17	50	8515	0.57	62	6046	0.58	35	6303	**0.54**	50	6656	**0.54**	51	6656
E30-FTTC	2.29	12	2222	0.01	20	1682	0.01	14	1795	**0.00**	22	1960	**0.00**	21	1849
A20-FTTB	10.46	25	5426	3.46	67	3681	3.58	38	3640	**3.23**	40	4880	**3.23**	40	4394
A25-FTTB	5.09	20	3857	3.53	78	2740	3.57	29	2851	**3.26**	32	3789	**3.26**	38	3457
A30-FTTB	8.25	14	2755	3.59	26	2036	3.60	13	2133	**2.73**	22	3074	**2.73**	26	2832
B20-FTTB	15.82	74	7895	0.36	54	4497	0.36	34	4977	**0.26**	38	5365	**0.26**	40	4944
B25-FTTB	13.71	45	6044	0.26	50	4503	0.26	32	4948	**0.17**	37	5051	**0.17**	37	4913
B30-FTTB	7.20	16	3478	0.11	25	2387	0.11	18	2661	**0.06**	22	2601	**0.06**	19	2664
C20-FTTB	16.75	657	15243	0.71	298	8207	0.71	148	8980	**0.62**	156	9785	**0.62**	158	9475
C25-FTTB	11.26	306	12195	0.88	93	6765	0.88	67	7266	**0.78**	118	7793	**0.78**	120	7364
C30-FTTB	6.73	84	4785	1.10	93	3380	1.10	50	3693	0.90	57	3942	**0.89**	56	3756
D20-FTTB	8.39	210	12375	1.10	65	5535	1.10	30	6459	**1.06**	59	7591	**1.06**	57	6476
D25-FTTB	4.56	75	8617	1.46	154	4824	1.46	50	4950	**1.44**	59	6104	**1.44**	65	5565
D30-FTTB	1.34	28	6054	0.67	43	4030	0.67	24	4287	**0.56**	46	4543	**0.56**	50	4308
E20-FTTB	18.30	213	12824	1.39	139	7394	1.39	68	7869	**1.38**	74	8747	**1.38**	64	8379
E25-FTTB	8.41	82	8547	3.09	113	5820	3.12	61	6010	**3.07**	50	6699	**3.07**	65	6443
E30-FTTB	5.36	20	2288	1.68	10	1617	1.68	13	1811	**1.55**	24	1937	**1.55**	24	1937
mean	8.27	110	7517	1.43	90	4740	1.44	46	5103	**1.31**	54	5592	**1.31**	56	5360
median	8.02	56	6102	1.04	67	4478	1.04	33	4881	**0.86**	43	5026	**0.86**	45	4928

Table 2: Comparison of LP relaxation lower bounds with different sets of valid inequalities turned on

a tailing off effect during separation. A comparison of the running times shows that separating valid inequalities with a different structure improves the overall running time of the LP relaxation. Approaches (Cut_Z) and (Cut_X) need 110 and 90 seconds on average, respectively whereas approach (Cut_Z)+(Cut_X) only takes 46 seconds to compute approximately the same lower bounds as (Cut_X). The separation routines in approaches *Most VI* and *All VI* require an additional 10 and 12 seconds on average. Thereby, the average LP gaps are improved from 1.44% ((Cut_Z)+(Cut_X)) to 1.31% (*Most VI* and *All VI*). However, *All VI* does not improve upon *Most VI* significantly.

There is a notable difference in the numbers of valid inequalities that were detected during the LP relaxations of the different settings. By far the most inequalities are found by setting (Cut_Z), even though the obtained LP bound is comparably weak. This is consistent with the long running time of the LP relaxation of setting (Cut_Z). Rather surprising is the fact, that setting *All VI* obtains the same LP bound as setting *Most VI* for 28 out of 30 settings but the number of valid inequalities found by *All VI* is smaller for 22 and larger for only 2 instances.

Table 3 shows the respective gap of the five different settings after 3, 10, 30 and 60 minutes. For these results we calculate the gaps as $(UB_t - LB_t)/UB_t$ where UB_t is the best integer solution found by the respective setting after t minutes and LB_t is the lower bound after t minutes. For each instance and running time the smallest gap of all five settings is indicated in bold. If no integer solution is available after t minutes we indicate this by a dash in the respective column. For each setting and time t the last three lines of the table indicate the mean and median of gaps over the instance set and how often the respective approach gives the smallest gap of all settings.

Contrary to what the LP gaps in Table 2 suggest the setting *All VI* with all valid inequalities enabled outperforms the other settings on a majority of instances. The performance of setting *Most VI* is only slightly worse (0.09%, 0.07% and 0.05% larger gap after 10, 30 and 60 minutes). The other settings perform significantly worse with between 0.46% and 2.58% larger gaps on average.

In Figures 7 and 8 we give a graphical illustration of the numbers reported in Table 3. The coordinates of each mark indicate how many out of 30 instances (ordinate axis) were solved within a given optimality gap (abscissa). Figure 7 shows the performance after 3 and 10 minutes and Figure 8 shows the performance after 30 and 60 minutes.

6. Conclusions

In this paper we introduce the Capacitated Connected Facility Location problem. We introduce various sets of cut set, minimum cardinality, cover and cut-set-cover inequalities to strengthen a basic integer programming model. After a detailed discussion of separation procedures we report the results of our computational experiments. These confirm that the proposed approach finds solution within a small optimality gap averaging to less than 2% for a set of realistic new benchmark instances.

Acknowledgements

Stefan Gollowitzer was supported by the Fonds de recherche du Québec - Nature et technologies under grant 163879 within the *Programme de Stages Internationaux*. This support is gratefully acknowledged.

	(Cut$_Z$)				(Cut$_X$)				(Cut$_Z$)+(Cut$_X$)				Most VI				All VI			
	g_3	g_{10}	g_{30}	g_{60}	g_3	g_{10}	g_{30}	g_{60}	g_3	g_{10}	g_{30}	g_{60}	g_3	g_{10}	g_{30}	g_{60}	g_3	g_{10}	g_{30}	g_{60}
A20-FTTC	7.28	5.44	4.97	4.12	-	-	4.76	4.76	4.25	4.22	4.03	**3.45**	4.34	4.22	4.21	4.18	**3.68**	**3.66**	**3.62**	3.61
A25-FTTC	5.54	5.26	5.04	4.96	-	-	4.99	4.96	5.88	5.82	5.77	5.74	4.44	4.37	4.32	4.29	**3.74**	**3.66**	**3.61**	**3.58**
A30-FTTC	6.65	5.82	5.31	5.17	-	5.81	5.66	5.52	5.72	5.55	4.37	3.93	**4.87**	3.61	**3.31**	3.02	4.93	**3.58**	3.40	**2.86**
B20-FTTC	11.59	5.55	3.44	2.13	0.07	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
B25-FTTC	7.58	4.65	2.25	1.20	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
B30-FTTC	1.48	0.69	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
C20-FTTC	-	15.91	6.21	5.03	-	-	1.94	1.48	-	1.83	1.79	0.61	-	**0.82**	0.59	**0.32**	-	1.00	**0.52**	0.46
C25-FTTC	-	5.85	4.09	3.72	-	1.03	0.99	0.90	0.99	0.93	0.88	0.85	1.06	0.86	0.70	0.66	**0.83**	**0.73**	**0.65**	**0.62**
C30-FTTC	5.22	3.37	3.00	2.68	1.47	1.42	1.33	1.24	1.34	1.19	1.10	1.01	0.95	0.82	0.30	0.05	**0.91**	**0.73**	**0.13**	**0.00**
D20-FTTC	-	-	3.34	2.83	-	-	2.59	2.59	-	2.49	2.46	2.42	-	2.53	**2.00**	**1.90**	-	**2.15**	2.10	2.07
D25-FTTC	4.14	2.37	2.07	1.77	-	-	1.99	1.80	**2.39**	1.61	1.24	1.18	-	1.62	**1.18**	**1.06**	-	**1.60**	1.46	1.13
D30-FTTC	1.53	1.19	0.96	0.89	-	1.31	1.23	0.96	1.23	**0.69**	**0.55**	**0.53**	**1.17**	1.04	0.87	0.59	1.20	0.91	0.60	0.58
E20-FTTC	-	8.32	5.50	4.25	-	1.98	1.77	1.66	1.79	1.43	1.22	1.16	**1.73**	**1.21**	**0.85**	**0.80**	**1.73**	**1.21**	**0.85**	**0.80**
E25-FTTC	4.37	2.56	2.24	1.85	1.99	1.82	0.70	0.67	**0.70**	**0.55**	**0.42**	**0.32**	0.73	0.59	0.46	0.35	0.73	0.59	0.46	0.35
E30-FTTC	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
A20-FTTB	8.31	7.29	5.56	5.21	-	-	5.07	5.05	**5.73**	5.68	5.62	5.49	-	3.17	3.06	2.96	-	**2.70**	**2.53**	**2.43**
A25-FTTB	6.90	6.58	6.27	4.47	-	3.52	3.45	3.37	4.82	4.73	4.59	4.46	3.50	3.26	**3.01**	**2.97**	**3.38**	**3.19**	3.05	3.02
A30-FTTB	6.85	5.22	4.86	4.65	-	6.13	5.98	5.95	5.06	4.88	4.78	4.73	2.94	2.66	**2.29**	**2.10**	**2.84**	**2.63**	2.42	2.34
B20-FTTB	16.17	11.15	6.52	4.58	0.65	0.60	0.55	0.43	0.44	0.37	0.25	0.23	**0.30**	0.27	**0.16**	**0.14**	0.32	**0.22**	0.18	0.16
B25-FTTB	10.99	7.87	5.22	4.11	0.58	0.47	0.38	0.34	0.23	0.10	0.05	0.03	**0.04**	**0.00**	**0.00**	**0.00**	0.06	0.02	**0.00**	**0.00**
B30-FTTB	3.57	1.41	**0.00**	**0.00**	0.02	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
C20-FTTB	-	-	11.43	8.69	-	-	1.32	1.31	-	1.13	0.88	0.85	-	**0.76**	**0.72**	**0.58**	-	0.80	0.78	0.72
C25-FTTB	-	-	5.43	4.68	-	-	1.84	1.19	-	1.32	1.02	1.00	-	0.89	0.69	0.65	-	**0.85**	**0.61**	**0.46**
C30-FTTB	7.32	2.79	1.16	0.81	-	0.69	0.66	0.62	0.59	0.43	0.37	0.34	0.42	0.35	0.27	0.25	**0.33**	**0.25**	**0.21**	**0.16**
D20-FTTB	-	4.68	2.73	2.25	-	-	-	1.41	-	1.21	1.19	1.17	-	1.39	1.17	1.14	-	**1.19**	**1.07**	**1.04**
D25-FTTB	-	2.96	2.10	1.85	-	-	-	2.29	-	**1.77**	1.72	1.70	-	1.93	1.49	1.45	-	2.28	**1.31**	**1.29**
D30-FTTB	1.40	1.12	0.92	0.80	-	0.83	0.70	0.62	0.87	0.58	0.43	0.41	0.64	0.51	**0.36**	**0.33**	**0.54**	**0.47**	0.40	**0.33**
E20-FTTB	-	10.18	6.09	4.47	-	-	1.66	1.63	2.33	**1.52**	1.36	1.35	2.57	1.79	1.34	**1.22**	**1.61**	1.54	**1.33**	1.31
E25-FTTB	-	3.92	2.39	2.13	-	1.68	0.87	0.71	**0.77**	**0.51**	**0.44**	0.41	1.41	0.62	0.45	**0.28**	0.91	0.57	0.50	0.44
E30-FTTB	2.18	1.15	0.12	0.10	0.40	0.20	0.16	0.15	**0.14**	0.12	0.10	0.08	**0.14**	**0.04**	**0.00**	**0.00**	**0.14**	**0.04**	**0.00**	**0.00**
mean	-	-	3.64	2.98	-	-	-	1.72	-	1.69	1.55	1.45	-	1.31	1.13	1.04	-	**1.22**	**1.06**	**0.99**
median	-	-	3.39	2.75	-	-	-	1.21	-	1.16	0.95	0.85	-	0.84	0.70	0.58	-	**0.83**	**0.61**	**0.52**
best	1	1	3	3	3	5	5	5	10	10	8	8	11	10	16	**18**	**16**	**22**	**19**	**18**

Table 3: Gaps to model upper bound

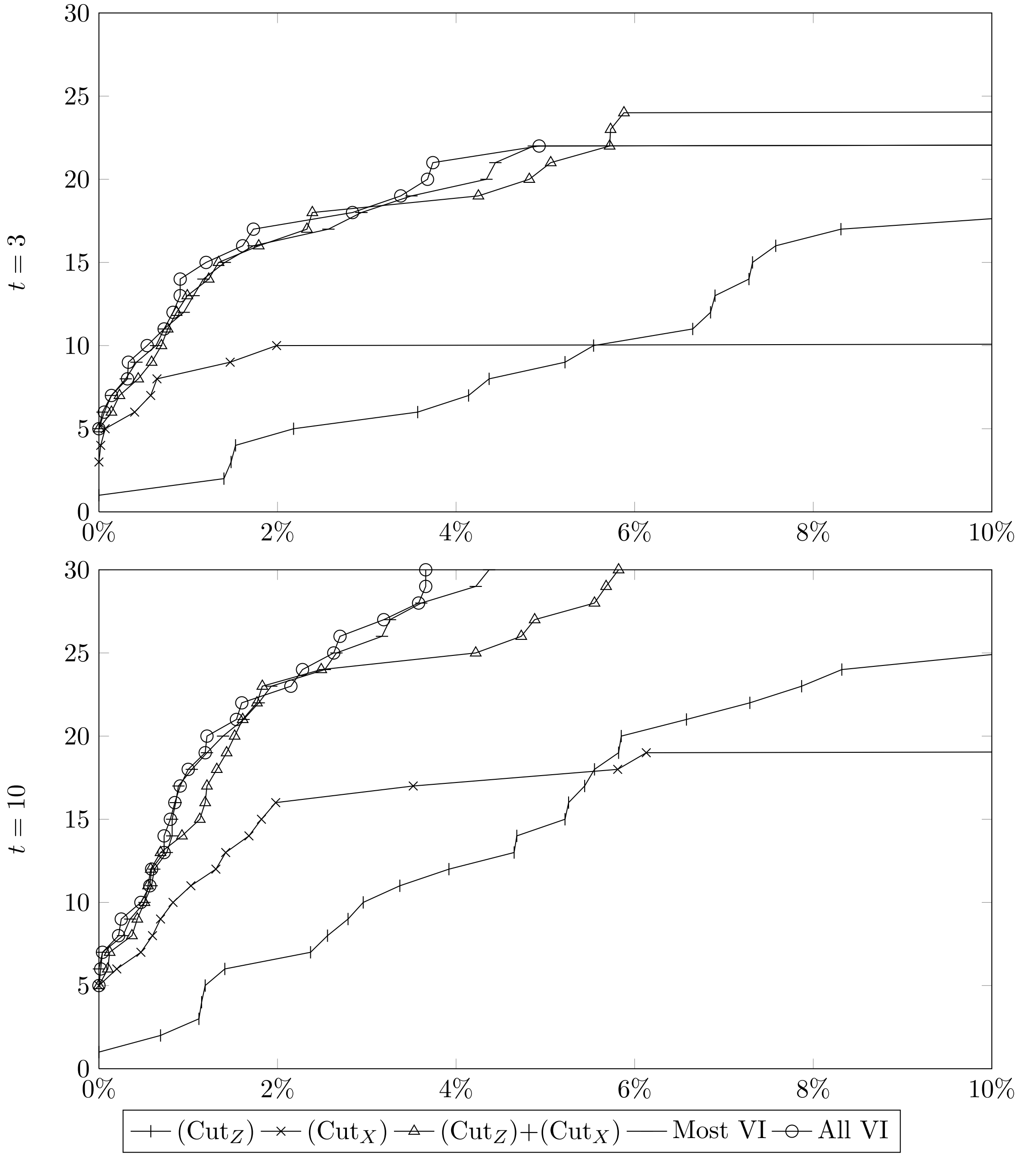


Figure 7: Performance chart for 3 minutes (top) and 10 minutes (bottom) runtime

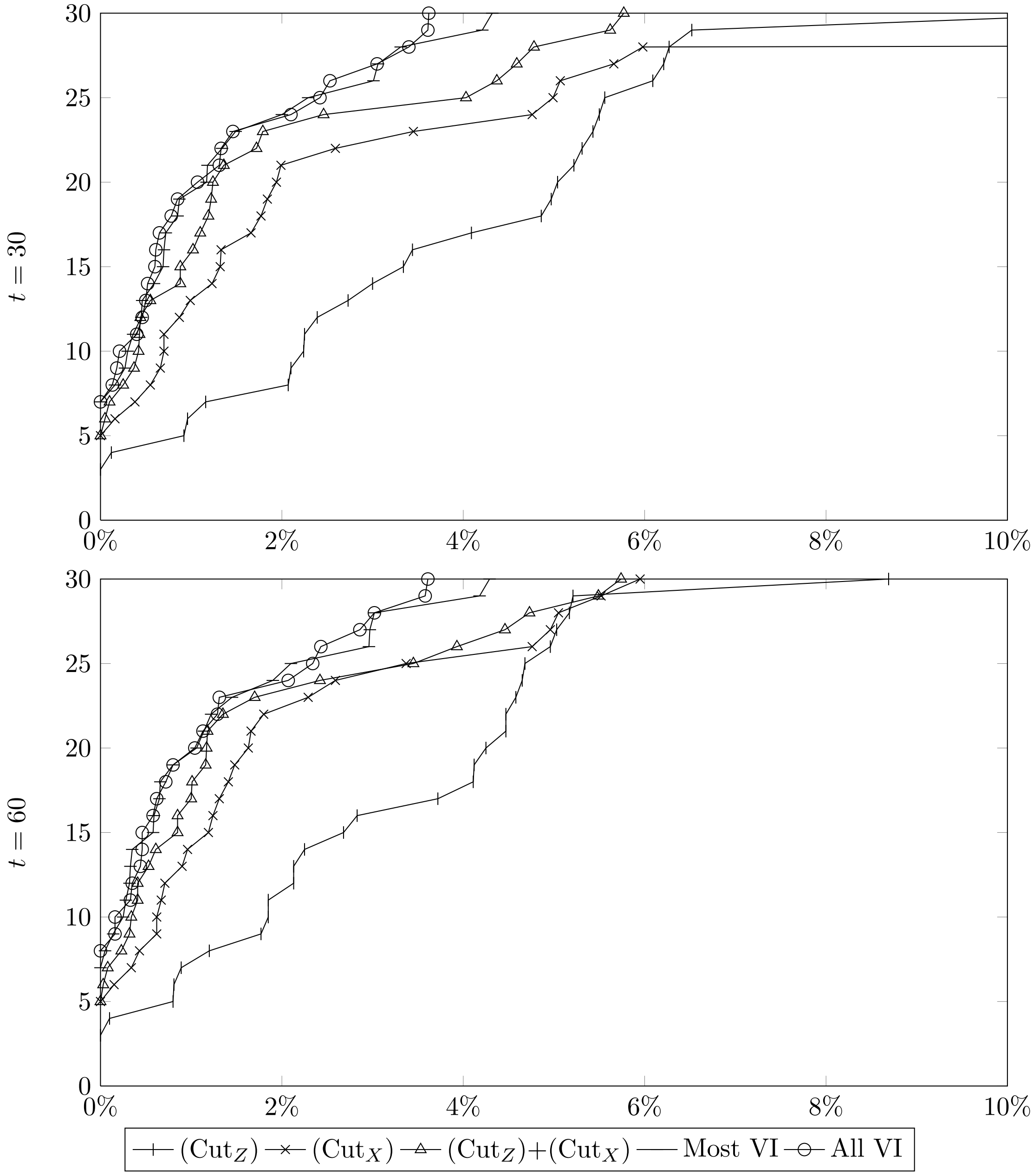


Figure 8: Performance chart for 30 minutes (top) and 60 minutes (bottom) runtime

Bibliography

[1] K. Aardal, Y. Pochet, and L. A. Wolsey. Capacitated facility location: Valid inequalities and facets. *Mathematics of Operations Research*, 20(3):562–582, 1995.

[2] A. Arulselvan, A. Bley, S. Gollowitzer, I. Ljubic, and O. Maurer. MIP modeling of incremental connected facility location. In J. Pahl, T. Reiners, and S. Voß, editors, *INOC*, volume 6701 of *Lecture Notes in Computer Science*, pages 490–502. Springer, 2011.

[3] M. G. Bardossy and S. Raghavan. Dual-based local search for the connected facility location and related problems. *INFORMS Journal on Computing*, 22(4):584–602, 2010.

[4] A. Ceselli, F. Liberatore, and G. Righini. A computational evaluation of a general branch-and-price framework for capacitated network location problems. *Annals OR*, 167(1):209–251, 2009.

[5] M. Chouman, T. G. Crainic, and B. Gendron. Commodity representations and cutset-based inequalities for multicommodity capacitated fixed-charge network design. Technical Report CIRRELT-2011-56, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), CIRRELT, 2011.

[6] I. A. Contreras and J. A. Díaz. Scatter search for the single source capacitated facility location problem. *Annals OR*, 157(1):73–89, 2008.

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

[8] T. G. Crainic and M. Gendreau. Cooperative parallel tabu search for capacitated network design. *J. Heuristics*, 8(6):601–627, 2002.

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

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

[11] Q. Deng and D. Simchi-Levi. Valid inequalities, facets and computational results for the capacitated concentrator location problem. Technical report, Department of Industrial Engineering and Operations Research, Columbia University, New York, 1992.

[12] J. A. Diaz and E. Fernández. A branch-and-price algorithm for the single source capacitated plant location problem. *Journal of the Operations Research Society*, 53:728–740, 2002.

[13] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Connected facility location via random facility sampling and core detouring. *Journal on Computer and System Sciences*, 76: 709–726, December 2010. ISSN 0022-0000.

[14] B. Gendron, T. G. Crainic, and A. Frangioni. Multicommodity capacitated network design. In B. Sanso and P. Soriano, editors, *Telecommunications Network Planning*, pages 1–19. Kluwer Academics Publisher, 1998.

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

[16] I. Ghamlouche, T. G. Crainic, and M. Gendreau. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Annals OR*, 131(1-4):109–133, 2004.

[17] S. Gollowitzer and I. Ljubić. MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research*, 38(2):435–449, 2011.

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

[19] K. Holmberg and D. Yuan. A lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, 48:461–481, 2000.

[20] K. Holmberg, M. Rönnqvist, and D. Yuan. An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research*, 113(3):544 – 559, 1999.

[21] IBM. CPLEX, November 22nd 2011. `http://www.ilog.com/products/cplex/`;.

[22] K. Kaparis and A. N. Letchford. Separation algorithms for 0-1 knapsack polytopes. *Mathematical Programming*, 124(1-2):69–91, 2010.

[23] G. Kliewer and L. Timajev. Relax-and-cut for capacitated network design. In G. S. Brodal and S. Leonardi, editors, *ESA*, volume 3669 of *Lecture Notes in Computer Science*, pages 47–58. Springer, 2005. ISBN 3-540-29118-0.

[24] M. Leitner and G. R. Raidl. Branch-and-cut-and-price for capacitated connected facility location. *Journal of Mathematical Modelling and Algorithms*, 10(3):245–267, 2011.

[25] I. Ljubić. A hybrid VNS for connected facility location. In T. Bartz-Beielstein, M. J. B. Aguilera, C. Blum, B. Naujoks, A. Roli, G. Rudolph, and M. Sampels, editors, *Hybrid Metaheuristics*, volume 4771 of *Lecture Notes in Computer Science*, pages 157–169. Springer, 2007.

[26] I. Ljubić, P. Putz, and J. Salazar-González. Exact approaches to the single-source network loading problem. *Networks*, 59(1):89–106, 2012.

[27] S. Raghavan and D. Stanojević. A note on search by objective relaxation. In *Telecommunications Planning: Innovations in Pricing, Network Design and Management*, volume 33 of *Operations Research/Computer Science Interfaces Series*, pages 181–201. Springer US, 2006.

[28] F. S. Salman, R. Ravi, and J. N. Hooker. Solving the capacitated local access network design problem. *INFORMS Journal on Computing*, 20(2):243–254, 2008.

Stefan Gollowitzer

Contact Information	Georg Siglgasse 9/9 1090 Vienna, Austria *web* http://stefan.gollo.at
Personal Information	Date of Birth: 31 August 1984 Place of Birth: Vienna, Austria Citizenship: Austria

Education

University of Vienna, Vienna, Austria

PhD in Statistics and Operations Research, since March 2010

- Thesis title: *Mixed Integer Programming approaches to network design problems with facility location*
- Supervisors: Prof. Immanuel Bomze and Ass.-Prof. Ivana Ljubić

Vienna University of Technology, Vienna, Austria

Master of Science in Technical Mathematics, October 2003 - January 2010

- Area of study: Business Mathematics

International experience

Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport (CIRRELT), Montréal, Canada

Visiting researcher, cooperation with Prof. Bernard Gendron
April 2011 - December 2011

Technical University of Denmark, Kgs. Lyngby, Denmark

Student exchange, Erasmus Programme of the European Union
September 2007 - June 2008

Journal publications

Ljubić, I. and S. Gollowitzer. Layered Graph Approaches to the Hop Constrained Connected Facility Location Problem. *INFORMS Journal on Computing*, forthcoming.

Gollowitzer, S., and I. Ljubić. MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research*, 38(2): 435–449, 2011.

Articles under review

Gollowitzer S., B. Gendron and I. Ljubić (2012). Capacitated Network Design with Facility Location.

Gollowitzer S., L. Gouveia, G. Laporte, D. L. Pereira and A. Wojciechowski (2012). A Comparison of Several Models for the Hamiltonian p-Median Problem.

Gollowitzer, S., L. Gouveia and I. Ljubić (2011). The Two Level Network Design Problem with Transition Facilities.

Gollowitzer, S. and I. Ljubić (2011). Hop Constrained Connected Facility Location: A Hierarchy of Formulations.

Refereed conference papers

Gollowitzer, S., L. Gouveia and I. Ljubić (2011). A Node Splitting Technique for Two Level Network Design Problems with Transition Nodes. *Lecture Notes in Computer Science*, vol. 6701. 57–70. Proceedings of the 5th International Network Optimization Conference (INOC).

Gollowitzer, S., L. Gouveia and I. Ljubić (2011). The Two Level Network Design Problem with Secondary Hop Constraints. *Lecture Notes in Computer Science*, vol. 6701. 71–76. Proceedings of the 5th International Network Optimization Conference (INOC).

Gollowitzer S., D. L. Pereira and A. Wojciechowski (2011). New models for and numerical tests of the Hamiltonian p-median problem. *Lecture Notes in Computer Science*, vol. 6701. 385–394. Proceedings of the 5th International Network Optimization Conference (INOC).

Arulselvan A., A. Bley, S. Gollowitzer, I. Ljubic and O. Maurer (2011). MIP modeling of Incremental Connected Facility Location. *Lecture Notes in Computer Science*, vol. 6701. 490–502. Proceedings of the 5th International Network Optimization Conference (INOC).

Ljubić, I., and S. Gollowitzer (2010). Modelling the Hop Constrained Connected Facility Location Problem on Layered Graphs. *Electronic Notes in Discrete Mathematics*, vol. 36. 207–214. Proceedings of the International Symposium on Combinatorial Optimization (ISCO).

Awards & Grants

EURO/IFORS scholar at ELAVIO 2012

EURO/IFORS scholarship to attend the ELAVIO 2012, XVI Latin American Summer School on Operations Research in Vale do Vinhedos, Rio Grande do Sul, Brazil

Grant of the *Fonds de recherche du Québec - Nature et technologies*

Grant within the *Programme de Stages Internationaux* for a research visit at the Centre interuniversitaire de recherche sur les reseaux dentreprise, la logistique et le transport (CIRRELT), Montréal, Canada

"ÖGOR-Würdigungspreis"

Awarded by the Austrian Society of Operations Research in September 2010 for the master thesis *MIP Models for (Hop-Constrained) Connected Facility Location*

"Förderungsstipendium"

Awarded by the Vienna University of Technology in December 2009 to attend *INFORMS Telecommunications Conference*, *Optimization Days* and *Spring School on Supply Chain and Transportation Network Design* in Montréal, Canada

Invited talks

Optimization Tools for Last Mile Access Networks. *Centre interuniversitaire de recherche sur les reseaux d'entreprise, la logistique et le transport (CIRRELT).* May 2012, Montréal, Canada.

New models for and numerical tests of the Hamiltonian p-median problem. *Centro de Investigação Operacional, University of Lisbon.* November 2011, Lisbon, Portugal.

Conference Presentations

Capacitated Network Design with Facility Location. *Optimization Days 2012.* May 2012, Montréal, Canada.

Capacitated Network Design with Facility Location. *ELAVIO 2012 - XVI Latin American Summer School on Operations Research.* February 2012, Vale dos Vinhedos, Rio Grande do Sul, Brazil.

A Two Level Network Design Problem with Transition Nodes. *International Network Optimization Conference.* June 2011, Hamburg, Germany.

New models for and numerical tests of the Hamiltonian p-median problem. *Optimization Days 2011.* May 2011, Montréal, Canada.

Hop Constrained Connected Facility Location. *EURO XXIV - 24th European Conference on Operations Research.* July 2010, Lisbon, Portugal.

Using Population-Based Algorithms for Column Generation. *Matheuristics 2010.* June 2010, Vienna, Austria.

Hop Constrained Connected Facility Location. *INFORMS Telecommunications Conference.* May 2010, Montréal, Canada.

Modelling the Hop Constrained Connected Facility Location Problem on Layered Graphs. *International Symposion on Combinatorial Optimization.* March 2010, Hammamet, Tunisia.

Hop Constrained Connected Facility Location. *Workshop on Combinatorial Optimization.* January 2010, Aussois, France.

Teaching Experience

University of Vienna, Vienna, Austria

Lecturer, Spring term 2010

- VK Grundzüge der Wirtschaftsmathematik
- FK Wirtschaftsmathematik 2

Professional activities

Reviewer for

- INFORMS Journal on Computing
- Networks
- Journal of Combinatorial Optimization
- Journal of Optimization Theory and Applications
- Applied Mathematical Modelling
- Central European Journal of Operations Research
- European Journal of Operational Research
- ISCO 2010 - International Symposium on Combinatorial Optimization