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, {ivana.ljubic, stefan.gollowitzer}@univie.ac.at

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 to model HC ConFL: i) as 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 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. Pathan and Buyya (2008)). 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. (2003), 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, $\bigcirc$ represents a Steiner node. $\boxed{\text{i}}$ represents a facility i. $\Diamond$ 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 undirected graph.

Observation 1. *Using the transformation given in Gollowitzer and Ljubić (2011), 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.*

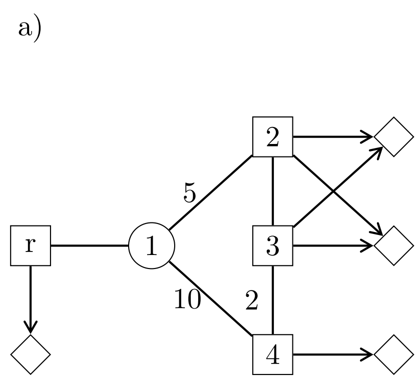

Figure 1: a) Original instance; b) Feasible solution.

Our Contribution We describe the Hop Constrained Connected Facility Location problem, that has not yet been considered in literature. By extending the ideas given by Gouveia et al. (2009) we propose two possibilities for modeling the HC ConFL: i) as 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.

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.*

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. (2006)).

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 contains also an extensive computational study conducted on a set of publicly available benchmark instances.

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 Eisenbrand et al. (2010). Ljubić (2007) 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 (2008) 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 (2010) 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 (2011), the authors present a branch-and-cut-and-price approach for a variant of ConFL with *capacities on facilities.*

In Gollowitzer and Ljubić (2011) 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 Steiner tree problem with hop constraints (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 Minimum Spanning Tree problem with hop constraints (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 Dahl et al. (2006). Gouveia et al. (2009) 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 Steiner tree problem with hop constraints: The problem was first mentioned by Gouveia (1998), who develops a strengthened version of a multi-commodity flow model for 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 Gouveia (1996). Results for instances with up to 100 nodes and 350 edges are presented.

Voß (1999) 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. Gouveia (1999) gives a survey of hop-indexed tree and flow formulations for the hop constrained spanning and Steiner tree problem.

Costa et al. (2008) 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. (2009) the authors introduce two new MIP models for 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 (1999) and Gouveia (1999).

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., Magnanti and Wolsey (1995)). 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 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 (1992), Gouveia (1996) and Gouveia (1998). 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 (2011)), 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 a lifted 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 $D \subseteq A$.

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:

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

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 $p = 1, \dots, H$, $i \in F \setminus \{r\}$. 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).

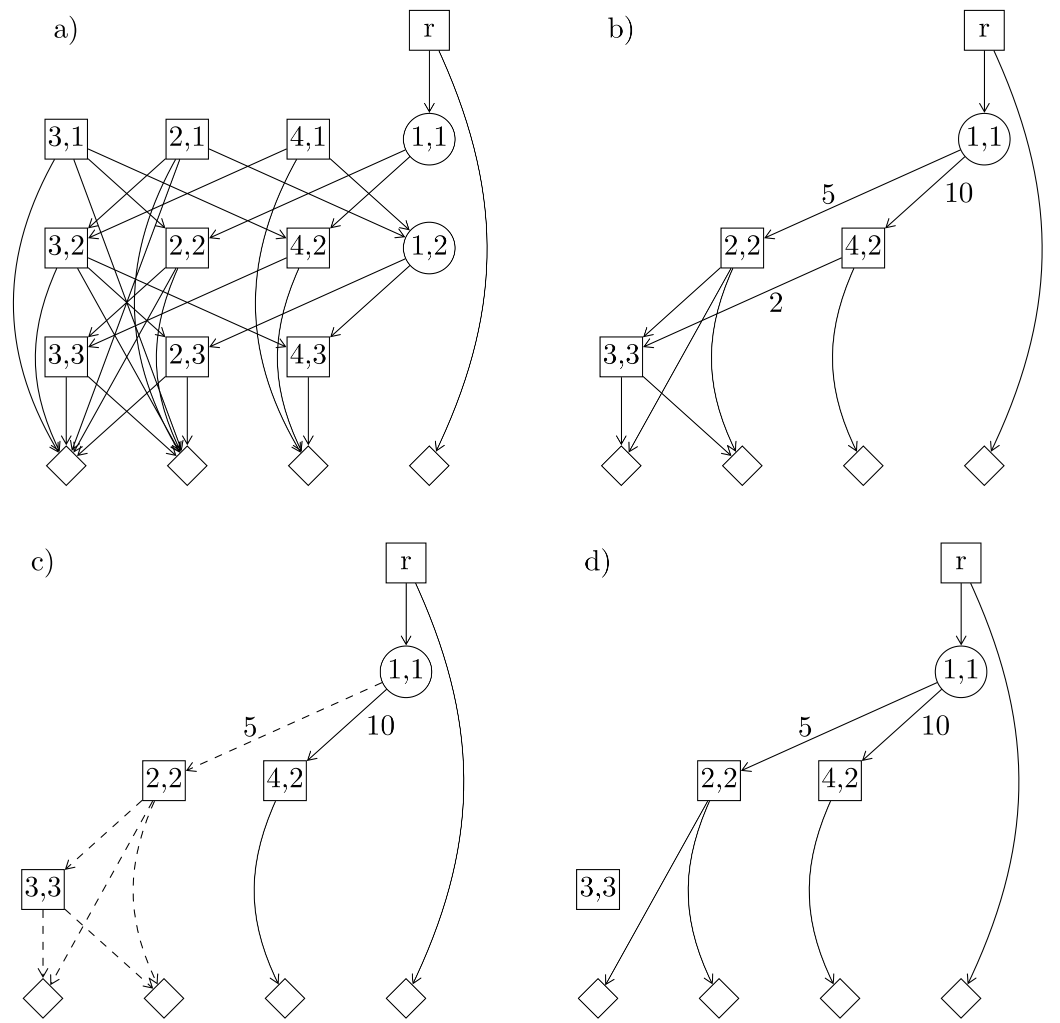

Figure 2: $LG_{x,z}$ for the example shown in Figure 1a) and $H = 3$: a) $LG_{x,z}$ before, and b) after preprocessing; c) The optimal LP-solution for $CUT^F_{x,z}$ – dashed and solid arcs take LP-value of 1/2 and 1, respectively; d) The optimal LP-solution for CUT^R_x which is already MIP-optimal; Figure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

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 (2011) we describe two cut-set based formulations for the (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:

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

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

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

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

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

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

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

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

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 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. $\square$

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 (2011) 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 (2009), it has been shown that cut-set based MIP models on layered graphs represent the tightest formulations known so far for modeling the hop constrained minimum spanning tree problem. 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 MTZ 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. (2009) and Dahl et al. (2006), respectively). In Gollowitzer (2010), the corresponding path- and jump-based MIP models for HC ConFL have been described, and compared to the other extended formulations for HC ConFL with respect to the quality of their lower bounds.

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:

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

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 $i \in F$.

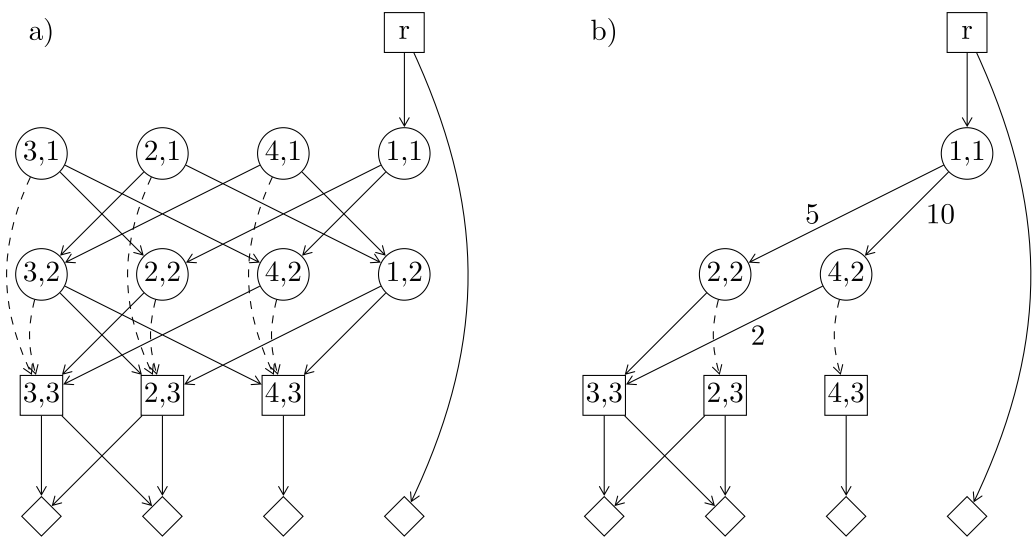

Figure 3: Layered graph LG_x for the instance given in Figure 1a) obtained a) before and b) after preprocessing.

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.

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 convienience, 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: $x_{ij} := \sum_{p=1}^{H} X'^p_{ij}$ 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:

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

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

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

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

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

$$z_r = 1 \qquad (14)$$

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

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

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

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 **c** and **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 Gollowitzer and Ljubić (2011)). 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 problem as the Steiner arborescence on even larger layered graphs. This transformation increases namely the number of nodes by $|F|$, but does not provide stronger lower bounds for the corresponding cut-set formulation (see Gollowitzer and Ljubić (2011)).

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 $1, \dots, H-1$. 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 Gouveia (1999) for solving the Hop Constrained STP. Voß (1999) has observed that this formulation is a disaggregation of a formulation based on Miller-Tucker-Zemlin constraints. Costa et al. (2009) 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 problem as follows:

$$(HOP) \quad \min \sum_{p=1}^{H} \sum_{ij \in A_S} c_{ij} X_{ij}^p + \sum_{jk \in A_R} c_{jk} x_{jk} + \sum_{i \in F} f_i z_i$$

$$\sum_{\substack{i \in S \setminus \{k\}: \\ ij \in A_S}} X_{ij}^{p-1} \geq X_{jk}^p \qquad \forall jk \in A_S,\ j \neq r,\ p = 2, \ldots, H \tag{19}$$

$$\sum_{ij \in A_S} \sum_{p=1}^{H} X_{ij}^p \geq z_j \qquad \forall j \in F \setminus \{r\} \tag{20}$$

$$(11) - (17)$$

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 $p \in \{1, \ldots, H\}$ 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. In Gollowitzer (2010) we have 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 $\frac{v_{LP}(CUT^R_{x,z})}{v_{LP}(CUT^F_{x,z})} \approx (|F| - 1)|H|$.*

Proof. The result given in Gollowitzer and Ljubić (2011) 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. □

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 resumes and illustrates their relations shown in this 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 $LG_{x,z}$
$CUT^{F+}_{x,z}$	Lifted 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}

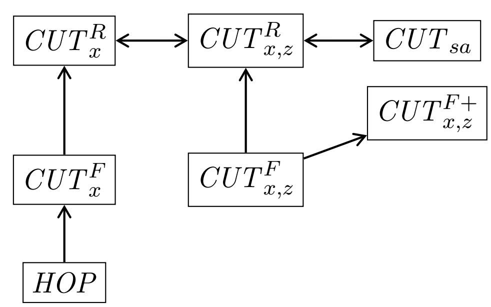

Figure 4: Summary and relations between the LP lower bounds of the presented formulations. In the chart on the right hand side an arrow denotes that the LP bound of its apex is greater or equal than the LP bound of its origin.

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 $W \subseteq S_x \setminus \{r\}$. 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., Koch and Martin (1998)).

Separation Separation of cut-set inequalities (10) and (18) is done in polynomial time by running the maximum-flow algorithm of Cherkassky and Goldberg (1994) on the corresponding support graphs. In 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, $j \in R$. 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 do so, 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 labelled 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 labelled 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 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.

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 variables z_i. Therefore, in our default branch-and-bound

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

```
for each a ∈ A_{S_x}
  do c_a ← X_a + ε
Pool ← ∅
for each i ∈ F_x s.t. z_i > 0
  do { f ← MAXFLOW(LG_x^{core}, c, r, i)
       while f < z_i and |Pool| < M_cut
         do { A_fw ← FORWARD(f)
              A_bw ← BACKWARD(f)
              Pool ← Pool ∪ {X_x[A_fw] ≥ z_i} ∪ {X_x[A_bw] ≥ z_i}
              for each a ∈ A_fw ∪ A_bw
                do c_a ← ∞
              f ← 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}.

implementation, the highest branching priority is assigned to facility variables z_i, $i \in F$. The CPLEX default enumeration strategy is used.

5.2. Data Set

We consider a class of benchmark instances, originally introduced in Ljubić (2007), and also used by Tomazic and Ljubić (2008) and Bardossy and Raghavan (2010). 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.

- We consider a set of non-trivial UFLP instances from UflLib (see Max-Planck-Institut für Informatik (2003)): Instances `mp{1,2}` and `mq{1,2}` have been proposed by Kratica et al. (2001). 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.

Algorithm 5.2: CustomerCuts(LG_x, X)

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

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

- STP instances: Instances `{c,d}n`, for $n \in \{5, 10, 15, 20\}$ were chosen from the OR-library (see Beasley (1990)) 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.

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 of 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

Table 1: Basic properties of benchmark instances.

STP	UFLP	$\|R\|$	$\|F\|$	$\|V \setminus R\|$	$\|E_S\|$	$\|E_R\|$
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

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.

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

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

In Fig 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 2 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 4 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 4 approaches is given in column $|Inst^{OPT}|$. The numbers

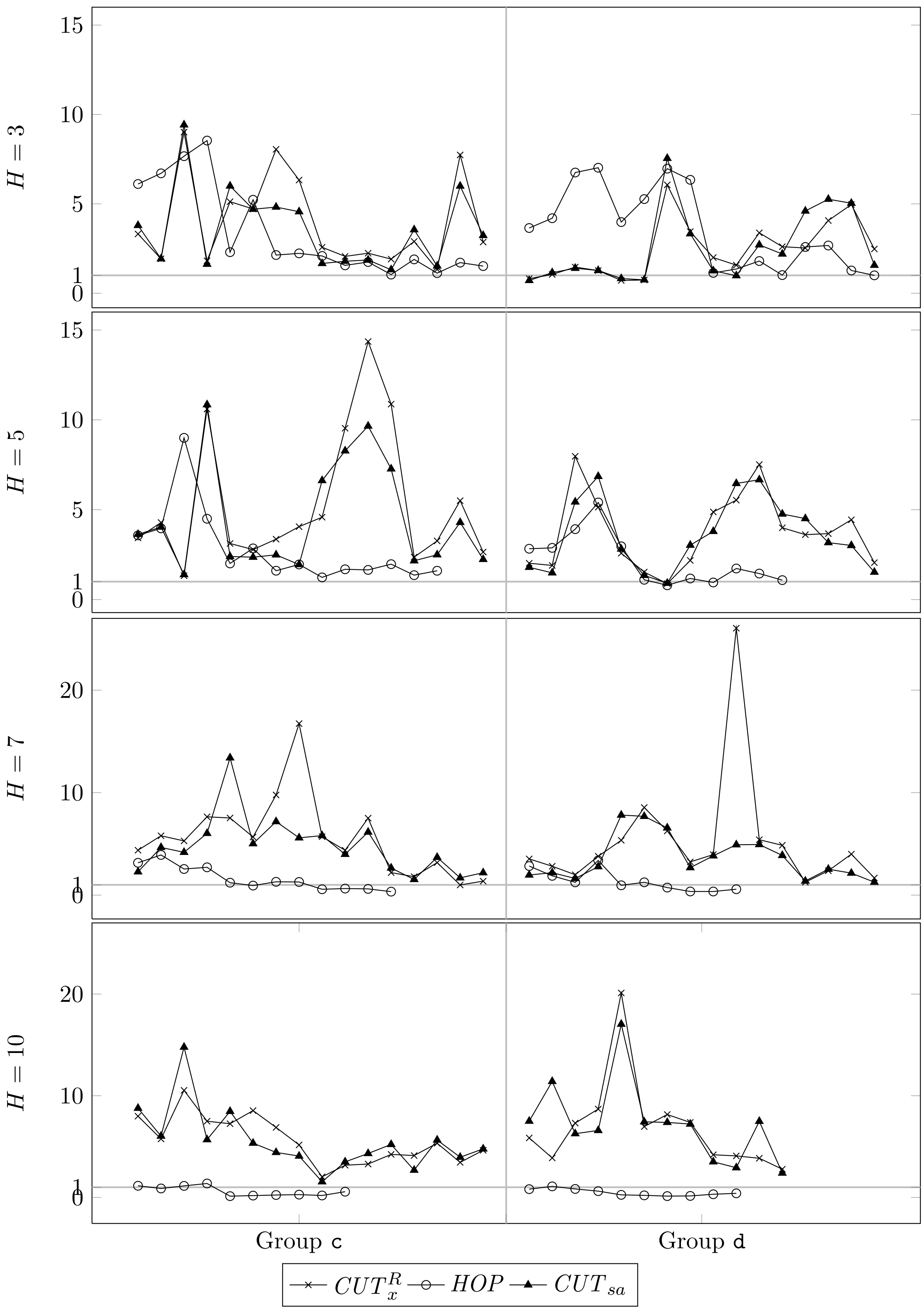

Figure 7: Speedup/slowdown factors of total runtime of models CUT_x^R, HOP and CUT_{sa} compared to CUT_x^F

shown are averages over the set of these instances, calculated separately for each value of H. To avoid dominance of either the harder or the easy instances in the results, we calculate arithmetic means μ_a and shifted geometric means μ_s for the shift $s \in \{1, 10, 100, 1000\}$. For non-negative values $v_i, i \in \{1, \ldots, k\}$ the shifted geometric mean for $s \geq 1$ is defined as

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

(cf. Achterberg (2009)). 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 2 or more approaches can perform equally well on one instance, thus the values for *#Best* do not necessarily add up to $Inst^{OPT}$. 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 is the only model, that solves the remaining 8 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 doesn't show any pattern. We recall that we used the default branching strategy of CPLEX enhanced by higher priorities associated to facilities.

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.

Looking at the difference between the overall and the separation time we observe that CUT_x^R is sometimes slower, which can be explained by the size of the underlying LP.

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

			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
		μ_s, $s=1$	8.3	**2**	**13.4**	**7.5**	37	33.8	8.0	32	31.5	7.7	33.3
		μ_s, $s=10$	12.4	**4**	**18.8**	11.2	64	45.4	12.0	56	42.1	**11.2**	36.5
		μ_s, $s=100$	16.9	**6**	**29.0**	14.8	103	73.6	15.8	88	66.9	**14.1**	46.0
		μ_s, $s=1000$	18.7	**6**	**34.8**	15.9	131	103.6	17.0	107	91.4	**15.0**	53.0
5	26	μ_a	19.0	**55**	**57.5**	19.0	371	291.1	**18.0**	317	241.0	19.0	90.0
		μ_s, $s=1$	**14.9**	**18**	**30.0**	15.4	243	112.4	15.3	215	102.1	15.0	61.0
		μ_s, $s=10$	**16.6**	**27**	**35.7**	17.2	267	125.0	17.0	236	113.7	16.7	64.1
		μ_s, $s=100$	18.7	**43**	**48.5**	19.2	301	171.4	**18.4**	265	154.2	18.5	75.9
		μ_s, $s=1000$	19.3	**54**	**56.0**	19.7	348	245.9	**18.8**	302	211.9	19.1	87.1
7	22	μ_a	30.0	**216**	164.7	24.0	722	992.9	**21.0**	584	726.8	26.0	**121.2**
		μ_s, $s=1$	24.4	**115**	**82.5**	21.2	642	450.9	**17.6**	517	354.9	22.8	92.8
		μ_s, $s=10$	25.9	**124**	**89.8**	22.3	643	464.5	**18.8**	518	370.9	23.9	95.3
		μ_s, $s=100$	28.9	**159**	115.6	24.1	651	545.4	**20.8**	529	449.0	26.0	**106.2**
		μ_s, $s=1000$	30.3	**201**	149.8	24.8	687	754.6	**21.5**	562	604.2	26.8	**118.0**
10	20	μ_a	37.0	**509**	580.9	**26.0**	1220	4128.4	28.0	1064	3730.8	55.0	**176.4**
		μ_s, $s=1$	31.3	**469**	367.2	**21.7**	1081	2321.4	23.7	970	2228.8	39.5	**148.5**
		μ_s, $s=10$	32.6	**470**	372.3	**22.8**	1083	2328.8	25.0	971	2233.5	41.8	**150.0**
		μ_s, $s=100$	35.8	**476**	407.0	**25.1**	1093	2395.6	27.3	979	2277.3	48.8	**158.8**
		μ_s, $s=1000$	37.5	**496**	499.5	**26.1**	1147	2770.8	28.2	1017	2557.9	54.0	**172.0**
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**

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.

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 values for $|V_x|$ and $|A_{S_x}|$ reported are 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 for that the layered graph is sometimes much smaller than G is the sparsity of the core

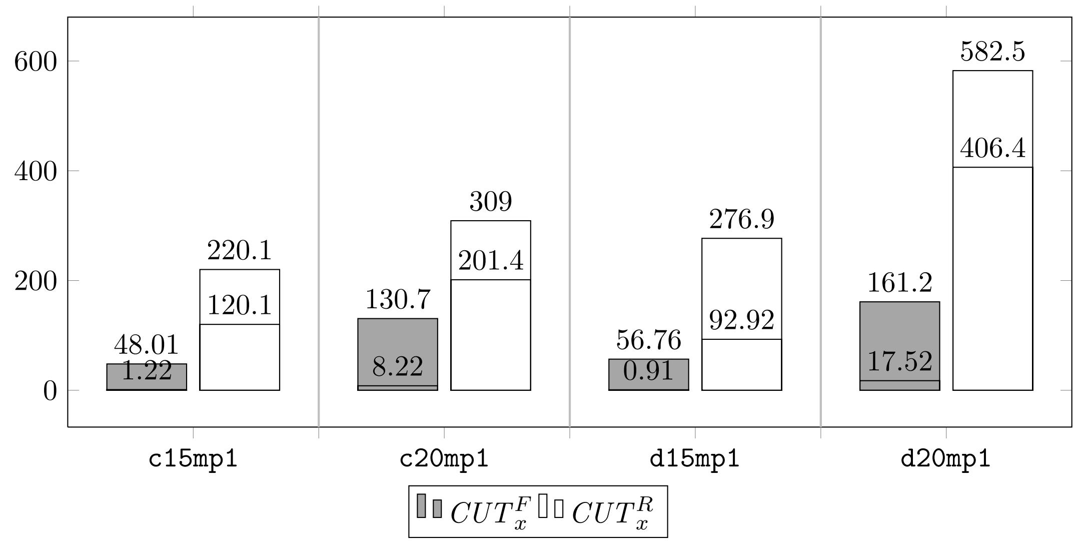

Figure 8: Comparison of the time spent for separation and the total running time for selected instances and $H = 5$.

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 (2011)). As the density of the graph or the value of H increase, 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.

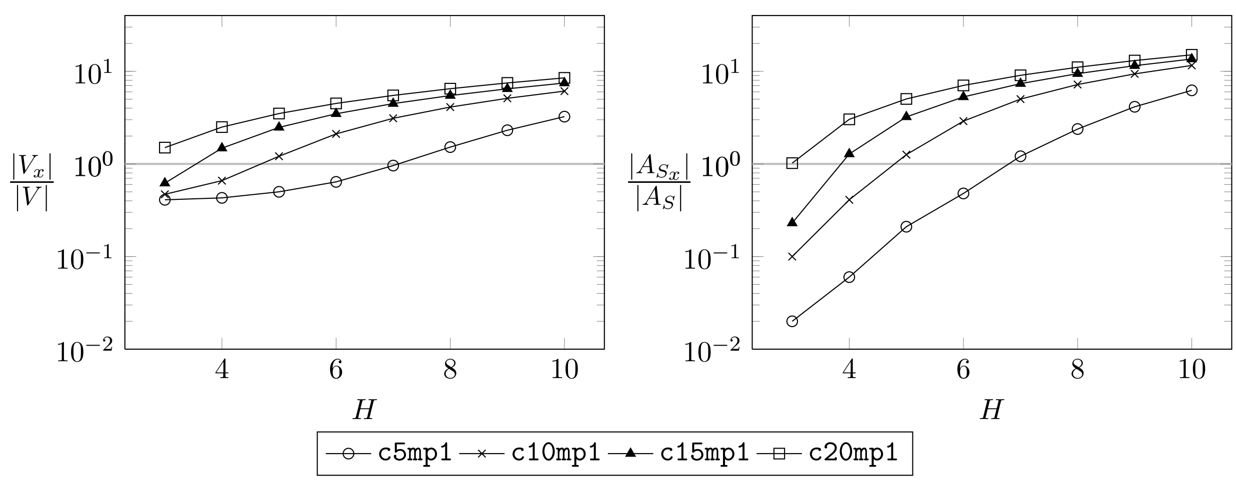

Figure 9: Size of V_x and A_{S_x} compared to size of V and A, respectively.

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. (2003) 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. Kompella et al. (1993)).

There are recent contributions on layered graph approaches to both of these variants of the Minimum Spanning and Steiner tree problem. Ruthmair and Raidl (2011) extend known layered graph models for the Delay constrained Minimum Spanning and Steiner tree problem. 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. (2009) 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 (diameter even) and an additional artificial layer (diameter odd).

We implemented the ideas presented in Gouveia et al. (2009) 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 the ones considered in Gouveia et al. (2009) (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:

- Layered graph obtained by the transformation suggested in Gouveia et al. (2009) 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 and / or adaptive layered graph frameworks.

7. Conclusions

For the time being, the strongest MIP models for the hop constrained minimum spanning tree problem are obtained on layered graphs (see Gouveia, Simonetti, and Uchoa (2009)). 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 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. 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'd like to thank to the area and associated editor as well as to 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 Hertha-Firnberg Fellowship of the Austrian Science Fund (FWF).

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

References

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

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

Balakrishnan, A., K. Altinkemer. 1992. Using a hop-constrained model to generate alternative communication network design. *INFORMS J. on Comput.* **4** 192–205.

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

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

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

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

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** 141–159.

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.

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

Gollowitzer, S. 2010. MIP models for (hop-constrained) connected facility location. Master's thesis, Vienna University of Technology.

Gollowitzer, S., I. Ljubić. 2011. MIP models for connected facility location: A theoretical and computational study. *Comput. & Oper. Res.* **38** 435–449.

Gouveia, L. 1996. Multicommodity flow models for spanning trees with hop constraints. *Eur. J. of Oper. Res.* **95** 178–190.

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

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.

Gouveia, L., L. Simonetti, E. Uchoa. 2009. Modeling hop-constrained and diameter-constrained minimum spanning tree problems as steiner tree problems over layered graphs. *Math. Programming* 1–26.

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

Kompella, V. P., G. C. Polyzos, J. C. Pasquale. 1993. Multicast routing for multimedia communication. *IEEE/ACM Trans. on Networking* **1** 286–292.

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

Krick, C., H. Räcke, M. Westermann. 2003. Approximation algorithms for data management in networks. *Theory of Comput. Systems* **36** 497–519.

Leitner, Markus, Günther R Raidl. 2011. Branch-and-cut-and-price for capacitated connected facility location. *J. of Math. Model. and Algorithms* To appear.

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.

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

Mahdian, M., Y. Ye, J. Zhang. 2006. Approximation algorithms for metric facility location problems. *SIAM J. on Comput.* **36** 411–432.

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.

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

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

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.

Ruthmair, M., G. R. Raidl. 2011. A layered graph model and an adaptive layers framework to solve delay-constrained minimum tree problems. *Proceedings of the 15th Conference on Integer Programming and Combinatorial Optimization (IPCO XV)*. To appear.

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

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

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 (1996) or Manyem (2009)). 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. (2006) 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_j^q the optimal sub-tree rooted at (j,q). We transform the solution as follows: a) We move the core arcs in T_j^q up by $q-p$ levels, such that the obtained tree is then rooted in (j,p). We then refer to it as T_j^p. b) For customers assigned to open facilities (i,l), $q \leq l \leq H$ in T_j^q, 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 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. 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 $CUT^{F+}_{x,z}$ is at least as strong as $CUT^{F}_{x,z}$. The strict relation holds because of the example in Figure 10. To show an instance for which $\frac{\upsilon_{LP}(CUT^{F+}_{x,z})}{\upsilon_{LP}(CUT^{F}_{x,z})} \approx H - 1$ 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 $p = 2, \ldots, H$ 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. □

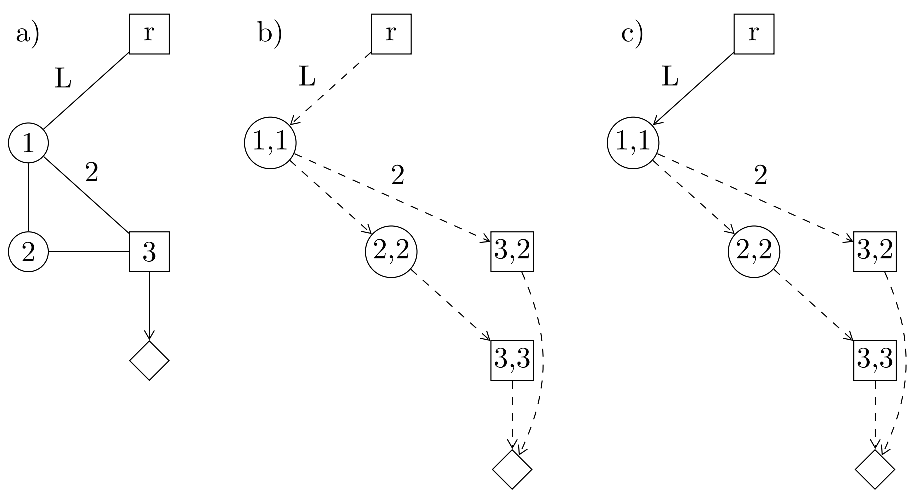

Figure 10: a) Instance on G with $H = 3$; b) LP optimal solution for $CUT^{F}_{x,z}$. Dashed and solid arcs take LP-values equal to 1/2 and 1, respectively. $\upsilon_{LP}(CUT^{F}_{x,z}) = L/2 + 4$; c) LP optimal solution for $CUT^{F+}_{x,z}$ with cost $L + 4$.

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:

$\upsilon_{LP}(CUT_x^F) \geq \upsilon_{LP}(HOP)$**:** It is enough to show that an optimal LP-solution of the formulation CUT_x^F 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_x^F formulation is given below. It comprises additional flow variables f_{ij}^{kp}, for all $ij \in A_S$, and $k \in F \setminus \{r\}$, $p = 1, \ldots, H$, 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:

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

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

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

$$0 \leq f_{ij}^{kp} \leq X_{ij}^p \forall ij \in A_S,\ k \in F \setminus \{r\},\ p = 1,\ldots,H \tag{26}$$

$$(11) - (17)$$

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_{jl}^p, $jl \in A_S$, $p = 1, \ldots, H$, there exists a commodity $k \in F \setminus \{r\}$ such that constraint (26) is tight, i.e. $X_{jl}^p = f_{jl}^{kp}$. 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 $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$.

$\upsilon_{LP}(CUT_x^F) > \upsilon_{LP}(HOP)$**:** 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.

□

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

$\upsilon_{LP}(CUT_{x,z}^R) \geq \upsilon_{LP}(CUT_x^R)$**:** Let $(\mathbf{X}, \mathbf{Z})$ be an optimal LP-solution of the model $CUT_{x,z}^R$. 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_x^R.

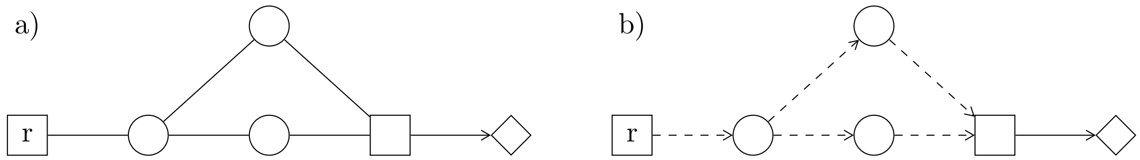

Figure 11: a) Instance G with $H = 3$. b) An optimal LP-solution for HOP_F in which dashed arcs take value 1/2.

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^F_x, CUT^R_x 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}).

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	$\|F_0\|$	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 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	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 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	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 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.

			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	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-