Layered Graph Approaches to the Hop Constrained Connected Facility Location Problem

Ivana Ljubić, Stefan Gollowitzer
Department of Statistics and Decision Support Systems, Faculty of Business, Economics, and Statistics, University of Vienna, Austria, {ivana.ljubic@univie.ac.at, 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 the root and an 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. 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.

Key words: Hop constrained Minimum Spanning trees; Hop constrained Steiner trees; Connected Facility Location; Mixed Integer Programming Models; LP-relaxations
History: Submitted August 12, 2010

---

1. Introduction

Connected Facility Location (ConFL) models data distribution and management problems in a network setting that arises in information/content distribution networks (see, e.g. Krick et al. (2003)). In these applications, there are facilities (e.g., servers) to be located on a network that will cache information. Demand nodes make requests for the information. Each demand node is served from the closest open facility. 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 via a Steiner tree.

A similar problem appears in the design of the last mile telecommunication networks. In (Gollowitzer and Ljubić, 2010) we have shown that the Fiber-to-the-Curb strategy is modeled by the Connected Facility Location problem (ConFL) as follows: Fiber optic cables run from a central office to a cabinet serving a neighborhood. End users connect to this cabinet using the existing copper connections. Expensive switching devices are installed in these cabinets. The problem is to minimize the costs by determining positions of cabinets, deciding which customers to connect to them, and how to reconnect cabinets among each other and to the central office (i.e., to the backbone) via a Steiner tree.

If connection costs are non-negative, ConFL solutions obey 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, data distribution 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 Assuming that a root facility is given and it needs to be open in any feasible solution, ConFL can 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*, $S \subset V$ the set of possible *Steiner nodes*, $F \subseteq S$ the set of *facilities*, and the *root* node $r \in F$. We are also given edge costs $c_e \geq 0, e \in E$ 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: 1) each customer is assigned to the closest open facility, 2) a Steiner tree connects all open facilities, and 3) the sum of assignment, facility opening and Steiner tree costs is minimized.

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 solution that is valid for ConFL and in which there are at most H hops between the root and any open facility.

Observation 1. *Using the transformation given in Gollowitzer and Ljubić (2010), 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.*

Our Contribution We first show that HC ConFL is an NP-hard optimization problem that does not belong to APX, i.e., it is not possible to have polynomial time heuristics that guarantee a constant approximation ratio. By extending the ideas given by Gouveia et al. (2010) we then propose two possibilities for modeling the hop constrained ConFL: i) as directed (asymmetric) ConFL on layered graphs, or ii) as the Steiner arborescence (SA) 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.*

Proof. This result can be obtained by applying an error-preserving polynomial reduction from SET COVER. Any SET COVER instance can be reduced into a hop constrained 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 by defining each terminal i to be a potential facility in HC ConFL and introducing a customer node c_i for every single facility 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 $P = NP$. □

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.*

Connected Facility Location Early work on ConFL mainly includes approximation algorithms. The problem can be approximated within a constant ratio and 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 Gollowitzer and Ljubić (2010) 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. 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. (2010) 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 edge e between nodes i and j from S by two directed arcs ij and ji and each 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 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 (2010)), flow-based MIP formulations are of limited usage if they are simply plugged in into a MIP solver without any usage of 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 make an explicit usage of $\mathbf{x}$ and $\mathbf{z}$ variables. They are rather provided in a lifted space of layered graphs, and the values of their variables are projected back into the space of $(\mathbf{x}, \mathbf{z})$ variables.

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 graphs*, 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 $D \subseteq A$. In the examples described in the following sections we will use the following symbols: ■ represents the root node, ∘ represents a Steiner node. $\square^l$ represents a facility with label l. ⋆ represents a customer. In these examples the default 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.

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 the models on 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:

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

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

A node (i, p) will also be referred to as a "node i at level p".

Preprocessing Observe that a node $(i, p) \in S_{x,z}$ whose in-degree is zero, can be removed from $LG_{x,z}$. Similarly, a Steiner node $(i, p) \in S_{x,z} \setminus F_{x,z}$ whose out-degree is zero, cannot contribute to any optimal solution. The removal of those redundant nodes is performed iteratively:

- Nodes with in-degree zero are removed starting from level 1 to H.
- Nodes with out-degree zero are removed starting from level $H - 1$ to 1.

Finally, we observe that, 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 $p = 1, \ldots, H$.

Figure 1 illustrates the layered graph $LG_{x,z} = (V_{x,z}, A_{x,z})$: Figure 1a) shows an original HC ConFL instance $G = (V, A)$ with $H = 3$; Figure 1b) represents the complete layered graph $LG_{x,z} = (V_{x,z}, A_{x,z})$; Figure 1c) shows the layered graph after the preprocessing; An optimal solution on $LG_{x,z}$ is given in Figure 1e), and its projection back onto the original graph $G = (V, A)$ is given in Figure 1f).

Lemma 2. *There always exists an optimal solution of directed ConFL on the layered graph $LG_{x,z}$ such that*

$$\sum_{p=1}^{H} \textit{in-degree}\{(i, p)\} \leq 1 \quad \forall i \in F \setminus \{r\} \tag{1}$$

and

$$\sum_{p=1}^{H-1} \textit{in-degree}\{(i, p)\} \leq 1 \quad \forall i \in S \setminus F. \tag{2}$$

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. □

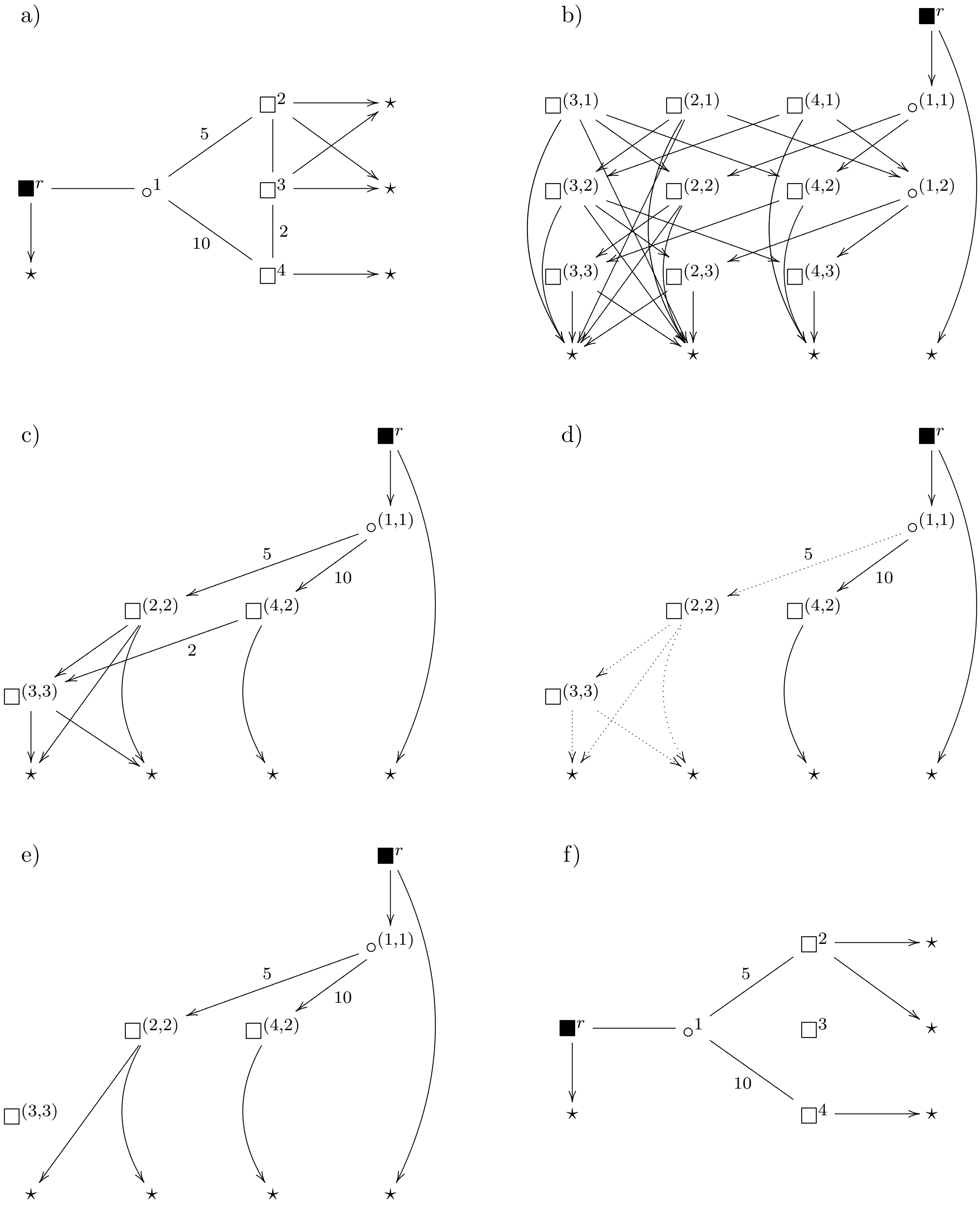

Figure 1: a) Original instance; b) $LG_{x,z}$ before, and c) after preprocessing; d) An optimal LP-solution for $LG_{x,z}CUT_F$ – dotted and solid arcs take LP-value of 1/2 and 1, respectively; e) An optimal LP-solution for LG_xCUT_R which is already MIP-optimal; f) Projection of solution in e) back onto the original graph.

Lemma 3. *Given the graph transformation from G to $LG_{x,z}$ described above, any* optimal *solution of the directed ConFL on $LG_{x,z}$ 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 directed ConFL solution on $LG_{x,z}$.*

Proof. Consider an optimal ConFL solution on $LG_{x,z}$. If it does not satisfy properties (1) and (2), we construct a solution of equal cost by performing the transformation given in the proof of Lemma 2.

Ignoring the second index on the nodes of that solution, we obtain a feasible HC ConFL solution (i.e., a ConFL solution with at most H hops in G). Because of Lemma 2, at most one copy of each facility is opened in the considered solution. Thus, the assignment and facility opening costs are the same for both solutions. To show that the Steiner tree costs are the same, assume that there exists an optimal solution on $LG_{x,z}$ whose cost is strictly greater than the cost of the optimal hop constrained solution on G. In that case, the ConFL solution on $LG_{x,z}$ projected back onto G either contains a cycle or uses the same edge twice, which again contradicts Lemma 2.

It is not difficult to see that every hop constrained ConFL solution on G corresponds to a ConFL solution with the same cost in the layered graph. Figures 1e) and 1f) illustrate such a pair of solutions. $\square$

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 $\mathbf{X}$ variables in the cut $\delta^-(W)$ in $LG_{x,z}$ defined by $W \subseteq V_{x,z} \setminus \{r\}$. In Gollowitzer and Ljubić (2010) we describe two cut-set based formulations for the (directed) ConFL problem. The models differ in the way they make use of the connectivity concept. In the first one, 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 $LG_{x,z}CUT_F$ and $LG_{x,z}CUT_R$. 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 (7) below).

Connectivity Cuts Between Root and Facilities The model $LG_{x,z}CUT_F$ reads as follows:

$$
\begin{aligned}
(LG_{x,z}CUT_F) \quad & \min \sum_{ij \in A} c_{ij} \sum_{p=1}^{H} X^p_{ij} + \sum_{i \in F \setminus \{r\}} f_i \sum_{p=1}^{H} Z^p_i + f_r z_r \\
X[\delta^-(W)] &\geq Z^p_i \qquad \forall W \subseteq S_{x,z} \setminus \{r\},\ (i,p) \in F_{x,z} \cap W && (3) \\
\sum_{jk \in A_R} \sum_{p=1}^{H} X^p_{jk} &= 1 \qquad \forall k \in R && (4) \\
X^p_{jk} &\leq Z^p_j \qquad \forall jk \in A_R,\ p = 1, \dots, H,\ j \neq r && (5) \\
z_r &= 1 && (6) \\
X^p_{ij} &= 0 \qquad ij \in A, \begin{cases} i = r,\ p = 2, \dots, H \\ i \neq r,\ p = 1 \\ j \in S \setminus F,\ p = H \end{cases} && (7) \\
X^p_{ij} &\in \{0,1\} \qquad \forall ij \in A, p = 1, \dots, H && (8) \\
Z^p_i &\in \{0,1\} \qquad \forall (i,p) \in F_{x,z} && (9)
\end{aligned}
$$

Constraints (3) 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 (4) are *assignment constraints*. They ensure that each customer $k \in R$ is assigned to exactly one facility from $F_{x,z} \cup \{r\}$. Inequalities (5) are *coupling constraints* — they necessitate a facility j at a level p to be open if a customer is assigned to it. Equation (6) 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 $ij \in A_R$, every optimal solution on $LG_{x,z}$ 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 2 and from the fact that for each $i \in F$, $Z^p_i \leq$ in-degree$\{(i,p)\}$, for all $p = 1, \dots, H$. Consequently, we can show the following

Lemma 4. *In the model $LG_{x,z}CUT_F$, connectivity cuts* (3) *can be replaced by the following stronger ones:*

$$X[\delta^-(W)] \geq \sum_{p=1}^{H} Z_i^p \qquad \forall W \subseteq S_{x,z} \setminus \{r\},\ i \in F \setminus \{r\} \tag{10}$$

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

The new MIP formulation, in which (3) is replaced by (10) will be denoted by $LG_{x,z}CUT_F^+$.

Connectivity Cuts Between Root and Customers By replacing (3) and (4) in the model $LG_{x,z}CUT_F$ with the following inequalities,

$$X[\delta^-(W)] \geq 1 \qquad \forall W \subseteq V_{x,z} \setminus \{r\}, W \cap R \neq \emptyset, \tag{11}$$

we obtain a new model that we denote by $LG_{x,z}CUT_R$.

Inequalities (11) are connectivity cuts on $LG_{x,z}$ between sets containing the root and a customer respectively. Our study on ConFL in Gollowitzer and Ljubić (2010) 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 (2010), it has been shown that cut-set based MIP models on layered graphs represent the tightest formulations 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 hop constrained 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

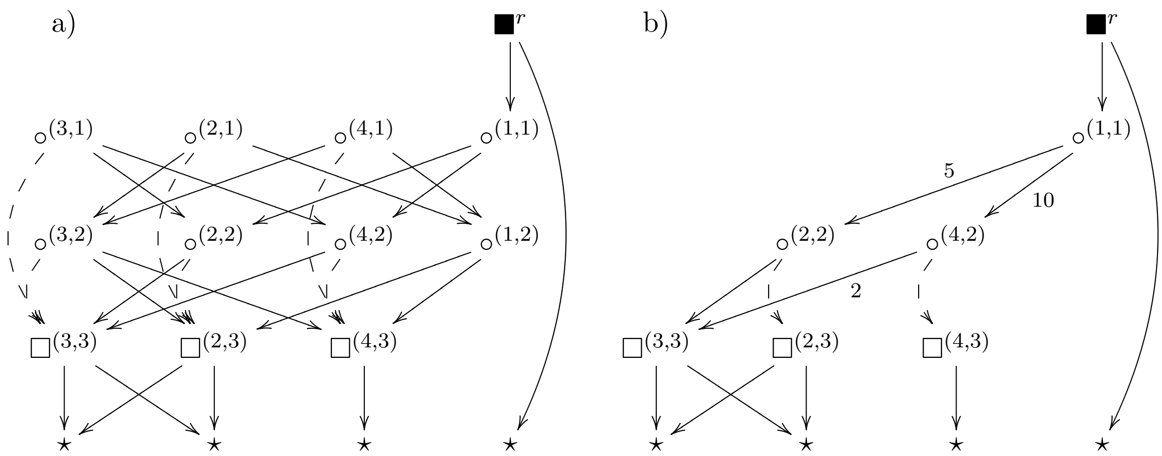

Figure 2: Layered graph LG_x for an instance given in Figure 1a) obtained a) before and b) after preprocessing.

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:

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

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

One observes that the same preprocessing rules explained for $LG_{x,z}$ also apply to LG_x. Furthermore, Lemma 2 applies as well. Hence, we can show the following:

Lemma 5. *Given the graph transformation from G to LG_x described above, any* optimal *solution of the directed ConFL on LG_x 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 directed ConFL solution on LG_x.*

Figure 2 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:

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

$$X_x[\delta^-(W)] \geq z_i \qquad \forall W \subseteq S_x \setminus \{r\},\ W \cap F_x \neq \emptyset \qquad (12)$$

$$\sum_{jk \in A_R} x_{jk} = 1 \qquad \forall k \in R \qquad (13)$$

$$x_{jk} \leq z_j \qquad \forall jk \in A_R \qquad (14)$$

$$X^p_{ij} = 0 \qquad ij \in A_S, \begin{cases} i = r,\ p = 2,\dots,H \\ i \neq r,\ p = 1 \end{cases} \qquad (15)$$

$$X^p_{ij} \in \{0,1\} \qquad ij \in A_S, p = 1,\dots,H \qquad (16)$$

$$z_i \in \{0,1\} \qquad \forall i \in F \setminus \{r\} \qquad (17)$$

$$x_{jk} \in \{0,1\} \qquad \forall jk \in A_R \qquad (18)$$

(6)

Constraints (12) are connectivity cuts on LG_x between sets containing the root and a facility i respectively. Equations (13) are the assignment constraints, and inequalities (14) are the coupling constraints.

Similarly, if we now replace constraints (12) and (13) by the following ones, we obtain a stronger formulation that we denote by LG_xCUT_R:

$$X_x[\delta^-(W)] \geq 1 \quad \forall W \subseteq V_x \setminus \{r\}, W \cap R \neq \emptyset \tag{19}$$

One observes that, if constraints (18) are relaxed to $x_{jk} \geq 0$, for all $jk \in A_R$, the optimal solution remains integral. Although constraints (13) 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 (2010)). 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 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 hop constrained ConFL problem as the Steiner arborescence on even more 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 (2010)).

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 now known in advance whether the opening costs of a potential facility node are going to be paid, or it will be used only as a Steiner node. However, looking at LG_x, one observes that in any optimal solution of the directed ConFL on LG_x, the only Steiner nodes that are taken into an optimal solution are at levels $1, \ldots, H-1$. In other words, if a facility node (i, H) belongs to an optimal solution, it serves only to connect the root with a customer, i.e., every node (i, H) that belongs to an optimal solution is an open facility. Because 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_{STP}.

Lemma 6. *Every optimal solution of the Steiner arborescence on LG_{STP} 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_{STP}.*

The corresponding MIP model reads then as follows:

$$(LG_{STP}CUT)\ \min \sum_{ij \in A_S} c_{ij} \sum_{p=1}^{H-1} X_{ij}^p + \sum_{jk \in A_R} c_{jk} x_{jk} + \sum_{i \in F} f_i \sum_{p=1}^{H-1} X_{ii}^p + \sum_{ij \in A_S, j \in F} (c_{ij} + f_j) X_{ij}^H + f_r$$
$$(13), (15), (16), (18), (19)$$

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 $LG_{STP}CUT$.

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 graph 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 graph 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:

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

$$\sum_{\substack{i \in S \setminus \{k\}: \\ ij \in A_S}} X_{ij}^{p-1} \geq X_{jk}^p \qquad \forall jk \in A_S,\ j \neq r,\ p = 2, \ldots, H \tag{20}$$

$$\sum_{ij \in A_S} \sum_{p=1}^{H} X_{ij}^p \geq z_j \qquad \forall j \in F \setminus \{r\} \tag{21}$$

$$(6), (13) - (18)$$

Constraints (20) are connectivity constraints given in a compact way — comparing HOP_F with the model $LG_x CUT_F$, we observe that the former one is obtained by replacing constraints (12) by (20) and (21). Constraints (20) 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 (21) 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_{STP}, one could replace inequalities in (21) 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_F, 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 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.

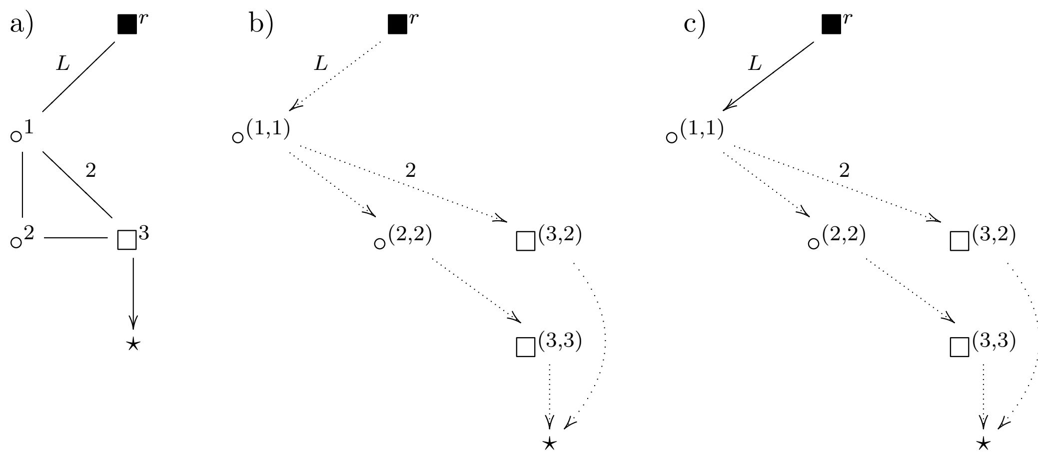

Figure 3: a) Instance on G with $H = 3$; b) LP optimal solution for $LG_{x,z}CUT_F$. Dotted and solid arcs take LP-values equal to 1/2 and 1, respectively. $\upsilon_{LP}(LG_{x,z}CUT_F) = L/2 + 4$; c) LP optimal solution for $LG_{x,z}CUT_F^+$ with cost $L + 4$.

Lemma 7. *Formulation $LG_{x,z}CUT_F^+$ is strictly stronger than formulation $LG_{x,z}CUT_F$. Furthermore, there exist HC ConFL instances for which $\frac{\upsilon_{LP}(LG_{x,z}CUT_F^+)}{\upsilon_{LP}(LG_{x,z}CUT_F)} \approx H - 1$.*

Proof. Constraints (10) dominate constraints (3). Thus, formulation $LG_{x,z}CUT_F^+$ is at least as strong as $LG_{x,z}CUT_F$. The strict relation holds because of the example in Figure 3. To show an instance for which $\frac{\upsilon_{LP}(LG_{x,z}CUT_F^+)}{\upsilon_{LP}(LG_{x,z}CUT_F)} \approx H - 1$ holds, we generalize the above example. The subgraph induced by nodes $\{1, 2, 3\}$ is replaced by the subgraph containing nodes $\{1, \ldots, H - 1\}$ being the Steiner nodes and a node H, being the facility node. This subgraph is connected as follows: Node H is connected to all $i = 1, \ldots, H - 1$ with an edge of cost $c_{iH} = H - i$. For each $i = 1, \ldots, H - 1$, node i is connected to $i + 1$ with an edge of cost $c_{i,i+1} = 1$. In the LP-relaxation of the model $LG_{x,z}CUT_F$, all facilities (H, p) at levels $p = 2, \ldots, H$ will be open with $Z_H^p = 1/(H - 1)$, and consequently, $X_{r1}^1 = 1/(H - 1)$, so that $\upsilon_{LP}(LG_{x,z}CUT_F) \approx L/(H - 1)$. In contrast, the optimal LP-value of the model $LG_{x,z}CUT_F^+$ is $\upsilon_{LP}(LG_{x,z}CUT_F^+) \approx L$, which proves the claim. □

Lemma 8. *The formulation LG_xCUT_F is strictly stronger than the formulation HOP_F.*

Proof. $\upsilon_{LP}(LG_xCUT_F) \geq \upsilon_{LP}(HOP_F)$**:** It is enough to show that an optimal LP-solution of the formulation LG_xCUT_F is also feasible for the model HOP_F. For that purpose we will use the max-flow min-cut theorem. A flow formulation on the graph G which is equivalent to the LG_xCUT_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

Figure 4: a) Instance G with $H = 3$. b) An optimal LP-solution for HOP_F in which dotted arcs take value 1/2.

the p-th position from the root node. We denote this formulation by MCF_F:

$$\sum_{ji \in A_S} f_{ji}^{k,p-1} - \sum_{ij \in A_S} f_{ij}^{kp} = 0 \quad \forall k \in F \setminus \{r\},\ i \in S \setminus \{r,k\},\ p = 2,\dots,H \tag{22}$$

$$\sum_{rj \in A_S} f_{rj}^{k1} = z_k \ \ \forall k \in F \setminus \{r\} \tag{23}$$

$$\sum_{p=1}^{H} \sum_{jk \in A_S} f_{jk}^{kp} = z_k \ \ \forall k \in F \setminus \{r\} \tag{24}$$

$$0 \le f_{ij}^{kp} \le X_{ij}^{p} \forall ij \in A_S,\ k \in F \setminus \{r\},\ p = 1,\dots,H \tag{25}$$

$$(6), (13) - (18)$$

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 $(\mathbf{X}', \mathbf{x}', \mathbf{z}') \in \mathcal{P}_{HOP_F}$. Constraints (21) are directly implied by inequalities (22)-(25). To show that constraints (20) are also satisfied, we first observe that, for every X_{jl}^{p}, $jl \in A_S$, $p = 1,\dots,H$, there exists a commodity $k \in F \setminus \{r\}$ such that constraint (25) is tight, i.e. $X_{jl}^{p} = f_{jl}^{kp}$. From the flow conservation constraints (22)-(24), it follows:

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

and thus, inequalities (20) hold for $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$.

$v_{LP}(LG_x CUT_F) > v_{LP}(HOP_F)$**:** Consider an example given in Figure 4. LP-solution for HOP_F shown in Figure 4b) is not feasible for $LG_x CUT_F$ and the strict inequality regarding the LP-values holds.

□

Lemma 9. *The following results hold:*

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

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

Proof. The result given in Gollowitzer and Ljubić (2010) 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. □

Proposition 1. *Formulations* $LG_{x,z}CUT_R$ *and* LG_xCUT_R *are equivalent.*

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

$v_{LP}(LG_{x,z}CUT_R) \geq v_{LP}(LG_xCUT_R)$**:** Let $(\mathbf{X}, \mathbf{Z})$ be an optimal LP-solution of the model $LG_{x,z}CUT_R$. We project $(\mathbf{X}, \mathbf{Z})$ into a solution $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$ and show that it is feasible for the model LG_xCUT_R. 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 (14)-(15) are satisfied, it only remains to show that $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$ satisfies (19). Denote by $\delta^-(W)_{|D} = \{ij \in \delta^-(W) \mid ij \in D\}$. Then, $X_x[\delta^-(W)] = X_x[\delta^-(W)_{|\cup_{i=1}^4 A_i}] + X_x[\delta^-(W)_{|A_6 \cup A_7}] = X[\delta^-(W)_{|\cup_{i=1}^4 A_i}] + X_x[\delta^-(W)_{|A_6 \cup A_7}] \geq X[\delta^-(W)_{|\cup_{i=1}^4 A_i}] + X[\delta^-(W)_{|A_5}] = X[\delta^-(W)] \geq 1$.

$v_{LP}(LG_xCUT_R) \geq v_{LP}(LG_{x,z}CUT_R)$**:** Let $(\mathbf{X}', \mathbf{x}', \mathbf{z}')$ be an optimal LP-solution of the model LG_xCUT_R. 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_6. Furthermore, we set $Z^p_j := X'^p_{jj}$, for all arcs from A_4, 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 (4)-(8). To show that constraints (11) 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.

□

5. Computations

In this section we present a computational comparison of the MIP models for solving HC ConFL given above. According to Proposition 1 and the theoretical analysis given in the previous section, transformations of G into $LG_{x,z}$ and LG_x provide two strongest MIP formulations with the same quality of lower bounds. Therefore, we concentrate on models derived from the layered graph LG_x, which comprises smaller number of edges and facilities. The computational comparison is conducted on three branch-and-cut (B&C) algorithms derived for MIP models with the exponential number of variables, and on one compact model, HOP_F (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: LG_xCUT_F, LG_xCUT_R and $LG_{STP}CUT$. B&C algorithms for ConFL are described in detail in our recent computational study on ConFL given in Gollowitzer and Ljubić (2010). The most important non-standard ingredients of this 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.

Initialization Each branch-and-cut algorithm is initialized with the assignment and coupling constraints, (13) and (14), 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 (12) and (19) 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 (12), the maximum flow is calculated between the root node and any facility i, such that $z_i > 0$. Inequalities (19) are separated by calculating the flow between the root and any customer j, $j \in R$.

Branching Among all binary variables, the biggest influences on the structure of the solution is due to facility variables z_i. Therefore, in our default branch-and-bound implementation, the highest branching priority is assigned to facility variables z_i, $i \in F$.

The remaining details of our implementation can be found in Gollowitzer and Ljubić (2010).

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 UFLP instance with an STP instance, to generate ConFL input graphs in the following way: Nodes indexed by $1, \dots, |F|$ 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 additional Steiner nodes.

- We consider a set of non-trivial UFLP instances from UflLib (see `http://www.mpi-inf.mpg.de/departments/d1/projects/benchmarks/UflLib/`): `mp-`{`1,2`} and `mq-`{`1,2`} instances 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 of 2 classes MP and MQ of sizes 200×200 and 300×300, respectively.

- STP instances: Instances {`c,d`}`n`, for $n \in \{5, 10, 15, 20\}$ were chosen randomly from the OR-library (see `http://people.brunel.ac.uk/~mastjjb/jeb/orlib/steininfo.html`) 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 numer of nodes in the core graph ($|\tilde{S} \cup F|$); 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.

Table 1: Basic properties of benchmark instances.

STP	UFLP	$\|R\|$	$\|F\|$	$\|\tilde{S} \cup F\|$	$\|E_S\|$	$\|E_R\|$
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

5.3. Comparison of Three Formulations

In the first step of our computational study we compare the performance of three proposed formulations: the compact formulation HOP_F and two cut set based formulations LG_xCUT_F and LG_xCUT_R. Tables 2 to 5 show the results of this experiment. For all reported results, the default time limit was set to 3600 seconds.

The first column shows the name of the instance; column OPT provides the value of the optimal solution; columns BB show the number of nodes in the branch-and-bound tree; columns $\#Iter$ give the number of iterations; in columns t [s] we show the CPU time (in seconds) needed to solve the instance. For formulations LG_xCUT_R and HOP_F we also provide two gap values: if the solver does not find the optimal solution within the given time limit, it terminates with a feasible solution, providing an upper bound (UB) and a global lower bound (LB). The percentage gap between this upper and lower bound, calculated as $(UB - LB)/UB$ is given in columns denoted by g. In columns denoted by g_{opt}, we show the percentage gap between the optimal solution (OPT, usually determined by running the model LG_xCUT_F) and the corresponding lower bound, calculated as $(OPT - LB)/OPT$.

Comparing these three models, we observe the following: the best performing model overall is LG_xCUT_F, which solves all the instances to optimality for $H \in \{3, 5, 7, 10\}$, except the four largest ones for $H = 10$. The average running time over all 32 instances for LG_xCUT_F increases from 25.9 seconds ($H = 3$) to 743.1 seconds ($H = 10$). We also observe that the complexity of the

Table 2: Comparison of models LG_xCUT_F, LG_xCUT_R and HOP_F with $H = 3$. The best running times are shown in bold.

	LG_xCUT_F				LG_xCUT_R					HOP				
Inst.	*OPT*	*BB*	*#Iter*	*t* [s]	*g*	g_{opt}	*BB*	*#Iter*	*t* [s]	*g*	g_{opt}	*BB*	*#Iter*	*t* [s]
c5mp1	2907.96	5	729	**1.6**	0.0	0.0	5	738	5.0	0.0	0.0	5	796	9.7
c5mp2	2912.63	3	988	**1.2**	0.0	0.0	3	988	2.4	0.0	0.0	3	1139	9.0
c5mq1	4505.04	11	2015	**2.5**	0.0	0.0	11	1939	26.1	0.0	0.0	11	2094	20.2
c5mq2	4082.42	0	1190	**2.5**	0.0	0.0	0	1190	4.7	0.0	0.0	0	1245	20.6
c10mp1	2861.05	39	15157	**7.3**	0.0	0.0	71	24508	41.9	0.0	0.0	34	13943	16.9
c10mp2	2760.27	3	5493	**2.2**	0.0	0.0	3	5889	9.6	0.0	0.0	3	5726	11.9
c10mq1	4092.96	17	23209	**13.7**	0.0	0.0	23	23141	74.8	0.0	0.0	13	17767	29.7
c10mq2	3946.52	29	22947	**15.8**	0.0	0.0	23	25142	92.0	0.0	0.0	29	23050	35.4
c15mp1	2668.48	9	17689	**9.0**	0.0	0.0	9	22761	38.2	0.0	0.0	9	20381	28.9
c15mp2	2679.63	9	20623	**12.5**	0.0	0.0	9	29522	40.7	0.0	0.0	9	21552	33.1
c15mq1	3861.57	25	48068	**32.1**	0.0	0.0	23	65601	197.4	0.0	0.0	21	46393	149.2
c15mq2	3694.56	63	65637	**50.8**	0.0	0.0	27	90172	299.6	0.0	0.0	23	58815	141.5
c20mp1	2618.66	11	25251	**20.6**	0.0	0.0	9	30989	68.4	0.0	0.0	13	26114	49.4
c20mp2	2630.46	7	20629	**20.2**	0.0	0.0	5	24935	52.5	0.0	0.0	9	20132	38.8
c20mq1	3828.50	45	81670	**84.9**	0.0	0.0	33	98490	429.1	0.0	0.0	43	76559	141.1
c20mq2	3687.49	37	86136	**156.3**	0.0	0.0	25	120643	484.7	0.0	0.0	27	82104	237.2
d5mp1	2846.01	0	605	**1.5**	0.0	0.0	0	605	1.9	0.0	0.0	0	667	10.2
d5mp2	2847.68	3	1180	**1.1**	0.0	0.0	3	1199	2.3	0.0	0.0	5	806	11.0
d5mq1	4190.20	0	2038	**2.4**	0.0	0.0	0	2038	3.6	0.0	0.0	0	2090	20.3
d5mq2	3978.17	0	2008	**3.0**	0.0	0.0	0	2008	4.1	0.0	0.0	0	2052	22.1
d10mp1	2970.53	0	779	**1.2**	0.0	0.0	0	779	1.6	0.0	0.0	0	818	10.8
d10mp2	2941.59	0	783	**1.2**	0.0	0.0	0	783	1.4	0.0	0.0	0	829	10.3
d10mq1	4212.81	7	3243	**2.9**	0.0	0.0	3	3340	18.8	0.0	0.0	3	3278	23.0
d10mq2	3979.59	3	2539	**3.0**	0.0	0.0	3	3380	11.6	0.0	0.0	3	3367	22.7
d15mp1	2805.22	123	69957	**42.8**	0.0	0.0	101	104514	117.8	0.0	0.0	75	58651	76.3
d15mp2	2692.85	11	15610	**7.3**	0.0	0.0	9	15069	29.0	0.0	0.0	11	12852	18.5
d15mq1	3890.39	19	31877	**36.0**	0.0	0.0	9	34984	107.0	0.0	0.0	17	31853	65.3
d15mq2	3788.07	25	38224	**32.4**	0.0	0.0	29	60221	209.9	0.0	0.0	27	42901	86.9
d20mp1	2621.66	11	24875	**23.0**	0.0	0.0	17	41773	125.3	0.0	0.0	13	27338	64.5
d20mp2	2632.46	5	21785	**17.9**	0.0	0.0	6	29507	73.7	0.0	0.0	9	20786	54.4
d20mq1	3830.50	49	79970	**112.2**	0.0	0.0	47	126686	572.3	0.0	0.0	39	75435	142.9
d20mq2	3687.49	38	88147	**108.0**	0.0	0.0	33	106292	421.3	0.0	0.0	31	80268	216.4
Avg.		19	25658	25.9	0.0	0.0	17	34370	111.5	0.0	0.0	15	24431	57.1

model increases, and its performance slows down with the increasing size of the assignment graph ($|E_R|$) and the increasing size of the core graph ($|E_S|$). The latter one has a stronger influence on the performance of the model LG_xCUT_F. As the value of H increases, the compact model HOP_F outperforms LG_xCUT_F on sparser instances. For dense graphs ({c,d}-{15,20}) the memory requirements of the compact model prevent even from solving its LP relaxation. The number of instances not solved by HOP_F was 0, 4, 8 and 12 for $H = 3, 5, 7$ and 10, respectively.

Comparing the other two models, HOP_F and LG_xCUT_R, we observe that in many cases the compact model HOP_F outperforms LG_xCUT_R with respect to the running time. While for $H = 3$ LG_xCUT_R solves 12 out of 32 instances faster than HOP_F, for $H = 10$ the compact model is faster on all instances for which the memory limit was not exceeded.

We compared the average running times of all three models for the instances that HOP_F (and,

Table 3: Comparison of models LG_xCUT_F, LG_xCUT_R and HOP_F with $H = 5$. The best running times are shown in bold.

	LG_xCUT_F				LG_xCUT_R					HOP				
Inst.	OPT	BB	$\#Iter$	t [s]	g	g_{opt}	BB	$\#Iter$	t [s]	g	g_{opt}	BB	$\#Iter$	t [s]
c5mp1	2839.80	33	8700	**3.8**	0.0	0.0	25	13010	17.6	0.0	0.0	27	10705	18.1
c5mp2	2839.05	15	7053	**2.8**	0.0	0.0	15	9016	16.5	0.0	0.0	15	6962	17.2
c5mq1	3986.08	0	6804	**3.2**	0.0	0.0	0	6804	4.3	0.0	0.0	0	7031	33.5
c5mq2	3928.49	23	15822	**7.9**	0.0	0.0	17	19274	82.3	0.0	0.0	15	16033	41.9
c10mp1	2683.48	11	27665	**20.8**	0.0	0.0	17	61003	98.5	0.0	0.0	13	27117	60.1
c10mp2	2663.46	7	20670	**10.3**	0.0	0.0	5	26022	30.5	0.0	0.0	7	19826	42.8
c10mq1	3867.57	27	67196	**53.2**	0.0	0.0	56	172822	628.9	0.0	0.0	29	67977	191.1
c10mq2	3733.85	57	87075	**94.5**	0.0	0.0	69	180209	682.4	0.0	0.0	33	81605	342.7
c15mp1	2637.66	17	48095	**39.0**	0.0	0.0	19	108243	352.4	0.0	0.0	31	33536	58.4
c15mp2	2644.46	10	27452	**18.9**	0.0	0.0	13	65202	193.0	0.0	0.0	15	23222	39.7
c15mq1	3846.50	39	115710	**111.2**	0.0	0.0	37	240929	1250.0	0.0	0.0	53	98950	184.3
c15mq2	3692.56	25	97008	**115.6**	0.0	0.0	45	268070	1271.0	0.0	0.0	30	81486	228.3
c20mp1	2618.66	11	30786	**93.6**	0.0	0.0	25	44749	355.4	0.0	0.0	19	40529	185.0
c20mp2	2626.46	6	22961	**58.0**	0.0	0.0	7	23322	174.1	0.0	0.0	9	26279	118.0
c20mq1	3826.50	44	114730	**210.1**	0.0	0.0	47	108013	1175.0	-	-	-	-	-
c20mq2	3686.49	31	105996	**335.1**	0.0	0.0	57	124667	1286.0	-	-	-	-	-
d5mp1	2766.52	9	6448	**3.3**	0.0	0.0	5	7873	10.8	0.0	0.0	9	6589	17.6
d5mp2	2795.15	11	6053	**2.8**	0.0	0.0	5	6116	8.2	0.0	0.0	9	5790	17.1
d5mq1	4124.65	13	19360	**12.2**	0.0	0.0	17	22402	78.4	0.0	0.0	15	19051	46.2
d5mq2	3826.77	9	12584	**7.9**	0.0	0.0	7	16854	35.3	0.0	0.0	11	13359	40.3
d10mp1	2759.67	13	18377	**9.3**	0.0	0.0	5	35134	46.1	0.0	0.0	11	17230	31.3
d10mp2	2782.68	37	24085	**11.3**	0.0	0.0	37	45084	56.6	0.0	0.0	15	21125	33.5
d10mq1	3892.51	9	30322	**21.6**	0.0	0.0	19	41035	109.8	0.0	0.0	7	28429	85.6
d10mq2	3760.49	17	37470	**34.2**	0.0	0.0	23	91095	275.0	0.0	0.0	23	38865	118.3
d15mp1	2643.66	21	46410	**52.4**	0.0	0.0	17	82080	243.5	0.0	0.0	21	30308	55.6
d15mp2	2647.46	9	28322	**22.2**	0.0	0.0	9	52534	130.0	0.0	0.0	7	20863	40.7
d15mq1	3850.06	53	94289	**102.2**	0.0	0.0	53	152246	621.2	0.0	0.0	51	81428	148.8
d15mq2	3702.56	23	83001	**129.0**	0.0	0.0	19	197594	924.4	0.0	0.0	23	78052	235.3
d20mp1	2619.66	15	36363	**149.8**	0.0	0.0	11	47237	617.5	1.6	1.6	0	17616	172.0
d20mp2	2628.46	7	22770	**91.3**	0.0	0.0	9	27663	397.4	1.0	1.0	0	21951	184.7
d20mq1	3828.50	46	115641	**513.3**	0.0	0.0	54	136100	2647.0	-	-	-	-	-
d20mq2	3685.49	35	98976	**429.6**	0.0	0.0	53	119892	1958.0	-	-	-	-	-
Avg.		21	46381	86.6	0.0	0.0	25	79759	493.0	0.1	0.1	18	33640	99.6

of course, also the other two models) could solve to optimality. The factor, by which the compact model is faster than LG_xCUT_R strongly increases with the allowed number of hops. For $H = 3$ it is 2.0, for $H = 5$ it is 3.2, for $H = 7$ it is 9.0 and for $H = 10$ it is 12.5. The comparison of corresponding average running times for HOP_F and LG_xCUT_F shows a slightly different picture. For $H = 3$ and $H = 5$, LG_xCUT_F is approximately twice as fast as the compact model (factors of 0.5 and 0.4 respectively). For $H = 7$ model HOP_F is 1.5 times faster and for $H = 10$ the corresponding ratio is 3.5.

Figure 5 shows the increase of costs caused by a reduced number of allowed hops in the solution. Provided values are obtained as averages over 28 instances we could solve for all values of $H \in \{3, 5, 7, 10\}$.

Table 4: Comparison of models LG_xCUT_F, LG_xCUT_R and HOP_F with $H = 7$. The best running times are shown in bold.

	LG_xCUT_F				LG_xCUT_R					HOP				
Inst.	OPT	BB	$\#Iter$	t [s]	g	g_{opt}	BB	$\#Iter$	t [s]	g	g_{opt}	BB	$\#Iter$	t [s]
c5mp1	2703.97	13	18447	**10.9**	0.0	0.0	11	27276	26.8	0.0	0.0	13	17923	34.1
c5mp2	2736.55	25	23094	**9.9**	0.0	0.0	27	47641	59.1	0.0	0.0	15	21726	37.1
c5mq1	3906.98	29	50650	**37.2**	0.0	0.0	24	67556	193.3	0.0	0.0	23	43942	96.2
c5mq2	3842.99	49	80539	**77.0**	0.0	0.0	75	307967	778.0	0.0	0.0	45	78156	209.5
c10mp1	2661.66	25	55921	**57.5**	0.0	0.0	21	175953	461.8	0.0	0.0	12	30416	72.8
c10mp2	2663.46	9	43840	44.9	0.0	0.0	5	98566	249.1	0.0	0.0	25	22988	**41.9**
c10mq1	3867.57	51	118689	**136.3**	0.0	0.0	39	334988	1295.0	0.0	0.0	31	81028	180.1
c10mq2	3733.85	47	171302	**262.1**	0.0	0.0	57	416607	2078.0	0.0	0.0	41	97491	338.8
c15mp1	2634.66	17	85993	169.2	0.0	0.0	15	214975	981.5	0.0	0.0	68	50970	**98.4**
c15mp2	2640.46	12	62810	91.2	0.0	0.0	11	98959	475.7	0.0	0.0	51	28929	**57.9**
c15mq1	3844.50	53	193202	371.9	0.0	0.0	42	486451	2737.0	0.0	0.0	52	106759	**227.0**
c15mq2	3689.56	57	266504	501.0	0.0	0.0	44	373958	2414.0	0.0	0.0	29	100087	**269.7**
c20mp1	2618.66	35	59624	**299.9**	0.0	0.0	39	50313	882.6	-	-	-	-	-
c20mp2	2626.46	12	27692	**118.5**	0.0	0.0	15	27392	567.5	-	-	-	-	-
c20mq1	3826.50	81	169237	**685.7**	0.0	0.0	38	121211	2022.0	-	-	-	-	-
c20mq2	3686.49	269	165396	**841.5**	0.0	0.0	86	111921	2485.0	-	-	-	-	-
d5mp1	2685.94	10	15504	**6.2**	0.0	0.0	3	21203	21.4	0.0	0.0	11	14203	28.1
d5mp2	2761.15	22	31293	**13.3**	0.0	0.0	11	36929	48.0	0.0	0.0	21	23968	31.3
d5mq1	3903.51	21	44667	**42.9**	0.0	0.0	17	70306	204.4	0.0	0.0	15	44749	138.7
d5mq2	3744.49	17	49148	**48.7**	0.0	0.0	11	67838	154.7	0.0	0.0	13	43107	163.8
d10mp1	2685.54	17	52164	**66.0**	0.0	0.0	17	176343	457.9	0.0	0.0	21	34661	80.4
d10mp2	2693.46	13	43555	**44.7**	0.0	0.0	13	106356	285.8	0.0	0.0	13	27404	60.4
d10mq1	3873.06	33	143529	290.8	0.0	0.0	69	694032	3248.0	0.0	0.0	33	75724	**213.1**
d10mq2	3724.49	51	200516	482.7	0.0	0.0	27	603672	2978.0	0.0	0.0	23	71748	**270.7**
d15mp1	2639.66	41	95420	240.3	0.0	0.0	13	168002	959.0	0.0	0.0	32	39481	**87.6**
d15mp2	2647.46	9	51261	132.9	0.0	0.0	17	106702	595.5	0.0	0.0	5	26433	**66.3**
d15mq1	3847.06	43	172332	525.0	0.0	0.0	46	375512	2697.0	24.4	3.1	0	39697	**93.2**
d15mq2	3698.49	45	205779	775.3	0.0	0.0	35	428176	3573.0	24.8	2.6	0	42271	**107.4**
d20mp1	2619.66	26	43846	**407.7**	0.0	0.0	44	58982	1611.0	-	-	-	-	-
d20mp2	2628.46	18	32745	**292.2**	0.0	0.0	28	33444	1182.0	-	-	-	-	-
d20mq1	3828.50	118	158943	**1173.0**	0.0	0.0	61	122887	3477.0	-	-	-	-	-
d20mq2	3685.49	59	136651	**894.4**	0.0	0.0	41	116773	2290.0	-	-	-	-	-
Avg.		41	95947	286.0	0.0	0.0	31	192153	1296.5	2.0	0.2	25	48494	125.2

5.4. Solving HC ConFL as Steiner Arborescence

The second aim of our computational study was to analyze, whether the transformation into the Steiner arborescence problem, described in Section 3.2, can speed up the performance of the model LG_xCUT_R. Table 6 summarizes the obtained results.

Each entry in that table is an average value calculated over all $H \in \{3, 5, 7, 10\}$ and over all instances that could be solved to optimality by LG_xCUT_R. The average running time in seconds of LG_xCUT_R is given in the third column. Column 4 shows the average speed-up factor obtained by solving HC ConFL as the Steiner arborescence problem (formulation $LG_{STP}CUT$). The last column shows the average speed-up factor of LG_xCUT_F, compared to LG_xCUT_R. One observes that the speed-up factor increases with the size of the assignment graph.

Table 5: Comparison of models LG_xCUT_F, LG_xCUT_R and HOP_F with $H = 10$. The best running times are shown in bold.

	LG_xCUT_F				LG_xCUT_R					HOP				
Inst.	OPT	BB	$\#Iter$	t [s]	g	g_{opt}	BB	$\#Iter$	t [s]	g	g_{opt}	BB	$\#Iter$	t [s]
c5mp1	2692.66	39	78276	**78.1**	0.0	0.0	17	294564	789.5	0.0	0.0	33	43198	105.0
c5mp2	2692.46	27	65906	72.7	0.0	0.0	15	166161	408.3	0.0	0.0	17	27260	**65.7**
c5mq1	3906.98	62	182486	**204.8**	0.0	0.0	67	657451	2029.0	0.0	0.0	54	103537	230.5
c5mq2	3769.56	95	203270	**321.3**	0.0	0.0	52	768108	2946.0	0.0	0.0	59	112824	437.9
c10mp1	2661.66	41	251694	801.1	0.9	0.9	5	771577	3602.0	0.0	0.0	31	47887	**101.5**
c10mp2	2661.46	15	132510	307.4	0.0	0.0	5	342710	1435.0	0.0	0.0	84	33250	**65.3**
c10mq1	3867.57	35	294028	866.0	2.9	2.4	5	569299	3603.0	0.0	0.0	47	102548	**208.6**
c10mq2	3732.56	51	419344	970.5	1.6	1.6	12	671745	3603.0	0.0	0.0	62	122362	**341.1**
c15mp1	2634.66	35	198342	932.1	0.0	0.0	36	323422	2204.0	0.0	0.0	215	90268	**176.2**
c15mp2	2640.46	7	61654	197.9	0.0	0.0	14	97618	708.2	0.0	0.0	160	58242	**110.9**
c15mq1	3842.50	37	326566	**1041.0**	3.3	1.4	23	476541	3604.0	-	-	-	-	-
c15mq2	3689.56	43	164733	**676.0**	2.8	2.3	25	379847	3603.0	-	-	-	-	-
c20mp1	2618.66	46	54360	**567.5**	0.0	0.0	89	94524	3026.0	-	-	-	-	-
c20mp2	2626.46	23	27248	**227.3**	0.0	0.0	21	28179	909.4	-	-	-	-	-
c20mq1	3826.50	91	182093	**1180.0**	1.5	0.6	40	139302	3616.0	-	-	-	-	-
c20mq2	3686.49	128	159428	**1180.0**	0.0	0.0	52	137822	3013.0	-	-	-	-	-
d5mp1	2677.94	23	72843	103.1	0.0	0.0	9	297567	880.9	0.0	0.0	11	28519	**85.0**
d5mp2	2713.63	23	83292	**76.7**	0.0	0.0	11	125418	282.2	0.0	0.0	15	30305	82.2
d5mq1	3878.98	37	199597	456.9	0.0	0.0	25	461187	2050.0	0.0	0.0	33	78927	**241.7**
d5mq2	3741.49	88	407613	1004.0	0.5	0.5	20	710614	3603.0	0.0	0.0	27	79118	**321.9**
d10mp1	2678.94	31	212180	491.1	1.2	1.2	5	703605	3602.0	0.0	0.0	55	61059	**120.9**
d10mp2	2682.46	21	137930	376.3	0.0	0.0	7	474851	2521.0	0.0	0.0	15	34725	**76.9**
d10mq1	3869.06	69	502630	1968.0	3.0	2.0	3	552368	3603.0	0.0	0.0	77	114371	**250.1**
d10mq2	3724.49	47	576931	2839.0	2.5	2.5	2	496475	3603.0	0.0	0.0	77	114541	**344.2**
d15mp1	2635.66	23	105521	502.4	0.0	0.0	33	300581	2657.0	0.0	0.0	19	45540	**155.4**
d15mp2	2647.46	11	82989	392.8	0.0	0.0	15	168835	1739.0	0.0	0.0	11	31820	**145.1**
d15mq1	3844.50	39	298551	**1711.0**	1.7	1.4	17	322490	3605.0	-	-	-	-	-
d15mq2	3698.49	39	216203	**1262.0**	1.8	1.0	28	303107	3603.0	-	-	-	-	-
d20mp1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
d20mp2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
d20mq1	-	-	-	-	-	-	-	-	-	-	-	-	-	-
d20mq2	-	-	-	-	-	-	-	-	-	-	-	-	-	-
Avg.		44	203508	743.1	0.8	0.6	29	386999	2530.3	0.0	0.0	55	68015	183.3

Although we can observe that there is a speed-up obtained by solving the problem as the Steiner arborescence on LG_{STP}, the formulation LG_xCUT_F remains the best performing one. This can be explained by the density of connectivity cuts: cut-sets (19) are dense cuts involving both, core and assignment arcs, in general. In contrast, connectivity cuts (12) involve only core arcs, so they can be much sparser, especially if the assignment graph is a complete bipartite graph. Finally, the overall number of cuts of type (12) is significantly smaller than the number of cut-sets (19). Our computational study shows that the trade-off between weaker lower bounds and the number of potential cuts has been resolved in favor of the slightly weaker model LG_xCUT_F.

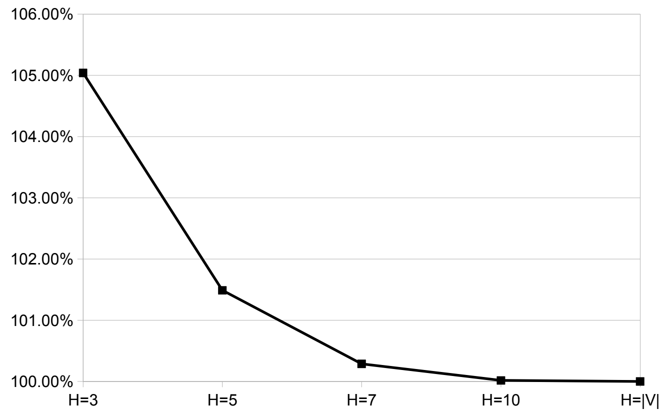

Figure 5: An average increase of cost induced by different hop limits.

Table 6: Speedup factors for solving HC ConFL as Steiner arborescence.

		t [s]	Relative speed-up	
STP	UFLP	LG_xCUT_R	$LG_{STP}CUT$	LG_xCUT_F
c	p1	511.5	1.2	**4.8**
c	p2	290.5	1.1	**4.8**
c	q1	772.8	1.3	**7.0**
c	q2	990.9	1.2	**6.2**
d	p1	458.7	1.1	**4.2**
d	p2	408.6	1.0	**4.3**
d	q1	1024.5	1.1	**4.9**
d	q2	651.8	1.1	**5.8**

5.5. 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. Figures 6 and 7 show 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: $Q_v = |V_x|/|V|$ (Figure 6) and $Q_a = |A_{Sx}|/|A_S|$ (Figure 7), for $H = 3, \ldots, 10$.

One observes that for sparse graphs (c5, c10) and smaller values of H, the graph LG_x is significantly smaller than G, which explains the efficacy of models on LG_x in those cases. Solving HC ConFL for $H = 3, 5$ 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 (2010)). As the density of the graph and / 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

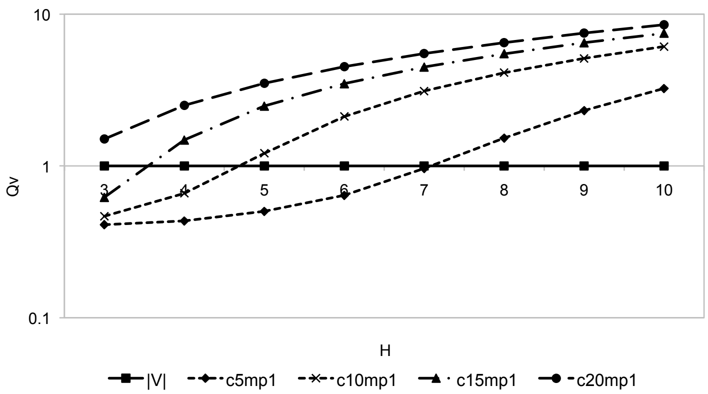

Figure 6: Relative size of V_x, $Q_v = |V_x|/|V|$.

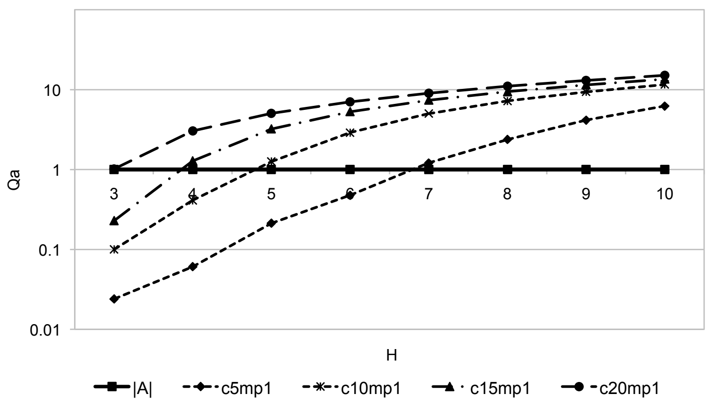

Figure 7: Relative size of A_{Sx}, $Q_a = |A_{Sx}|/|A_S|$.

graph models are better suited for sparse core graphs and / or smaller values of H. We recall that the density of the assignment graph does not influence the size of the layered graph LG_x.

6. Conclusions

Strongest MIP models for the hop constrained minimum spanning tree problem are obtained on layered graphs (see Gouveia, Simonetti, and Uchoa (2010)). Following this concept, we described two possibilities to develop strongest MIP models for hop constrained ConFL by modeling it as the directed ConFL problem on layered graphs. In the first transformation, a disaggregation of both, the core and the assignment graph 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 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 the memory usage. The size of the layered graph may drastically increase with the density of the core graph and with the number of allowed hops.

Acknowledgements

Ivana Ljubić is supported by the Hertha-Firnberg Fellowship of the Austrian Science Fund (FWF). The authors thank to Luis Gouveia for very useful discussions on topics related to HCMST and HC-STP. Preliminary results of this paper appeared in the Proceedings of the International Symposium on Combinatorial Optimization (ISCO), 2010.

Bibliography

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

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

Bardossy, M. G., S. Raghavan. 2010. Dual-Based Local Search for the Connected Facility Location and Related Problems. *INFORMS Journal on Computing* , to appear.

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. *European Journal of Operational Research* **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. Rothvoß, G. Schäfer. 2010. Connected facility location via random facility sampling and core detouring. *Journal of Computer and System Sciences* doi:10.1016/j.jcss.2010.02.001. In press.

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

Gollowitzer, S., I. Ljubić. 2010. MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research* doi:10.1016/j.cor.2010.07.002. , to appear.

Gouveia, L. 1996. Multicommodity flow models for spanning trees with hop constraints. *European Journal of Operational Research* **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 Computing* **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. 2010. Modelling hop-constrained and diameter-constrained minimum spanning tree problems as Steiner tree problems over layered graphs. *Mathematical Programming* doi:10.1007/s10107-009-0297-2. , to appear.

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

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

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

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 Computer Science*, vol. 4771. Springer, 157–169.

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

Mahdian, M., Y. Ye, J. Zhang. 2006. Approximation algorithms for metric facility location problems. *SIAM Journal on Computing* **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.

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. *Annals of Operations Research* **86** 321–345.