Capacitated Network Design with Facility Location

Stefan Gollowitzer[a], Bernard Gendron[b], Ivana Ljubić[a]

[a] *Department of Statistics and Operations Research,*
Faculty of Business, Economics, and Statistics,
University of Vienna, Austria
[b] *Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), and*
Department of Computer Science and Operations Research, Université de Montréal
C.P. 6128, Succursale Centre-Ville, Montréal, Canada H3C 3J7

Abstract

We consider a network design problem that arises in the design of last mile telecommunication networks. It combines the capacitated network design problem (CNDP) with the single-source capacitated facility location problem (SSCFLP). We will refer to it as the Capacitated Connected Facility Location Problem (CapConFL). We develop a basic integer programming model based on single-commodity flows. Based on valid inequalities for the subproblems, CNDP and SSCFLP, we derive several (new) classes of valid inequalities for the CapConFL. We use them in a branch-and-cut framework and show their applicability on a set of real-world instances.

Keywords: Capacitated Network Design, Facility Location, Connected Facility Location, Mixed Integer Programming Models, Telecommunications

1. Introduction

Given a set of customers, a set of potential facility locations and some inter-connection nodes, the goal of the *Connected Facility Location* problem (ConFL) is to find the minimum-cost way of assigning each customer to exactly one open facility, and connecting the open facilities via a Steiner tree. The sum of costs for the Steiner tree, the facility opening costs and the assignment costs needs to be minimized. This problem has been used to model a network design problem that arises in the design of last mile telecommunication networks when the *fiber to the curb* (FTTC) deployment strategy is applied (see, e.g., [17]). Contrary to the *fiber to the home* strategy, where each customer, i.e., household, has its own fiber-optic uplink, in the FTTC strategy some of the existing copper wire infrastructure is used. More precisely, in an FTTC network, fiber optic cables run from a central office to a cabinet serving a neighborhood. End users connect to this cabinet using the existing copper connections. Expensive switching devices are installed in these cabinets. The usage of the last d meters of copper wire between the customer and a switching device may significantly reduce deployment costs while still enabling broadband connections of reasonable quality.

In more detailed planning of FTTC networks, capacities of the links and of multiplexor devices are limited and this aspect was not captured by the ConFL variants studied in the literature so

Email addresses: `stefan.gollowitzer@univie.ac.at` (Stefan Gollowitzer), `gendron@iro.umontreal.ca` (Bernard Gendron), `ivana.ljubic@univie.ac.at` (Ivana Ljubić)

far. In this paper we consider a new capacitated variant of the ConFL problem, that we will refer to as the *Capacitated Connected Facility Location Problem* (CapConFL).

In a typical application from telecommunications, demands of customers are given as the number of twisted copper lines that are to be “served” at the respective customer location. Switching (or multiplexor) devices have both capacity and demand. Capacity is defined in terms of the number of twisted copper lines a device can serve. The demand of a switching device is defined as the number of fiber-optic uplinks required to connect the device to the central office (which is further connected to the backbone network). The number of uplinks is fixed for each device and independent of the number of customers that are finally assigned to it. The CapConFL consists of deciding on the location of switching devices, the assignment of customers to these devices and the routing of the uplinks from the switching devices to the central office, while minimizing the overall investment costs.

1.1. Problem definition

More formally, CapConFL can be defined as follows. The input is a graph $G = (V, E_S \cup A_R)$ with the set of nodes V partitioned into the set of *customers* (R), the set of *potential facility locations* (F) and the set of *potential Steiner nodes* ($V \setminus (F \cup R)$). A root node $r \in V \setminus (F \cup R)$ represents the connection to a higher order (e.g., backbone) network. The network $G_S = (V_S, E_S)$, where $V_S := V \setminus R$ and $E_S := \{e = \{i, j\} \in E \mid i, j \in V_S\}$ is called the *core network*. The *assignment network* $G_R = (F \cup R, A_R)$ consists of directed arcs between potential facilities and customers, i.e., $A_R = \{(i, k) \mid i \in F, k \in R\}$. The following input parameters are associated to the network:

- Facility opening cost $f_i \geq 0$, capacity $v_i > 0$ and demand $d_i > 0$ for each $i \in F$.
- Arc cost $c_e \geq 0$ and capacity $u_e > 0$ for each $e \in E_S$.
- Assignment cost $c_{ij} \geq 0$ for each $(i, j) \in A_R$.
- Customer demand $b_k > 0$ for each $k \in R$.

The goal is to find a subnetwork of G consisting of the set of open facilities F', the set of core edges E'_S and the set of assignment arcs A'_R such that:

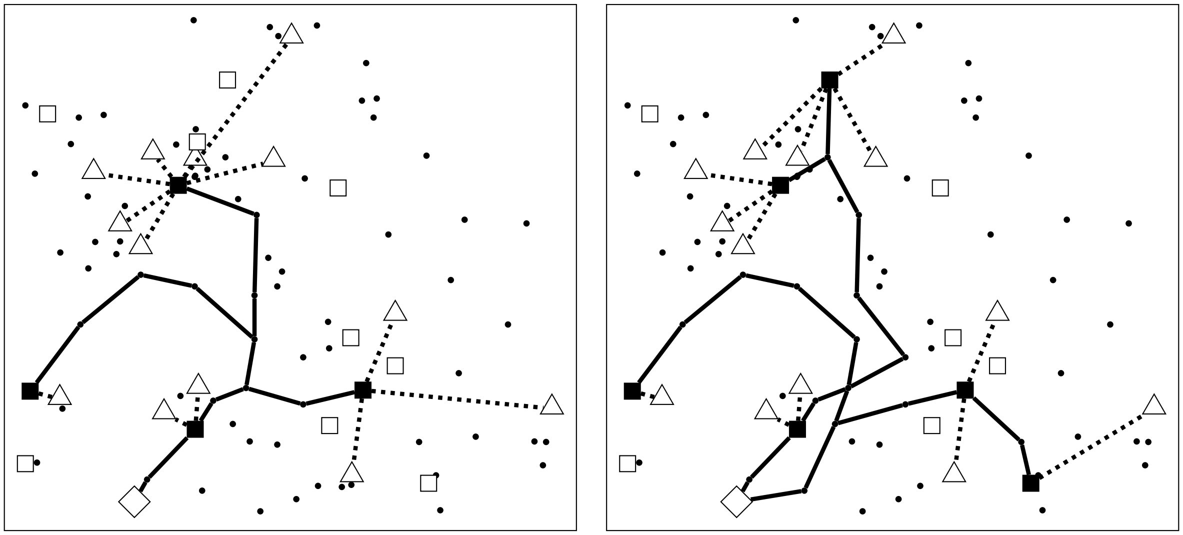
(P1) Each customer is assigned to exactly one open facility using arcs from A'_R.

(P2) The sum of customers’ demands assigned to a facility i does not exceed its capacity v_i.

(P3) In the core subnetwork induced by E'_S, we can simultaneously route the flow from the root node to satisfy the demand of all open facilities, without violating the edge capacities.

(P4) The sum of assignment, facility opening and edge costs, given by $\sum_{e \in E'_S} c_e + \sum_{i \in F'} f_i + \sum_{(i,j) \in A'_R} c_{ij}$, is minimized.

Obviously, by setting capacities $u_e = \infty$, for all $e \in E_S$ and $v_i = \infty$, for all $i \in F$, we obtain the previously studied ConFL problem. Figure 1 illustrates solutions for ConFL and CapConFL. Squares and triangles denote facilities and customers, respectively. A black fill indicates that a facility is open. A diamond denotes the root node. Solid edges are in the core network, dotted edges represent the assignments. In the CapConFL the limited facility capacities require two



(a) ConFL solution

(b) CapConFL solution

Figure 1: Feasible solutions of the ConFL and CapConFL problem, respectively.

additional open facilities and a different assignment of customers to facilities. The limited edge capacities require additional edges in the core network.

Notice that ConFL combines the Steiner tree problem and the uncapacitated facility location problem. On the other hand, CapConFL combines the capacitated network design problem with the single-source capacitated facility location problem. To see this, consider a feasible CapConFL instance whose core graph has a star topology. One can easily transform this input graph into an instance of the single-source capacitated facility location problem: the facility opening costs for each $i \in F$ are now defined as $c_e + f_i$ where c_e is the corresponding adjacent edge, and the assignment graph remains unchanged. Similarly, a feasible CapConFL instance in which the assignment arcs are such that each customer is adjacent to exactly one facility can be reduced into an instance of the single-source capacitated network design problem.

1.2. Literature review

Since CapConFL has not been considered before, we provide a detailed literature overview of three closely related problems: connected facility location, capacitated network design and single-source capacitated facility location.

Connected Facility Location. Early work on ConFL mainly includes approximation algorithms. ConFL can be approximated within a constant ratio and the currently best-known approximation ratio is provided by Eisenbrand et al. [13]. Recently, heuristic approaches have been proposed by Ljubić [25] and Bardossy and Raghavan [3]. Gollowitzer and Ljubić [17] present and compare several formulations for ConFL, both theoretically and computationally. Some of these results will be discussed and related to the CapConFL later on. Arulselvan et al. [2] consider a time-dependent variant of the ConFL and present a branch-and-cut approach based on cover, cut set cover and degree balance inequalities. Leitner and Raidl [24] propose a branch-and-cut-and-price approach for a variant of ConFL with *capacities on facilities.* Cutting planes are used to ensure paths between the root and open facilities, while column generation is used for selecting open facilities and assigning customers to them.

(Single-Source) Capacitated Network Design Problems (CNDP). In a typical CNDP setting, a network is given with a limited capacity available on each edge. A subset of edges of minimum cost needs to be installed in the network such that commodities with multiple origins and multiple destinations can be routed through the network without violating installed edge capacities. There exists a large body of work on the CNDP and related problems.

It includes exact methods based on Lagrangian relaxation or decomposition [14, 19, 9, 23], heuristic methods based on tabu search, neighbourhood search, slope scaling and Lagrangian relaxation [8, 15, 16, 10]. Recent developments comprise a theoretical study and comparison of Benders, metric and cut set inequalities [7] and a hybrid method combining mathematical programming and neighbourhood search techniques [18]. Finally, Chouman et al. [5] present a branch-and-cut approach that compares several families of valid inequalities for the CNDP.

A generalization of the single-source CNDP is the *Local Access Network Design* problem (LAN). In this problem multiple copies of each edge are available. The Local Access Network Design problem was studied by Raghavan and Stanojević [27], Salman et al. [28] and Ljubić et al. [26].

The Single-Source Capacitated Facility Location Problem (SSCFLP). Aardal et al. [1] and Deng and Simchi-Levi [11] proposed MIP models and studied the corresponding polyhedra of the SSCFLP and related problems. Holmberg et al. [20] present a branch-and-bound method based on a Lagrangean heuristic, Diaz and Fernández [12] develop a branch-and-price approach based on a decomposition of the SSCFLP and Contreras and Díaz [6] propose a scatter search heuristic. Ceselli et al. [4] give an exhaustive computational evaluation of branch-and-cut and branch-and-price approaches for a general class of facility location problems that includes the SSCFLP.

1.3. Contribution and outline

In Section 2 we introduce a basic integer programming model for CapConFL and discuss the relation of CapConFL and the Connected Facility Location problem. In particular, we show that a domination result between two sets of valid inequalities for ConFL does not hold for CapConFL. In Section 3 we derive cover and extended cover inequalities for the various knapsack type constraints in our model. In addition, we provide two generalizations of recently proposed cut-set-cover inequalities and cover inequalities for single cut sets. Separation procedures for these valid inequalities are discussed in Section 4. In Section 5 we illustrate the effectiveness of the proposed model and the valid inequalities by computational experiments on a set of new, realistic benchmark instances based on real data. We conclude the paper in Section 6.

2. Mixed integer programming models

In this section we introduce a first basic model for the CapConFL. It is based on models familiar in the context of the SSCFLP and the CNDP. We then strengthen this model using concepts known from the Connected Facility Location problem [17].

Since all demands of open facilities have to be routed from a single source node, it can be shown (see, e.g., [26]) that without loss of generality we can replace the undirected core network G_S by a bidirected graph in which each edge $e \in E_S$ is replaced by two directed arcs, except for the edges adjacent to the root node, where it is sufficient to consider outgoing arcs from r. The set of arcs of the bidirected core network will be denoted by A_S. Since the flow routed through an edge will always be routed in one of the two opposite directions, we define cost and capacities as $c_{ij} = c_e$ and $u_{ij} = u_e$, respectively, for each $e = \{i, j\}$ in E_S. The union of core and assignment arcs is denoted by $A = A_S \cup A_R$. For a set of customers $J \subset R$ we denote the set of facilities that can serve these customers by $F(J) = \bigcup_{k \in J} F(k)$, where $F(k) := \{i \in F : (i,k) \in A_R\}$. Likewise, for $I \subset F$ we denote by $R(I) = \bigcup_{i \in I} R(i)$ where $R(i) := \{k \in R : (i,k) \in A_R\}$. For $W \subset V$ we denote the set of ingoing arcs by $\delta^-(W)$.

2.1. The basic MIP model

In our models we will use the following binary decision variables:

$$x_{ij} = \begin{cases} 1, & \text{if arc } (i,j) \text{ is installed} \\ 0, & \text{else} \end{cases} \qquad (i,j) \in A$$

$$z_i = \begin{cases} 1, & \text{if facility } i \text{ is installed} \\ 0, & \text{else} \end{cases} \qquad i \in F$$

In addition, continuous flow variables g_{ij} indicate the total amount of the flow between the root r and all open facilities in F routed through the arc $(i,j) \in A$.

The following model combines the single-commodity flow (SCF) formulation for the CNDP (see, e.g., [27, 28]) with a formulation for the SSCFLP (see, e.g., [20]):

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

$$\text{s.t.} \quad \sum_{ji \in A_S} g_{ji} - \sum_{ij \in A_S} g_{ij} = \begin{cases} d_l z_l & i = l \\ -\sum_{l \in F} d_l z_l & i = r \\ 0 & \text{else} \end{cases} \quad i \in V_S \tag{1a}$$

$$0 \le g_{ij} \le u_{ij} x_{ij} \qquad (i,j) \in A_S \tag{1b}$$

$$\sum_{k \in R(i)} b_k x_{ik} \le v_i z_i \qquad i \in F \tag{1c}$$

$$x_{ik} \le z_i \qquad i \in F, k \in R(i) \tag{1d}$$

$$\sum_{i \in F(k)} x_{ik} = 1 \qquad k \in R \tag{1e}$$

$$x_{ij} \in \{0,1\} \qquad (i,j) \in A \tag{1f}$$

$$z_i \in \{0,1\} \qquad i \in F \tag{1g}$$

Constraints (1c)-(1e) are the strong relaxation of the SSCFLP. The assignment constraints (1e) model the property (P1) and constraints (1c)-(1d) ensure the property (P2). In constraints (1a)-(1b) we use the single-commodity flow variables to ensure the property (P3). This model is intuitive, but it provides weak lower bounds, due to the following facts: 1) big-M constraints (1b) are used to model the arc capacities, and 2) the connectivity between the root and the open facilities, rather than between the root and the customers, is required. The model is impractical to solve in a branch-and-bound framework, even for medium sized instances.

Using the following *capacitated cut set inequalities* we can project out the flow variables from the previous model and replace the constraints (1a)-(1b) by the following ones (see, e.g., Ljubić et al. [26]):

$$\sum_{ij \in \delta^-(W)} u_{ij} x_{ij} \geq \sum_{l \in F \cap W} d_l z_l \qquad W \subseteq V_S \setminus \{r\} \tag{Cut$_{SCF}$}$$

The obtained model contains an exponential number of inequalities and provides the same lower bounds as the corresponding flow model. However, inequalities (Cut$_{SCF}$) can be strengthened as follows:

$$\sum_{ij \in \delta^-(W)} \min(u_{ij}, \sum_{l \in F \cap W} d_l) x_{ij} \geq \sum_{l \in F \cap W} d_l z_l \qquad W \subseteq V_S \setminus \{r\}$$

2.2. Relations to Connected Facility Location and cut set inequalities

In [17] we studied MIP formulations for ConFL and provided a complete hierarchy of several MIP formulations with respect to the quality of their LP-bounds. Among others, we described two cut set-based formulations for ConFL. The models differ in the way they require connectivity.

In the first model, connectivity is ensured between the root and any open facility as follows:

$$\sum_{ij \in \delta^-(W)} x_{ij} \geq z_l \qquad W \subseteq V_S \setminus \{r\},\ l \in W \cap F \tag{Cut$_Z$}$$

These inequalities state that for each open facility the edges on at least one path between the root node and the respective facility need to be installed. Additional assignment constraints (1d) and (1e) are required between the facilities and customers.

The second model replaces constraints (Cut$_Z$) by the following cut set inequalities that ensure connectivity between the root and every customer:

$$\sum_{ij \in \delta^-(W)} x_{ij} \geq 1 \qquad W \subseteq V \setminus \{r\}, W \cap R \neq \emptyset \tag{Cut$_X$}$$

We showed that for ConFL the second model provides theoretically stronger lower bounds, but is computationally outperformed by the first model on the set of benchmark instances considered there.

Both sets of inequalities, (Cut$_Z$) and (Cut$_X$) are also valid for CapConFL. It is interesting to mention that, unlike for the ConFL, for which the inequalities (Cut$_Z$) are implied by the model with (Cut$_X$) constraints, the two families of inequalities can be used complementary to each other for CapConFL:

Lemma 1. *Inequalities* (Cut$_Z$) *and* (Cut$_X$) *both strengthen the LP-relaxation of the basic model (SCF). However, the MIP models (SCF)+*(Cut$_X$) *and (SCF)+*(Cut$_Z$) *are incomparable w.r.t. the quality of their LP-bounds.*

Proof. It is not difficult to see that inequalities (Cut$_Z$) and (Cut$_X$) both strengthen the LP-relaxation of (SCF). To see that (Cut$_Z$) inequalities are not implied by (SCF)+(Cut$_X$), consider the example shown in Figure 2. A vector $(\mathbf{x}, \mathbf{z})$ that satisfies (Cut$_{SCF}$) is $x_{12} = 0.75$, $x_{23} = x_{24} = 0.25$, $z_3 = z_4 = 0.75$, $x_{35} = x_{46} = 0.75$ and $x_{45} = x_{36} = 0.25$. This solution is cut off by the (Cut$_X$) constraints $x_{23}+x_{24} \geq 1$ and $x_{12} \geq 1$. Finally, inequalities (Cut$_Z$) are not redundant for (SCF)+(Cut$_X$)

since they ensure $x_{23} + x_{24} \geq 1.5$ which further strengthens the model.

Conversely, the model (SCF)+(Cut$_Z$) does not imply (Cut$_X$) constraints, which follows from the previous results for ConFL in [17], i.e., a CapConFL instance with sufficiently large capacities on arcs and facilities will have the desired property. □

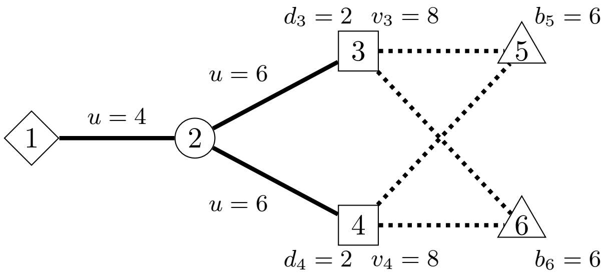


Figure 2: Example for comparison of cut set inequalities

3. Valid Inequalities

For the well-known subproblems of CapConFL, SSCFLP and CNDP, several sets of strengthening valid inequalities are known. We will review ideas that seem relevant in the context of the CapConFL and propose several sets of new valid inequalities based on the combination of the facility location and network design aspect.

3.1. Cover inequalities for single facilities

Deng and Simchi-Levi [11] proposed cover inequalities for the SSCFLP with uniform capacities. These inequalities are better known in the context of general mixed integer programming to strengthen knapsack-type constraints. We will use the concept of *extended cover inequalities* (see, e.g., the recent work of Kaparis and Letchford [22]).

Consider an arbitrary potential facility node $i \in F$. We call a set $R' \subseteq R(i)$ a cover for $i \in F$ if $\sum_{k \in R'} b_k > v_i$ and minimal if $\sum_{k \in R'} b_k - b_\ell \leq v_i$ for all $\ell \in R'$. We call it a minimal cover if it is minimal and a cover. For a minimal cover R', we define $E(R') = \{k \in R(i) \setminus R' : b_k \geq b^*\}$, where $b^* = \max_{k \in R'} b_k$.

Let the set of all minimal covers of $i \in F$ be denoted by $MC(i)$. Then the following *extended knapsack cover* inequalities are valid for the CapConFL:

$$\sum_{j \in R' \cup E(R')} x_{ij} \leq (|R'| - 1) z_i \qquad R' \in MC(i), i \in F \tag{EKS}$$

3.2. Inequalities involving multiple facilities

We derive two new families of inequalities that are implied by the limited capacities of facilities and the limited number of assignments edges in A_R.

Minimum cardinality inequalities on facilities. For a given set of customers $J \subset R$ and the corresponding subset of facilities $F(J)$, let $p(J)$ be the minimum number of facilities in $F(J)$ that is required to assign the customers in J in a feasible way, i.e., by respecting the allowed possible assignments and satisfying the capacity constraints on the facilities in $F(J)$. In other words, $p(J)$ is the optimum solution of a capacitated bin-packing problem with the set of bins $F(J)$, capacities v_i for $i \in F(J)$, the set of items J, demands b_j for $j \in J$ and such that each item $j \in J$ is only allowed to be assigned to bins in $F(j)$. W.l.o.g. we can assume that $b_k \leq v_i$ for all $(i,k) \in A_R$ and thus $p(\{k\}) = 1$ for all $k \in R$ and $p(J) \leq \min\{|F(J)|, |J|\}$ for all $J \subseteq R$.

Then the following *minimum cardinality* inequalities are valid for the CapConFL:

$$\sum_{i \in F(J)} z_i \geq p(J) \qquad J \subseteq R \tag{MC$_F$}$$

(Extended) Cover inequalities on facilities. Next we apply the idea of cover inequalities to the relation of facility capacities and customer demands. Let again $J \subseteq R$. We call a set $F' \subset F(J)$ a *capacity cover* with respect to J if $\sum_{i \in F(J) \setminus F'} v_i < b(J)$ and we call it minimal if $v_k + \sum_{i \in F(J) \setminus F'} v_i \geq b(J)$ for all $k \in F'$. Let $CC(F(J))$ denote the set of all such capacity covers of $F(J)$. We call the following set of constraints *cover inequalities on facilities*:

$$\sum_{i \in F'} z_i \geq 1 \qquad F' \in CC(F(J)), J \subseteq R \tag{2}$$

Similar to the cover inequalities for single facilities we can extend the covers and obtain stronger inequalities. Let $v^* = \max_{i \in F'} v_i$ and let $E(F') = \{i \in F(J) \setminus F' : v_i \geq v^*\}$ be the set of remaining facilities from $F(J)$ with a capacity of at least v^*. We refer to the following inequalities as *extended cover inequalities on facilities*:

$$\sum_{i \in F' \cup E(F')} z_i \geq 1 + |E(F')| \qquad F' \in CC(F(J)), J \subseteq R \tag{Cov$_F$}$$

To see that these inequalities are valid we can rewrite inequalities (2) as $\sum_{i \in F'} (1 - z_i) \leq |F'| - 1$. The corresponding extended cover inequality is then

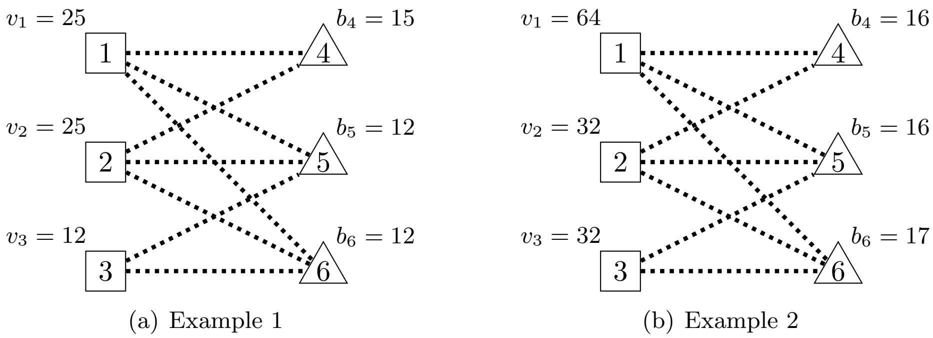
$$\sum_{i \in F' \cup E(F')} (1 - z_i) \leq |F'| - 1.$$

Rewriting this inequality gives (Cov$_F$).

The sets of inequalities (MC$_F$) and (Cov$_F$) do not contain each other as the following counterexamples show. In the example in Figure 3(a) a valid (Cov$_F$) inequality is $z_1 + z_2 \geq 2$, while the (MC$_F$) inequalities only ensure $z_1 + z_2 + z_3 \geq 2$. On the contrary, for the example given in Figure 3(b) the (Cov$_F$) inequalities are $z_1 + z_2 \geq 1$ and $z_1 + z_3 \geq 1$, but they are strictly dominated by the (MC$_F$) inequality $z_1 + z_2 + z_3 \geq 2$ that also implies $z_2 + z_3 \geq 1$.

General representation of cover inequalities on facilities. Consider now a general valid inequality of type

$$\sum_{i \in \hat{F}} z_i \geq p \tag{3}$$



(a) Example 1

(b) Example 2

Figure 3: Counterexamples for comparison of (MC$_F$) and (Cov$_F$)

defined for a set $\hat{F} \subseteq F$ and $p \geq 1$. For $p = 1$ we have the simple cover inequalities (2) (i.e., $\hat{F} \in CC(F(J))$) and, for $p \geq 2$, inequalities of type (MC$_F$) and (Cov$_F$) belong to this family, i.e., we have $\hat{F} \in F(J) \cup \{F' \cup E(F') \mid F' \in CC(F(J))\}$, for $J \subseteq R$. The following family of *general cover inequalities on facilities* is then also valid for our problem:

$$\sum_{i \in \tilde{F}} z_i \geq 1 \quad \tilde{F} \subseteq F, |\tilde{F} \cap \hat{F}| \geq |\hat{F}| - p + 1 \tag{Cov$_{gen}$}$$

It is not difficult to see that the latter inequalities are implied by (3). However, they are of particular interest when combined with cut set inequalities, as explained below.

3.3. Cut-set-cover inequalities

This new family of valid inequalities combines cut set inequalities with the general cover inequalities for facilities of the form (Cov$_{gen}$). Inequalities (Cov$_{gen}$) state that at least one facility in $\tilde{F}$ needs to be opened in a feasible solution. Consequently, for every subset of nodes $W \subset V$ containing all nodes in $\tilde{F}$, at least one ingoing arc needs to be installed. Let $\mathcal{F}$ denote the family of all subsets of facilities for which (Cov$_{gen}$) is valid, i.e.:

$$\mathcal{F} = \bigcup_{J \subseteq R} F(J) \cup \{F' \cup E(F') \mid F' \in CC(F(J))\}$$

and let

$$p(\hat{F}) = \begin{cases} 1 + |E(F')|, & \hat{F} = F' \cup E(F'),\ F' \in CC(F(J)) \\ p(J), & \hat{F} = F(J) \end{cases}$$

for all $\hat{F} \in \mathcal{F}$. The following *cut-set-cover* inequalities are valid for CapConFL and not implied by any of the previously described sets of constraints:

$$\sum_{ij \in \delta^-(W)} x_{ij} \geq 1 \qquad \tilde{F} \subseteq W \cap F, |\tilde{F} \cap \hat{F}| \geq |\hat{F}| - p(\hat{F}) + 1,\ \hat{F} \in \mathcal{F} \tag{Cut$_{Cov}$}$$

Inequalities (Cut$_{Cov}$) are a generalization of the previously introduced cut-set-cover inequalities for the incremental ConFL studied in Arulselvan et al. [2]. Figure 4 illustrates inequalities (Cut$_{Cov}$) for two different subsets W and a cover inequality $z_1 + z_2 \geq 1$ of type (Cov$_F$). Figure 5 illustrates inequalities (Cut$_{Cov}$) for the minimum cardinality inequality $z_1 + z_2 + z_3 \geq 2$.

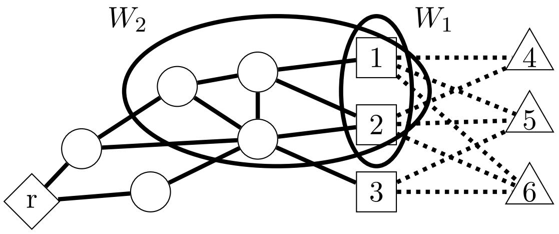


Figure 4: Example for cut-set-cover inequalities (Cut_{Cov}) derived from an inequality (Cov_F).

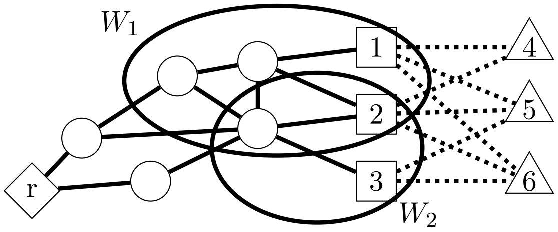


Figure 5: Example for cut-set-cover inequalities (Cut_{Cov}) derived from an inequality (MC_F).

3.4. Cover inequalities for single cut sets

The following set of valid inequalities generalizes the cover inequalities known for the capacitated network design problem studied in Chouman et al. [5]. Consider a (Cut_{SCF}) cut set inequality $\sum_{ij\in\delta^-(W)} u_{ij}x_{ij} \geq \sum_{l\in F\cap W} d_l z_l$ defined by a cut set $\delta^-(W)$ for $W \subseteq V \setminus \{r\}$. Let $F' \subseteq F \cap W$ and $d(F') = \sum_{l\in F'} d_l$. A set $C \subset \delta^-(W)$ is called a *cover* with respect to $\delta^-(W)$ and F', if $\sum_{ij\in\delta^-(W)\setminus C} u_{ij} < d(F')$ and a *minimal cover* if, in addition,

$$\sum_{ij\in\delta^-(W)\setminus C} u_{ij} + u_{lk} \geq d(F') \text{ for all } lk \in C.$$

Let $MC(W, F')$ denote the set of all minimal covers with respect to $\delta^-(W)$ and F'. Then the following *cover inequalities on single cut sets* are valid for the CapConFL:

$$\sum_{ij\in C} x_{ij} \geq 1 + \sum_{l\in F'} (z_l - 1) \qquad \forall C \in MC(W, F'). \tag{$\mathrm{Cov}_{\delta^-(W)}$}$$

Figure 6 illustrates inequalities ($\mathrm{Cov}_{\delta^-(W)}$). Edge (b, d) is a cover with respect to $W = \{1, 2, 3, d, e, f\}$ and $F' = \{2, 3\}$.

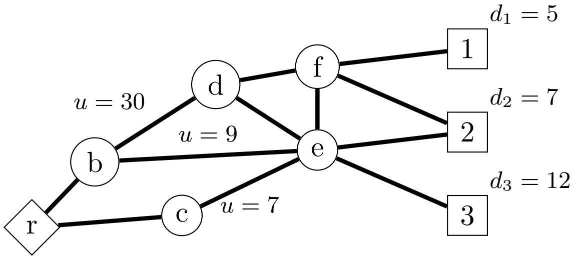


Figure 6: Illustration of cut set cover inequalities

4. Separation procedures

In this section we describe the separation procedures used in our branch-and-cut algorithm. We refer to the variable values of the current fractional solution by $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$.

4.1. Separation of inequalities (Cut_{SCF}) *and* (Cut_{Z})

Inequalities (Cut_{SCF}) can be separated in polynomial time (see also Ljubić et al. [26]). We define the support graph $G' = (V', A')$ where $V' := V_S \cup t$ with an additional sink node t, $A' := A_S \cup A_t$ and $A_t := \{(i,t) \mid i \in F, \bar{z}_i > 0\}$. We define capacities on arcs as $u_{ij}\bar{x}_{ij}$ for each arc $ij \in A_S$ and $d_i\bar{z}_i$ for each arc $it \in A_t$. We calculate the minimum cut between r and t in G'. Let $\delta^-(W)$ denote the arcs of this cut. If $\delta^-(W) \cap A_S \neq \emptyset$ and $\sum_{ij \in \delta^-(W) \cap A_S} u_{ij}\bar{x}_{ij} < \sum_{i \in W \cap F} d_i\bar{z}_i$ we have detected a violated inequality (Cut_{SCF}).

Inequalities (Cut_{Z}) can be separated in similar fashion (see also Gollowitzer and Ljubić [17]). The support graph in this case is the bidirected core network (V_S, A_S) with arc capacities set to $\bar{x}_{ij}$ for each arc $ij \in A_S$. A minimum cut in A_S between r and $l \in F$ with a weight of less than $\bar{z}_l$ corresponds to a violated inequality (Cut_{Z}).

4.2. Separation of inequalities (Cut_{X})

For the separation of (Cut_{X}) inequalities we define a support graph G_j for each $j \in R$. Thereby, $G_j = (V \cup \{j\}, A_S \cup A_j)$ where $A_j = \{(i,j) \mid i \in F(j)\}$. Capacities on the arcs from $A_S \cup A_j$ are set to $\bar{x}_{ij}$. Each minimum cut in G_j between r and $j \in R$ whose weight is less than 1 corresponds to a violated inequality (Cut_{X}).

If the number of customers is large, complete separation of inequalities (Cut_{X}) is very time-consuming. We therefore reduce the set of customers considered in the separation to a subset that still ensures all violated inequalities are identified. A customer $c_1 \in C$ is ignored if there exists another customer $c_2 \in C$ such that $F(c_2) \subset F(c_1)$. If $F(c_i)$ are identical for all $c_i \in \bar{C} \subseteq C$ only one customer in $\bar{C}$ is considered.

4.3. Separation of inequalities (EKS)

For a fractional point $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$ and for each $i \in \bar{F} = \{i \in F \mid \bar{z}_i > 0\}$, the separation of (simple, non-extended) inequalities (EKS) is equivalent to solving a knapsack problem which is described by the following integer program:

$$\begin{aligned} \min\ z = \sum_{j \in R} (\bar{z}_i - \bar{x}_{ij}) s_j & \\ \text{s.t.} \quad \sum_{j \in R} b_j s_j > v_i & \\ s_j \in \{0,1\} & \qquad \forall j \in R \end{aligned}$$

If $z < \bar{z}_i$ a simple, non-extended cover inequality is violated.

Instead of solving the separation problem exactly (e.g., using a dynamic programming procedure), we restrain to a heuristic separation which was proposed by Kaparis and Letchford [22]. Their efficient and fast separation heuristic detects extended knapsack cover inequalities. Adapted to our (EKS) inequalities, this procedure consists of the following steps, executed for each $i \in F$:

1. Sort the items in $R(i)$ in non-decreasing order of $(\bar{z}_i - \bar{x}_{ij})/b_j$, and store them in a list L. Initialize the cover R' as the empty set and initialize $b^* = v_i$.
2. Remove an item from the head of the sorted list L. If its weight is larger than b^*, ignore it, otherwise insert it into R'. If R' is now a cover, go to step 4.
3. If L is empty, stop. Otherwise, return to step 2.

4. If the extended cover inequality corresponding to R' is violated by $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$, output it.
5. Let $k^* = \arg\max_{j \in R'} b_j$ be the customer in R' with the highest demand. Set $b^* = b_{k^*}$ and delete k^* from R'. Return to step 2.

In fact, we perform two variants of this algorithm. The one stated above and one where the customers are sorted in non-increasing order of $\bar{x}_{ij}$.

4.4. Separation of inequalities (MC$_F$) *and* (Cov$_F$)

We consider subsets of facilities $F' \in F_C := F_1 \cup F_2$, where $F_1 := \{F(k) \mid k \in R\}$ and $F_2 := \{F(k_1) \cup F(k_2) \mid |F(k_1) \cap F(k_2)| / \min(|F(k_1)|, |F(k_2)|) \geq 0.5, k_1, k_2 \in R\}$, i.e., F_2 contains unions of $F(k_1)$ and $F(k_2)$ such that at least half the facilities of either $F(k_1)$ or $F(k_2)$ are common to both these sets. For each F' we define the subset of customers to be considered in the separation of (MC$_F$) and (Cov$_F$) inequalities as $J(F') := \{k' \in R \mid F(k') \subseteq F'\}$.

Separation of inequalities (MC$_F$). We calculate $p(J)$ for J and $F(J)$ by solving a bin-packing problem with assignment restrictions and non-uniform bin capacities:

$$\begin{aligned} p(J) = \min & \sum_{i \in F(J)} t_i & \\ \text{s.t.} & \sum_{k \in R(i)} b_k s_{ik} \leq v_i t_i & i \in F(J) \\ & s_{ik} \leq t_i & k \in J, i \in F(k) \\ & \sum_{i \in F(k)} s_{ik} = 1 & k \in J \\ & s_{ik} \in \{0,1\} & k \in J, i \in F(k) \\ & t_i \in \{0,1\} & i \in F(J) \end{aligned}$$

We consider $F' \in F_C$ as candidate sets for $F(J)$ and determine $J = J(F')$ as described in the previous paragraph. The values of $p(J)$ are calculated for all such J during preprocessing. In the separation procedure we repeatedly check whether the current fractional solution violates any of the stored inequalities (MC$_F$). By doing so we consider at most $|F_C| \leq |R| + |R|^2$ inequalities of type (MC$_F$).

Separation of inequalities (Cov$_F$). Given $J \subseteq R$ and $F(J)$, the separation of (simple, non-extended) covers on facilities (2) is equivalent to solving the following knapsack problem:

$$\begin{aligned} \min\ z = & \sum_{i \in F(J)} \bar{z}_i s_i & \\ \text{s.t.} & \sum_{i \in F(J)} v_i s_i > \sum_{i \in F(J)} v_i - b(J) & \\ & s_i \in \{0,1\} & i \in F(J) \end{aligned}$$

If $z < 1$, an inequality (2) is violated.

We consider $F(J)$ for $J = J(F')$ and $F' \in F_C$. To find covers in $CC(F(J))$ we use the separation procedure described in Section 4.3 with the following modifications: The facilities in $F(J)$ are ordered according to $\bar{z}_i/v_i$ in non-decreasing fashion and b^* is initialized with the maximum capacity of the facilities in $F(J)$.

4.5. Separation of inequalities (Cut_{Cov})

In the separation of (Cut_{Cov}) we consider all inequalities of the form (3) that were found by the separation procedure for (Cov_F) and (MC_F). For the corresponding set of facilities $\hat{F}$ and right-hand side p we randomly generate up to p sets $\bar{F} \subseteq \hat{F}$ such that $|\bar{F}| = |\hat{F}| - p + 1$. We separate inequalities (Cut_{Cov}) by running a maximum flow algorithm on graph G' defined as in Section 4.1, but with capacities of 1 on arcs it if $i \in \bar{F}$ and 0 if $i \notin \bar{F}$.

4.6. Separation of inequalities $(\text{Cov}_{\delta^-(W)})$

Given a cut set $W \subseteq V \setminus \{r\}$ and the set of facilities contained in that cut set, $F' = W \cap F$, a violated cut set cover inequality is detected by solving the following integer program:

$$\begin{aligned} \min\ z = & \sum_{ij \in \delta(W)} \bar{x}_{ij} s_{ij} - \sum_{l \in F'} (\bar{z}_l - 1) t_l \\ \text{s.t.} \quad & \sum_{ij \in \delta^-(W)} u_{ij} s_{ij} + \sum_{l \in F'} d_l t_l > \sum_{ij \in \delta^-(W)} u_{ij} \\ & s_{ij} \in \{0,1\} \quad ij \in \delta^-(W) \\ & t_l \in \{0,1\} \quad l \in F' \end{aligned}$$

A $(\text{Cov}_{\delta^-(W)})$ inequality is violated if $z < 1$.

We separate inequalities $(\text{Cov}_{\delta^-(W)})$ as follows: All cut sets W that are obtained during the separation of inequalities (Cut_{SCF}) are kept in a pool. We choose $F' = \{l \in F \cap W \mid \bar{z}_l > 0.1\}$. Then we use the following heuristic procedure to find minimal covers $C \in MC(W, F'')$, where $F'' \subseteq F'$:

1. Sort the items in $\delta^-(W)$ and F' in non-decreasing order of $(1 - \bar{z}_i)/d_i$ and $\bar{x}_{ij}/u_{ij}$ and store them in a list L. Initialize the cover C and F'' as empty sets and initialize $b^* = \sum_{ij \in \delta^-(W)} u_{ij}$.
2. Remove an item from the head of the sorted list L.
   (a) If it is an arc and its weight is larger than b^*, ignore it, otherwise insert it into R'.
   (b) If it is a facility insert it into F''.
   If C is now a cover with respect to $\delta^-(W)$ and F'', go to step 4.
3. If L is empty, stop. Otherwise, return to step 2.
4. If the cover inequality corresponding to $C \in MC(W, F'')$ is violated by $(\bar{\mathbf{x}}, \bar{\mathbf{z}})$, output it.
5. Let $ij^* = \arg\max_{ij \in C} u_{ij}$ be the arc in C with the highest capacity. Set $u^* = u_{ij^*}$ and delete ij^* from C. Return to step 2.

5. Computational results

In this section we report the results of our computational experiments. They were performed on a desktop machine with an 8-core Intel Core i7 CPU at 2.80 GHz and 8 GB RAM. Each run was performed on a single processor. We used the CPLEX [21] branch-and-cut framework, version 12.2. All cutting plane generation procedures provided by CPLEX are turned off unless stated explicitly. All heuristics provided by CPLEX are turned off. The other parameters are set to their default values.

5.1. Branch-and-cut framework

The settings described in this section are the result of our preliminary testing.

To reduce the number of constraints that need to be identified by our separation routines we add degree balance constraints and subtour elimination constraints for cycles of size two to our model:

$$
\begin{aligned}
x_{ji} &\le x(\delta^+(i)) + z_i && (j,i) \in A_S, i \in F && \text{(4a)}\\
x_{ji} &\le x(\delta^+(i)) && (j,i) \in A_S, i \in V_S \setminus (F \cup \{r\}) && \text{(4b)}\\
z_i &\le x(\delta^-(i)) && i \in F && \text{(4c)}\\
x_{ij} &\le x(\delta^-(i)) && (i,j) \in A_S, i \in V_S \setminus \{r\} && \text{(4d)}\\
x_{ij} + x_{ji} &\le 1 && (i,j) \in A_S,\ i < j,\ i,j \neq r && \text{(4e)}
\end{aligned}
$$

In order to reduce the size of the linear programs solved throughout the process we relax constraints (1d) and add them only if they are violated. Separation procedures are called in the following order: (EKS) - (Cov_F) - (MC_F) - (1d) - ($\mathrm{Cov}_{\delta^-(W)}$) - ($\mathrm{Cut}_{Cov}$) - ($\mathrm{Cut}_Z$) - ($\mathrm{Cut}_X$) - ($\mathrm{Cut}_{SCF}$). To prevent a tailing off effect of the separation procedures we stop separating valid inequalities if the lower bound has improved by less than 0.05% for the last 10 calls of the separation procedures. We apply this rule in each node of the branch-and-bound tree.

Inequalities (Cov_F), (MC_F), ($\mathrm{Cov}_{\delta^-(W)}$), ($\mathrm{Cut}_{Cov}$) and ($\mathrm{Cut}_{SCF}$) are only separated at the root node of the branch-and-bound tree. Inequalities (EKS) and (1d) are separated at every node, separation of (Cut_X) is done at every 10th node and separation of (Cut_Z) is done at every 100th node. In all nodes of the branch-and-bound tree we ensure the feasibility of our model by testing potential integer solutions for violation of inequalities (Cut_{SCF}).

To improve the computational efficiency of the separation procedures for cut set inequalities, we search for nested minimum cardinality cuts. To do so, all capacities in the respective separation graph are increased by some $\epsilon > 0$. Thus, every detected violated cut contains the least possible number of arcs. We resolve the linear program after adding at most 30 violated inequalities of any class. Finally, we randomly choose the target nodes to search for violated cuts.

5.2. Instances

We generated a set of realistic benchmark instances derived from real world data we were given. The real world data contain most of the information needed for complete CapConFL instances: The sets of facilities, Steiner nodes and edges of the core network; a set of customers with associated demands; a set of assignment arcs connecting customers and facilities, including their distance and an estimate of the bandwidth provided by the respective assignment arc; lengths of core edges and assignment arcs. These inputs define five graphs with different topologies that will be denoted by `A`, `B`, `C`, `D` and `E`. To complete the instances with respect to the input required by CapConFL we applied the following steps:

- For each instance a minimum customer bandwidth is selected, assignment arcs that provide less than this bandwidth are removed. We chose 20, 25 and 30 MBit/s and denote this by `20`, `25` and `30` in the instance label.

- At most 20 assignment arcs per customer are considered.

- Customers without assignment arcs are removed and facilities without assignment arcs are replaced by Steiner nodes.

- Steiner nodes with a degree of two and their adjacent edges are replaced by a single edge.
- A technology for each facility is randomly selected. For FTTB instances we consider the following combinations of capacity, demand and cost: $(32, 4, 4000)$, $(64, 5, 6000)$, $(128, 7, 8000)$. For FTTC instances we choose between $(64, 4, 13000)$, $(128, 4, 16000)$ and $(192, 4, 20000)$.
- Edge capacities are uniformly randomly selected from $[0.7\mu, 1.3\mu]$, where μ is equal to the demand of the smallest set of facilities needed to feasibly assign the customers, given the facility capacities chosen before [9].

The key figures for the instances we use are listed in Table 1.

5.3. Comparison against basic model and general purpose solver

In the first part of our computational study we assess the influence of the cutting plane generation procedures built into CPLEX compared to the influence of the valid inequalities proposed in this work. To this end we ran our model with the following different settings: *Basic* is the cut set based model corresponding to SCF, i.e., the model consisting of constraints (1c)-(1g), (Cut_{SCF}) and (4a)-(4e). *Basic+CPX* is the basic model with all CPLEX cuts turned on. *All VI* is the basic model with all valid inequalities from Section 3 added. *All VI+CPX* is the basic model with all valid inequalities and CPLEX cuts turned on.

In Table 1 we compare the LP gaps (g_{LP}) and time to solve the LP relaxation (t_{LP}) for these four models. We calculated the gaps as $(UB - LB)/UB$, where UB is the best known integer solution found in all our tests and LB is the solution value of the LP relaxation of the respective model. In the last two lines we show the mean and median of the values in the respective column. The best LP gap of the four models is shown in bold.

From the results in Table 1 we conclude that the model without valid inequalities provides a weak LP bound with an LP gap of 14.28% on average over the considered instance set. The cutting planes provided by CPLEX can reduce the LP gaps of the basic model by almost one half to an average of 7.23%. This average gap is still substantial compared to 1.31% obtained by the model that is strengthened by the valid inequalities proposed in this paper. Using CPLEX cuts in addition only improves the average gap to 1.13%.

5.4. Influence of different sets of valid inequalities

In the second part of our computational study we assess the influence of the different sets of valid inequalities proposed in this work. We compare five different settings that differ by the sets of valid inequalities considered. For each setting we add a subset of valid inequalities to the basic model described above. Settings (Cut_Z), (Cut_X) and (Cut_Z)+(Cut_X) are self-explaining. Setting *All VI* is defined as above and setting *Most VI* uses inequalities (Cut_Z), (Cut_X), (EKS), (Cov_F) and (MC_F).

For each of these settings, Table 2 shows the gap of the linear programming relaxation, g_{LP}, calculated as in Table 1, the time needed to solve linear programming relaxation, t_{LP}, and the number of cutting planes added, *Cuts*. In the last two lines we show the mean and median of the values in the respective column. The best LP gap of all models is shown in bold.

We would like to point out several interesting aspects. The LP gaps of setting (Cut_Z) are substantially larger than the ones of all other settings. Surprisingly the same does not hold for setting (Cut_X), which on average gives even stronger LP bounds than setting (Cut_Z)+(Cut_X). We trace the difference between the gaps of (Cut_X) and (Cut_Z)+(Cut_X) to the criteria we used to prevent

	Instance properties							Basic		Basic+CPX		All VI		All VI+CPX	
	$\|V\|$	$\|E\|$	$\|F\|$	$\|R\|$	$\|E_S\|$	$\|E_R\|$	UB	g_{LP}	t_{LP}	g_{LP}	t_{LP}	g_{LP}	t_{LP}	g_{LP}	t_{LP}
A20-FTTC	2405	12827	1504	805	1636	11191	3315257	12.98	10	7.67	45	**3.56**	33	**3.56**	36
A25-FTTC	2283	11349	1371	794	1525	9824	3717919	9.54	11	4.15	18	3.36	36	**3.11**	43
A30-FTTC	1949	7689	1053	694	1289	6400	3356273	8.62	5	4.74	12	3.45	29	**2.77**	38
B20-FTTC	2300	9695	1624	510	1824	7871	4601666	15.87	17	4.37	62	**0.10**	39	**0.10**	39
B25-FTTC	2265	9141	1580	507	1792	7349	5051559	14.42	13	5.37	29	0.15	36	**0.07**	41
B30-FTTC	1848	5610	1152	450	1432	4178	6030938	9.18	8	4.06	16	0.10	17	**0.02**	18
C20-FTTC	4742	21388	3237	1160	3720	17668	8897504	21.06	149	11.12	409	0.45	141	**0.36**	152
C25-FTTC	4498	17477	2961	1134	3498	13979	9734526	16.29	55	9.91	228	0.57	134	**0.53**	145
C30-FTTC	3269	9303	1748	840	2557	6746	9332930	11.14	47	4.73	75	0.82	35	**0.70**	58
D20-FTTC	4042	24923	2241	1463	2661	22262	7962207	13.18	50	8.85	156	1.93	63	**1.90**	79
D25-FTTC	3925	21403	2106	1449	2557	18846	9490801	8.86	38	4.31	76	1.32	62	**1.01**	74
D30-FTTC	3407	11539	1588	1213	2274	9265	9971466	4.03	12	2.66	30	0.72	61	**0.45**	73
E20-FTTC	3426	18045	2143	1038	2492	15553	4858219	23.95	101	15.81	302	1.06	61	**0.86**	93
E25-FTTC	3290	12574	1965	1023	2369	10205	7419660	13.21	58	5.13	114	0.54	51	**0.34**	74
E30-FTTC	2149	4429	750	497	1747	2682	6082850	6.93	7	2.54	9	**0.00**	21	**0.00**	21
A20-FTTB	2405	12827	1504	805	1636	11191	2506131	16.00	10	9.56	37	3.23	40	**2.42**	48
A25-FTTB	2267	11284	1370	778	1525	9759	2741939	12.64	8	7.27	33	3.26	38	**2.81**	47
A30-FTTB	1949	7689	1053	694	1289	6400	2406722	10.01	4	3.75	17	2.73	26	**2.14**	37
B20-FTTB	2300	9695	1624	510	1824	7871	3781915	19.99	13	7.83	66	0.26	40	**0.21**	49
B25-FTTB	2265	9141	1580	507	1792	7349	4146838	18.22	11	5.77	31	0.17	37	**0.07**	37
B30-FTTB	1848	5610	1152	450	1432	4178	4797642	12.12	10	3.75	18	0.06	19	**0.05**	20
C20-FTTB	4742	21388	3237	1160	3720	17668	7337282	26.20	176	14.11	397	0.62	158	**0.56**	171
C25-FTTB	4498	17477	2961	1134	3498	13979	7836463	20.75	66	10.93	229	0.78	120	**0.69**	159
C30-FTTB	3269	9303	1748	840	2557	6746	7332424	14.18	44	6.85	63	0.89	56	**0.79**	61
D20-FTTB	4042	24923	2241	1463	2661	22262	7056336	13.94	46	9.30	131	1.06	57	**1.02**	68
D25-FTTB	3925	21403	2106	1449	2557	18846	8315107	10.26	39	6.22	78	1.44	65	**1.20**	71
D30-FTTB	3407	11539	1588	1213	2274	9265	8210407	4.67	11	3.05	28	0.56	50	**0.41**	60
E20-FTTB	3426	18045	2143	1038	2492	15553	3956572	29.98	188	19.18	413	1.38	64	**1.23**	88
E25-FTTB	3290	12574	1965	1023	2369	10205	5885571	19.53	99	9.27	170	3.07	65	**2.89**	60
E30-FTTB	2149	4429	750	497	1747	2682	4651295	10.53	5	4.66	9	1.55	24	**1.48**	23
mean								14.28	44	7.23	110	1.31	56	**1.13**	66
median								13.19	15	6.00	62	0.86	45	**0.75**	59

Table 1: Comparison of LP relaxation lower bounds with and without CPLEX cuts

	(Cut$_Z$)			(Cut$_X$)			(Cut$_Z$)+(Cut$_X$)			Most VI			All VI		
	g_{LP}	t_{LP}	*Cuts*	g_{LP}	t_{LP}	*Cuts*	g_{LP}	t_{LP}	*Cuts*	g_{LP}	t_{LP}	*Cuts*	g_{LP}	t_{LP}	*Cuts*
A20-FTTC	8.56	32	5711	3.57	48	3879	3.58	27	4168	**3.56**	31	4776	**3.56**	33	4523
A25-FTTC	4.48	20	4623	3.50	94	2984	3.51	31	3101	3.38	32	3528	**3.36**	36	3446
A30-FTTC	7.12	17	3565	4.05	47	2987	4.05	24	3144	**3.45**	27	3855	**3.45**	29	3660
B20-FTTC	11.85	105	7323	0.15	39	4459	0.15	26	5100	**0.10**	39	5140	**0.10**	39	5140
B25-FTTC	10.40	26	5669	0.18	50	4442	0.18	33	4815	**0.15**	35	4908	**0.15**	36	4908
B30-FTTC	5.03	17	3447	0.12	24	2382	0.12	16	2661	**0.10**	17	2728	**0.10**	17	2728
C20-FTTC	12.98	351	14191	0.54	293	9042	0.54	144	9813	**0.45**	142	10114	**0.45**	141	9831
C25-FTTC	8.32	265	11612	0.62	215	6947	0.62	122	7298	**0.56**	122	8107	0.57	134	7710
C30-FTTC	4.91	62	4514	0.98	79	3290	0.98	32	3809	**0.82**	35	3873	**0.82**	35	3799
D20-FTTC	7.79	232	13921	2.00	97	7561	2.00	51	8193	**1.93**	62	8751	**1.93**	63	8124
D25-FTTC	3.68	62	8784	1.33	68	5901	1.34	39	6056	**1.32**	62	6454	**1.32**	62	6529
D30-FTTC	1.28	50	6150	0.82	109	4578	0.82	55	4737	**0.72**	58	5001	**0.72**	61	4974
E20-FTTC	13.69	142	12891	1.10	140	8633	1.10	58	9566	**1.06**	61	10021	**1.06**	61	10021
E25-FTTC	4.17	50	8515	0.57	62	6046	0.58	35	6303	**0.54**	50	6656	**0.54**	51	6656
E30-FTTC	2.29	12	2222	0.01	20	1682	0.01	14	1795	**0.00**	22	1960	**0.00**	21	1849
A20-FTTB	10.46	25	5426	3.46	67	3681	3.58	38	3640	**3.23**	40	4880	**3.23**	40	4394
A25-FTTB	5.09	20	3857	3.53	78	2740	3.57	29	2851	**3.26**	32	3789	**3.26**	38	3457
A30-FTTB	8.25	14	2755	3.59	26	2036	3.60	13	2133	**2.73**	22	3074	**2.73**	26	2832
B20-FTTB	15.82	74	7895	0.36	54	4497	0.36	34	4977	**0.26**	38	5365	**0.26**	40	4944
B25-FTTB	13.71	45	6044	0.26	50	4503	0.26	32	4948	**0.17**	37	5051	**0.17**	37	4913
B30-FTTB	7.20	16	3478	0.11	25	2387	0.11	18	2661	**0.06**	22	2601	**0.06**	19	2664
C20-FTTB	16.75	657	15243	0.71	298	8207	0.71	148	8980	**0.62**	156	9785	**0.62**	158	9475
C25-FTTB	11.26	306	12195	0.88	93	6765	0.88	67	7266	**0.78**	118	7793	**0.78**	120	7364
C30-FTTB	6.73	84	4785	1.10	93	3380	1.10	50	3693	0.90	57	3942	**0.89**	56	3756
D20-FTTB	8.39	210	12375	1.10	65	5535	1.10	30	6459	**1.06**	59	7591	**1.06**	57	6476
D25-FTTB	4.56	75	8617	1.46	154	4824	1.46	50	4950	**1.44**	59	6104	**1.44**	65	5565
D30-FTTB	1.34	28	6054	0.67	43	4030	0.67	24	4287	**0.56**	46	4543	**0.56**	50	4308
E20-FTTB	18.30	213	12824	1.39	139	7394	1.39	68	7869	**1.38**	74	8747	**1.38**	64	8379
E25-FTTB	8.41	82	8547	3.09	113	5820	3.12	61	6010	**3.07**	50	6699	**3.07**	65	6443
E30-FTTB	5.36	20	2288	1.68	10	1617	1.68	13	1811	**1.55**	24	1937	**1.55**	24	1937
mean	8.27	110	7517	1.43	90	4740	1.44	46	5103	**1.31**	54	5592	**1.31**	56	5360
median	8.02	56	6102	1.04	67	4478	1.04	33	4881	**0.86**	43	5026	**0.86**	45	4928

Table 2: Comparison of LP relaxation lower bounds with different sets of valid inequalities turned on

a tailing off effect during separation. A comparison of the running times shows that separating valid inequalities with a different structure improves the overall running time of the LP relaxation. Approaches (Cut_Z) and (Cut_X) need 110 and 90 seconds on average, respectively whereas approach (Cut_Z)+(Cut_X) only takes 46 seconds to compute approximately the same lower bounds as (Cut_X). The separation routines in approaches *Most VI* and *All VI* require an additional 10 and 12 seconds on average. Thereby, the average LP gaps are improved from 1.44% ((Cut_Z)+(Cut_X)) to 1.31% (*Most VI* and *All VI*). However, *All VI* does not improve upon *Most VI* significantly.

There is a notable difference in the numbers of valid inequalities that were detected during the LP relaxations of the different settings. By far the most inequalities are found by setting (Cut_Z), even though the obtained LP bound is comparably weak. This is consistent with the long running time of the LP relaxation of setting (Cut_Z). Rather surprising is the fact, that setting *All VI* obtains the same LP bound as setting *Most VI* for 28 out of 30 settings but the number of valid inequalities found by *All VI* is smaller for 22 and larger for only 2 instances.

Table 3 shows the respective gap of the five different settings after 3, 10, 30 and 60 minutes. For these results we calculate the gaps as $(UB_t - LB_t)/UB_t$ where UB_t is the best integer solution found by the respective setting after t minutes and LB_t is the lower bound after t minutes. For each instance and running time the smallest gap of all five settings is indicated in bold. If no integer solution is available after t minutes we indicate this by a dash in the respective column. For each setting and time t the last three lines of the table indicate the mean and median of gaps over the instance set and how often the respective approach gives the smallest gap of all settings.

Contrary to what the LP gaps in Table 2 suggest the setting *All VI* with all valid inequalities enabled outperforms the other settings on a majority of instances. The performance of setting *Most VI* is only slightly worse (0.09%, 0.07% and 0.05% larger gap after 10, 30 and 60 minutes). The other settings perform significantly worse with between 0.46% and 2.58% larger gaps on average.

In Figures 7 and 8 we give a graphical illustration of the numbers reported in Table 3. The coordinates of each mark indicate how many out of 30 instances (ordinate axis) were solved within a given optimality gap (abscissa). Figure 7 shows the performance after 3 and 10 minutes and Figure 8 shows the performance after 30 and 60 minutes.

6. Conclusions

In this paper we introduce the Capacitated Connected Facility Location problem. We introduce various sets of cut set, minimum cardinality, cover and cut-set-cover inequalities to strengthen a basic integer programming model. After a detailed discussion of separation procedures we report the results of our computational experiments. These confirm that the proposed approach finds solution within a small optimality gap averaging to less than 2% for a set of realistic new benchmark instances.

Acknowledgements

Stefan Gollowitzer was supported by the Fonds de recherche du Québec - Nature et technologies under grant 163879 within the *Programme de Stages Internationaux*. This support is gratefully acknowledged.

	(Cut$_Z$)				(Cut$_X$)				(Cut$_Z$)+(Cut$_X$)				Most VI				All VI			
	g_3	g_{10}	g_{30}	g_{60}	g_3	g_{10}	g_{30}	g_{60}	g_3	g_{10}	g_{30}	g_{60}	g_3	g_{10}	g_{30}	g_{60}	g_3	g_{10}	g_{30}	g_{60}
A20-FTTC	7.28	5.44	4.97	4.12	-	-	4.76	4.76	4.25	4.22	4.03	**3.45**	4.34	4.22	4.21	4.18	**3.68**	**3.66**	**3.62**	3.61
A25-FTTC	5.54	5.26	5.04	4.96	-	-	4.99	4.96	5.88	5.82	5.77	5.74	4.44	4.37	4.32	4.29	**3.74**	**3.66**	**3.61**	**3.58**
A30-FTTC	6.65	5.82	5.31	5.17	-	5.81	5.66	5.52	5.72	5.55	4.37	3.93	**4.87**	3.61	**3.31**	3.02	4.93	**3.58**	3.40	**2.86**
B20-FTTC	11.59	5.55	3.44	2.13	0.07	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
B25-FTTC	7.58	4.65	2.25	1.20	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
B30-FTTC	1.48	0.69	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
C20-FTTC	-	15.91	6.21	5.03	-	-	1.94	1.48	-	1.83	1.79	0.61	-	**0.82**	0.59	**0.32**	-	1.00	**0.52**	0.46
C25-FTTC	-	5.85	4.09	3.72	-	1.03	0.99	0.90	0.99	0.93	0.88	0.85	1.06	0.86	0.70	0.66	**0.83**	**0.73**	**0.65**	**0.62**
C30-FTTC	5.22	3.37	3.00	2.68	1.47	1.42	1.33	1.24	1.34	1.19	1.10	1.01	0.95	0.82	0.30	0.05	**0.91**	**0.73**	**0.13**	**0.00**
D20-FTTC	-	-	3.34	2.83	-	-	2.59	2.59	-	2.49	2.46	2.42	-	2.53	**2.00**	**1.90**	-	**2.15**	2.10	2.07
D25-FTTC	4.14	2.37	2.07	1.77	-	-	1.99	1.80	**2.39**	1.61	1.24	1.18	-	1.62	**1.18**	**1.06**	-	**1.60**	1.46	1.13
D30-FTTC	1.53	1.19	0.96	0.89	-	1.31	1.23	0.96	1.23	**0.69**	**0.55**	**0.53**	**1.17**	1.04	0.87	0.59	1.20	0.91	0.60	0.58
E20-FTTC	-	8.32	5.50	4.25	-	1.98	1.77	1.66	1.79	1.43	1.22	1.16	**1.73**	**1.21**	**0.85**	**0.80**	**1.73**	**1.21**	**0.85**	**0.80**
E25-FTTC	4.37	2.56	2.24	1.85	1.99	1.82	0.70	0.67	**0.70**	**0.55**	**0.42**	**0.32**	0.73	0.59	0.46	0.35	0.73	0.59	0.46	0.35
E30-FTTC	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
A20-FTTB	8.31	7.29	5.56	5.21	-	-	5.07	5.05	**5.73**	5.68	5.62	5.49	-	3.17	3.06	2.96	-	**2.70**	**2.53**	**2.43**
A25-FTTB	6.90	6.58	6.27	4.47	-	3.52	3.45	3.37	4.82	4.73	4.59	4.46	3.50	3.26	**3.01**	**2.97**	**3.38**	**3.19**	3.05	3.02
A30-FTTB	6.85	5.22	4.86	4.65	-	6.13	5.98	5.95	5.06	4.88	4.78	4.73	2.94	2.66	**2.29**	**2.10**	**2.84**	**2.63**	2.42	2.34
B20-FTTB	16.17	11.15	6.52	4.58	0.65	0.60	0.55	0.43	0.44	0.37	0.25	0.23	**0.30**	0.27	**0.16**	**0.14**	0.32	**0.22**	0.18	0.16
B25-FTTB	10.99	7.87	5.22	4.11	0.58	0.47	0.38	0.34	0.23	0.10	0.05	0.03	**0.04**	**0.00**	**0.00**	**0.00**	0.06	0.02	**0.00**	**0.00**
B30-FTTB	3.57	1.41	**0.00**	**0.00**	0.02	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**	**0.00**
C20-FTTB	-	-	11.43	8.69	-	-	1.32	1.31	-	1.13	0.88	0.85	-	**0.76**	**0.72**	**0.58**	-	0.80	0.78	0.72
C25-FTTB	-	-	5.43	4.68	-	-	1.84	1.19	-	1.32	1.02	1.00	-	0.89	0.69	0.65	-	**0.85**	**0.61**	**0.46**
C30-FTTB	7.32	2.79	1.16	0.81	-	0.69	0.66	0.62	0.59	0.43	0.37	0.34	0.42	0.35	0.27	0.25	**0.33**	**0.25**	**0.21**	**0.16**
D20-FTTB	-	4.68	2.73	2.25	-	-	-	1.41	-	1.21	1.19	1.17	-	1.39	1.17	1.14	-	**1.19**	**1.07**	**1.04**
D25-FTTB	-	2.96	2.10	1.85	-	-	-	2.29	-	**1.77**	1.72	1.70	-	1.93	1.49	1.45	-	2.28	**1.31**	**1.29**
D30-FTTB	1.40	1.12	0.92	0.80	-	0.83	0.70	0.62	0.87	0.58	0.43	0.41	0.64	0.51	**0.36**	**0.33**	**0.54**	**0.47**	0.40	**0.33**
E20-FTTB	-	10.18	6.09	4.47	-	-	1.66	1.63	2.33	**1.52**	1.36	1.35	2.57	1.79	1.34	**1.22**	**1.61**	1.54	**1.33**	1.31
E25-FTTB	-	3.92	2.39	2.13	-	1.68	0.87	0.71	**0.77**	**0.51**	**0.44**	0.41	1.41	0.62	0.45	**0.28**	0.91	0.57	0.50	0.44
E30-FTTB	2.18	1.15	0.12	0.10	0.40	0.20	0.16	0.15	**0.14**	0.12	0.10	0.08	**0.14**	**0.04**	**0.00**	**0.00**	**0.14**	**0.04**	**0.00**	**0.00**
mean	-	-	3.64	2.98	-	-	-	1.72	-	1.69	1.55	1.45	-	1.31	1.13	1.04	-	**1.22**	**1.06**	**0.99**
median	-	-	3.39	2.75	-	-	-	1.21	-	1.16	0.95	0.85	-	0.84	0.70	0.58	-	**0.83**	**0.61**	**0.52**
best	1	1	3	3	3	5	5	5	10	10	8	8	11	10	16	**18**	**16**	**22**	**19**	**18**

Table 3: Gaps to model upper bound

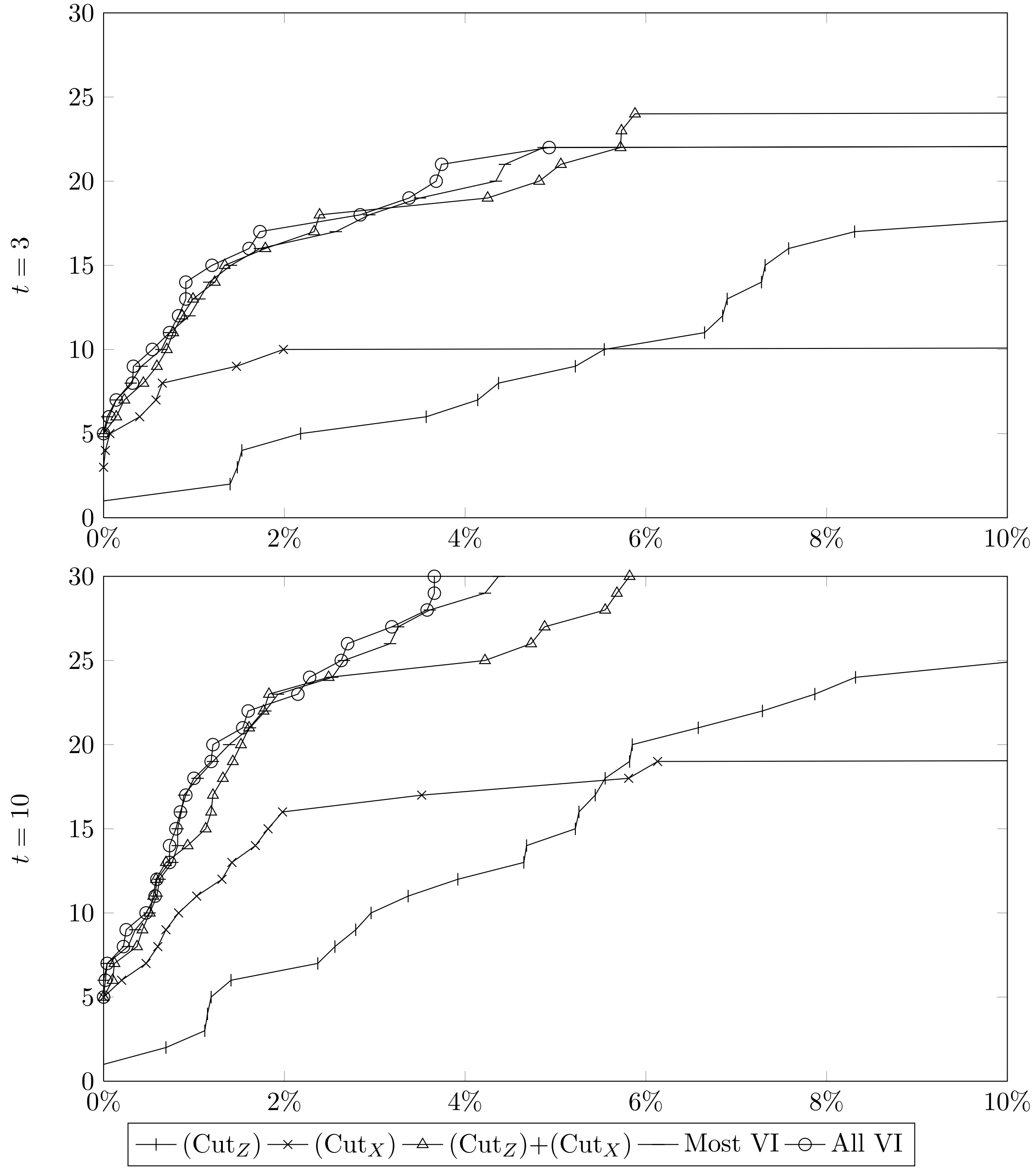


Figure 7: Performance chart for 3 minutes (top) and 10 minutes (bottom) runtime

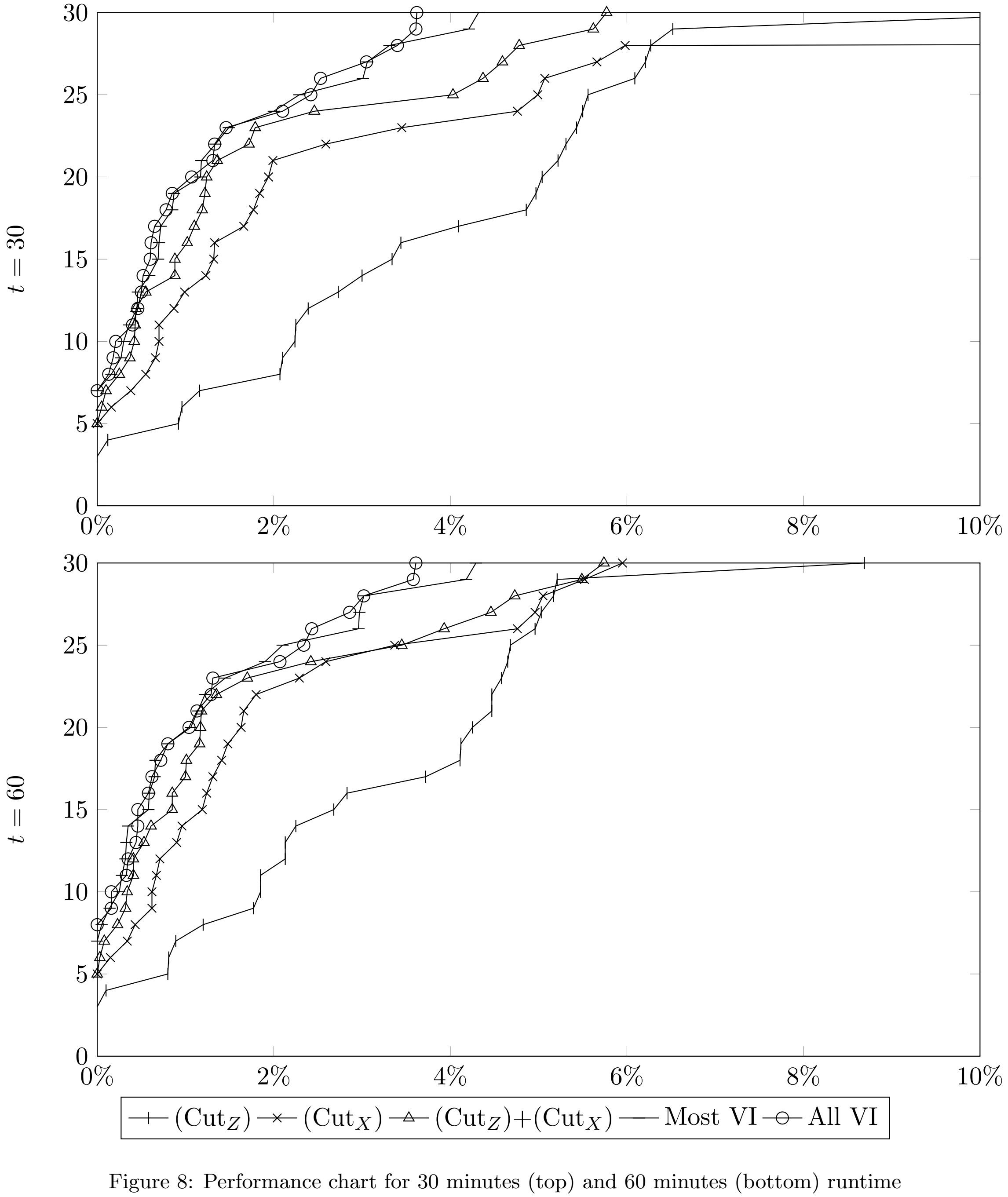


Figure 8: Performance chart for 30 minutes (top) and 60 minutes (bottom) runtime

Bibliography

[1] K. Aardal, Y. Pochet, and L. A. Wolsey. Capacitated facility location: Valid inequalities and facets. *Mathematics of Operations Research*, 20(3):562–582, 1995.

[2] A. Arulselvan, A. Bley, S. Gollowitzer, I. Ljubic, and O. Maurer. MIP modeling of incremental connected facility location. In J. Pahl, T. Reiners, and S. Voß, editors, *INOC*, volume 6701 of *Lecture Notes in Computer Science*, pages 490–502. Springer, 2011.

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

[4] A. Ceselli, F. Liberatore, and G. Righini. A computational evaluation of a general branch-and-price framework for capacitated network location problems. *Annals OR*, 167(1):209–251, 2009.

[5] M. Chouman, T. G. Crainic, and B. Gendron. Commodity representations and cutset-based inequalities for multicommodity capacitated fixed-charge network design. Technical Report CIRRELT-2011-56, Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT), CIRRELT, 2011.

[6] I. A. Contreras and J. A. Díaz. Scatter search for the single source capacitated facility location problem. *Annals OR*, 157(1):73–89, 2008.

[7] A. M. Costa, J.-F. Cordeau, and B. Gendron. Benders, metric and cutset inequalities for multicommodity capacitated network design. *Computational Optimization and Applications*, 42:371–392, 2009.

[8] T. G. Crainic and M. Gendreau. Cooperative parallel tabu search for capacitated network design. *J. Heuristics*, 8(6):601–627, 2002.

[9] T. G. Crainic, A. Frangioni, and B. Gendron. Bundle-based relaxation methods for multicommodity capacitated fixed charge network design. *Discrete Applied Mathematics*, 112(1-3): 73–99, 2001.

[10] T. G. Crainic, B. Gendron, and G. Hernu. A slope scaling/lagrangean perturbation heuristic with long-term memory for multicommodity capacitated fixed-charge network design. *J. Heuristics*, 10(5):525–545, 2004.

[11] Q. Deng and D. Simchi-Levi. Valid inequalities, facets and computational results for the capacitated concentrator location problem. Technical report, Department of Industrial Engineering and Operations Research, Columbia University, New York, 1992.

[12] J. A. Diaz and E. Fernández. A branch-and-price algorithm for the single source capacitated plant location problem. *Journal of the Operations Research Society*, 53:728–740, 2002.

[13] F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Connected facility location via random facility sampling and core detouring. *Journal on Computer and System Sciences*, 76: 709–726, December 2010. ISSN 0022-0000.

[14] B. Gendron, T. G. Crainic, and A. Frangioni. Multicommodity capacitated network design. In B. Sanso and P. Soriano, editors, *Telecommunications Network Planning*, pages 1–19. Kluwer Academics Publisher, 1998.

[15] I. Ghamlouche, T. G. Crainic, and M. Gendreau. Cycle-based neighbourhoods for fixed-charge capacitated multicommodity network design. *Operations Research*, 51(4):655–667, 2003.

[16] I. Ghamlouche, T. G. Crainic, and M. Gendreau. Path relinking, cycle-based neighbourhoods and capacitated multicommodity network design. *Annals OR*, 131(1-4):109–133, 2004.

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

[18] M. Hewitt, G. L. Nemhauser, and M. W. P. Savelsbergh. Combining exact and heuristic approaches for the capacitated fixed-charge network flow problem. *INFORMS Journal on Computing*, 22(2):314–325, 2010.

[19] K. Holmberg and D. Yuan. A lagrangian heuristic based branch-and-bound approach for the capacitated network design problem. *Operations Research*, 48:461–481, 2000.

[20] K. Holmberg, M. Rönnqvist, and D. Yuan. An exact algorithm for the capacitated facility location problems with single sourcing. *European Journal of Operational Research*, 113(3):544 – 559, 1999.

[21] IBM. CPLEX, November 22nd 2011. `http://www.ilog.com/products/cplex/`;.

[22] K. Kaparis and A. N. Letchford. Separation algorithms for 0-1 knapsack polytopes. *Mathematical Programming*, 124(1-2):69–91, 2010.

[23] G. Kliewer and L. Timajev. Relax-and-cut for capacitated network design. In G. S. Brodal and S. Leonardi, editors, *ESA*, volume 3669 of *Lecture Notes in Computer Science*, pages 47–58. Springer, 2005. ISBN 3-540-29118-0.

[24] M. Leitner and G. R. Raidl. Branch-and-cut-and-price for capacitated connected facility location. *Journal of Mathematical Modelling and Algorithms*, 10(3):245–267, 2011.

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

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

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

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