

The Incremental Connected Facility Location Problem

Ashwin Arulsevan^a, Andreas Bley^b, Ivana Ljubić^c

^a*Department of Management Science, University of Strathclyde, ashwin.arulsevan@strath.ac.uk*

^b*Institute for Mathematics, University of Kassel, andreas.bley@uni-kassel.de*

^c*ESSEC Business School of Paris, IDS Department, ivana.ljubic@essec.edu*

Abstract

We consider the incremental connected facility location problem (incremental ConFL), in which we are given a set of potential facilities, a set of interconnection nodes, a set of customers with demands, and a planning horizon. For each time period, we have to select a set of facilities to open, a set of customers to be served, the assignment of these customers to the open facilities, and a network that connects the open facilities. Once a customer is served, it must remain served in subsequent periods. Furthermore, in each time period the total demand of all customers served must be at least equal to a given minimum coverage requirement for that period. The objective is to minimize the total cost for building the network given by the investment and maintenance costs for the facilities and the network summed up over all time periods.

We propose a mixed integer programming approach in which, in each time period, a single period ConFL with coverage restrictions has to be solved. For this latter problem, which is of particular interest in itself, new families of valid inequalities are proposed: these are set union knapsack cover (SUKC) inequalities, which are further enhanced by lifting and/or combined with cut-set inequalities, which are primarily used to ensure connectivity requirements. Details of an efficient branch-and-cut implementation are presented and computational results on a benchmark set of large instances are given, including examples of telecommunication networks in Germany.

Keywords: Mixed Integer Programming, Facility Location, Branch-and-Cut, Multi-Period Network Design, Incremental Network Design

1. Introduction

In many network optimization problems that deal with the infrastructure planning, the underlying network is constructed in multiple stages over a given period of time. Thereby, certain network properties in the intermediate stages have to be respected – most frequently, operational requirements are imposed to partially built subnetworks in each time period. In this family of network optimization problems, known as *incremental network design*, simultaneous decisions involving network design and scheduling have to be made. Incremental network design of well-known and polynomially solvable problems, like shortest paths, spanning trees and maximum flows, has been studied in (Baxter et al., 2014; Engel et al., 2017; Kalinowski

et al., 2015), respectively. Further examples of applications dealing with the integrated network design and scheduling are known in the design of transportation networks or restoration of networks after natural disasters, see, e.g., (Averbakh and Pereira, 2015; Cavdaroglu et al., 2013), respectively, and further references therein.

The problem we consider is an optimal design of a network topology in the context of a multi-period planning of local access networks. Design of local access networks is frequently modeled as the *connected facility location problem*, which combines the classical facility location problem with the Steiner tree problem (i.e., one has to decide about opening a set of facilities, allocating customers to open facilities and finally connecting them through a Steiner tree), see, e.g. Gollowitzer and Ljubić (2011); Leitner et al. (2017a). In our scheduling environment, the local access network is built over multiple periods, and in each time period multiple edges can be added to the network and multiple facilities can be opened. We optimize the cumulative cost over the planning horizon, while a prescribed level of coverage has to be ensured in each of the time periods.

More precisely, a telecommunication company wants to increase the speed of broadband connections by combining fiber optic technology with existing copper connections, i.e., by means of the *Fiber-to-the-Curb* (FTTC) technology. Street segments along which fiber optic cables can be installed, determine the *core network*. Potential optical and *existing copper cables* intersect at locations where multiplexor devices can be installed. Between a multiplexor and an end-customer, the existing copper connection is used. The existing copper paths are pre-processed for building an *assignment network* whose edges are assignment links between potential multiplexor locations and end-customers. To build an FTTC network, one has to decide on which locations to install multiplexor devices so that each end-customer is assigned to a multiplexor, and each multiplexor is connected to the *central office* by a fiber optic path. This problem has been modeled as the Connected Facility Location Problem (ConFL), see, e.g. Gollowitzer and Ljubić (2011).

Due to the huge investment costs needed to build an FTTC network, network providers may decide to do the deployment in several stages. The company takes the strategic decision of fixing a minimal level of service provision that shall be ensured in each time period. In our context, this is the percentage of *customer demands* that should be served at each of the stages (see, e.g. Wassermann (2012)). Thereby, demand of a customer is defined as the number of end-subscribers (e.g., offices and/or households) behind the customer's address. The coverage of customer demands need to be increased over time. Other relevant applications of ConFL include the design of Fiber-to-the-Air networks or content distribution networks, see, e.g. Havet and Wennink (2004).

To model this problem we introduce the *Incremental Connected Facility Location Problem* denoted as *incremental ConFL*, or iConFL. We first define the *Connected Facility Location Problem* ConFL introduced by Gollowitzer and Ljubić (2011) and explain how this can be extended to iConFL. In ConFL, we are given

a rooted graph, whose set of nodes V is partitioned into three subsets: a set of potential facilities F , a set of customers R , and a set of potential Steiner nodes $V \cap (R \cup F)$. The root node $r \in V \cap R$ represents a central office with an uplink to the backbone network. We denote by $S = V \cap R$ the set of nodes that build the *core network*. Potential connections among the nodes in S are given as the undirected edge set E_S . The corresponding directed arc set is $A_S = \{(i;j);(j;i) \mid ij \in E_S\}$. Possible connections between the facilities F and the customers R are given by the edges $E_R \subseteq F \times R$, which define the directed arc set $A_R = \{(i;j) \in F \times R \mid ij \in E_R\}$. Note that it is sufficient to consider only arcs directed from facilities to customers. We let $A = A_S \cup A_R$ and $E = E_S \cup E_R$. In addition, we are given *setup costs* for edges $c: E \rightarrow \mathbb{R}_+$ and facilities $g: F \rightarrow \mathbb{R}_+$ for *opening* the edge or facility. We are also given a *maintenance costs* for edges $m: E \rightarrow \mathbb{R}_+$ and facilities $m_f: F \rightarrow \mathbb{R}_+$. We are interested in determining the set of facilities to be opened, edges to be used to connect the open facilities and an assignment of the open facilities to the customers in order to service them such that the associated cost is minimized. We require that all customers are provided service by some open facility.

In the incremental problem variant iConFL, we consider a planning horizon given as a discrete set of time periods $T = \{1, \dots, T\}$, $T > 1$. We are given customer *demands* $d: R \rightarrow \mathbb{Z}_+$ and a *minimum coverage requirement* $D^t \in \mathbb{Z}_+$ for each time period $t \in T$ that needs to be satisfied. The setup costs will be incurred once when the edge or a facility is opened for the first time and the maintenance costs will be incurred for each period an edge or a facility is actually *used*. The individual time periods do not have to be necessarily equally long and this could be modeled by time variant costs and all our results would still hold. We seek a schedule that, for each time period, identifies a subset of facilities that can provide service, a subset of customers to be served, an assignment of every served customer to a serving open facility, and the set of edges in the core network that connects all open facilities. In each time step, the total demand of the served customers must satisfy the corresponding minimum coverage requirement and the chosen edges in E_S must form a network connecting the open facilities with the root node. Once a customer is served, it has to be served for the remainder of the planning periods. The goal is to minimize the total costs of the network, defined as the sum of investment and maintenance costs over all periods.

The coverage requirements is considered for two practical reasons. From a planning perspective, infrastructural projects are carried out in phases and the blueprints are usually provided for the components that have to be built in each phase. The coverage requirements indicate what percentage of the network should be built within a phase. From a cost perspective, the facility location problem with outliers has been studied (see Cordeau et al. (2019) and Charikar et al. (2001)), where it has been observed that a significant amount of cost will be incurred in serving a small percentage of customers who are located far away from the majority of the customers. If we were to provide an incremental solution, we would be interested in identifying such outlying customers who could be served at a later time period in order save cost.

Figure 1 illustrates an input graph and a feasible solution for four time periods. Notice that there is a facility that is open in the second period, which is then used only as a Steiner node in the later periods. Also notice that a served customer cannot be denied service in a future period, but it can be reassigned to other facilities. As indicated in this example, in each subsequent period, new customers can be served, new facilities can be opened, and new edges in the core network can be built. Some of the previously opened facilities can later be used as Steiner nodes only.

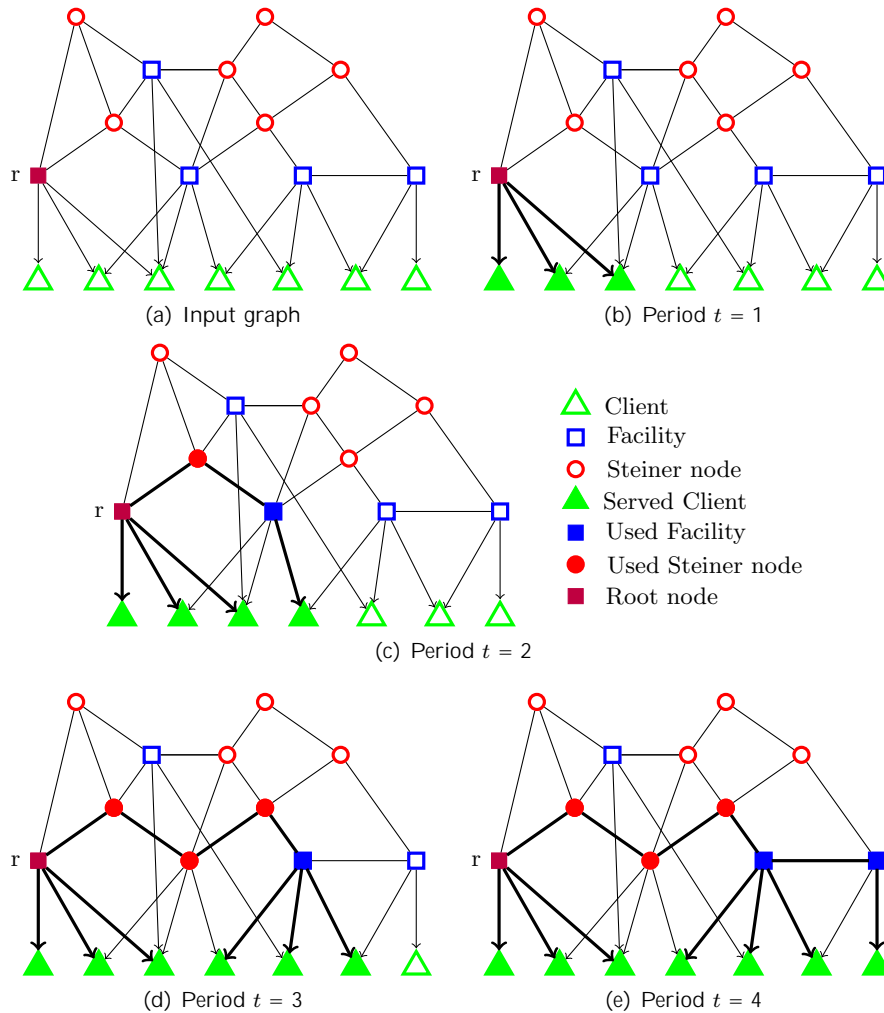


Figure 1: Input graph and the schedule for serving clients, for opening and connecting facilities for $T = 4$. Bold edges indicate the edges used in the corresponding time period.

Our Contribution

We propose a mixed integer programming approach to this new problem that has not been studied in the previous literature. We discuss some valid inequalities to strengthen the basic formulation. We also study the problem associated to a single period ConFL with coverage restrictions, which is of particular interest by

itself. For the latter problem we derive new families of valid inequalities: set union knapsack cover (SUKC) inequalities, cut-set-SUKC inequalities and lifted SUKC constraints. Details of an efficient branch-and-cut implementation are presented and computational results on a set of benchmark instances and a set of large realistic instances are given.

The paper is organized as follows. After providing the literature review in the remainder of this section, we present a mixed integer programming (MIP) formulation in Section 2. In Section 3 we discuss various classes of strengthening inequalities. These inequalities may also be used to strengthen the formulations of the underlying single-period ConFL with coverage restriction. Section 4 provides a description of the specific separation subroutines and Section 5 outlines the branch-and-cut procedure. In Section 6 we describe the benchmark data sets, details of our implementation of the branch-and-cut algorithm, and results of our computational experiments.

1.1. Literature Review

To the best of our knowledge, our article is the first study related to the incremental ConFL problem – preliminary computational results related to this work have only appeared in Arulselvan et al. (2011). In this section we provide a short overview of the literature related to the iConFL. We divide the related work into problems dealing with multi-period optimization (facility location, network design) and those dealing with single-period ConFL variants.

Related Multi-Period Optimization Problems. The facility location problem over time is a well-studied problem (see, e.g., a survey of Owen and Daskin (1998)). In a recent work, Albareda-Sambola et al. (2009) considered a multi-period *incremental facility location problem*, where the coverage of customer demand has to be increased over time. The authors combined subgradient optimization and a Lagrangian approach and generated feasible solutions with a Lagrangian based heuristic. Arulselvan et al. (2015) gave an algorithm for the incremental facility location problem, which outputs a sequence of open facilities and served customers along with their assignment that has a constant competitive factor of 8. Their problem does not require the open facilities to be connected via a backbone network and the framework will not provide the same guarantee when extended to iConFL. In fact, a competitive ratio similarly defined for iConFL setting will be unbounded in the worst case scenario.

There has been intense research on multi-period *network design* problems since publication of the seminal articles by Christofides and Nicos (1974), Doulliez and Rao (1975) and Zadeh (1974). Optimization methods have been used for an incremental design of networks for telecommunication and transportation (Ukkusuri and Patil, 2009), distribution of gas or water (Suhl and Hilbert, 1998), and many others. Recent references dealing with integrated network design and scheduling are (Averbakh and Pereira, 2015; Baxter et al., 2014; Cavdaroglu et al., 2013; Engel et al., 2017; Kalinowski et al., 2015). Recent contributions concerning the

applications in the telecommunications sector are given by Bienstock et al. (2006) and Lardeux and Nace (2007).

Single-Period Connected Facility Location. Early work on ConFL mainly includes approximation algorithms. The problem can be approximated within a constant ratio and the currently best-known approximation ratio is provided by Eisenbrand et al. (2010). Heuristic approaches to ConFL are given in Ljubić (2007); Tomazic and Ljubić (2008); Bardossy and Raghavan (2010). Gollowitzer and Ljubić (2011) studied MIP formulations for ConFL, both theoretically and computationally. The authors provided a complete hierarchy of ten MIP formulations with respect to the quality of their LP bounds. A polyhedral study of the ConFL polytope is given in Leitner et al. (2017b). The problem was later computationally investigated (along with several other tree-star problems) in Leitner et al. (2017a). The authors proposed a dual ascent procedure for the Steiner arborescence problem within a branch-and-cut algorithm to improve the quality of primal and dual bounds and warm start the branch-and-cut procedure. A bi-objective version of ConFL in which there are two conflicting objectives, namely the minimization of the investment cost and maximization of the covered demand, was computationally tested in Leitner et al. (2016). Capacitated versions of ConFL are studied in Gollowitzer et al. (2013); Leitner and Raidl (2011a,b, 2008).

Recently, Cordeau et al. (2019) studied the (single-period) facility location problem with coverage restrictions and proposed an efficient Benders decomposition approach for solving it to optimality. In contrast to the incremental ConFL studied in this paper, the problem studied in Cordeau et al. (2019) does not impose connectivity between open facilities, and there are no assignment costs between customers and open facilities.

2. MIP Modeling

In this section we present a (generic) MIP formulation for the iConFL problem. We use the following notation: For a given $W \subseteq V$, let $\mathcal{A}(W) := \{(i,j) \in \mathcal{A} \mid j \in W, i \notin W\}$. For a given $j \in R$, let F_j denote the set of facilities that can serve j , i.e., $F_j := \{i \in F \mid (i,j) \in \mathcal{A}_{RG}\}$. For a given $J \subseteq R$, let $F_J := \bigcup_{j \in J} F_j$ and $D(J) := \sum_{j \in J} d_j$ and, finally, for a given $I \subseteq F$, let R_I be the subset of customers served by I , i.e., $R_I := \{j \in R \mid \exists i \in I \text{ with } (i,j) \in \mathcal{A}_{RG}\}$, and let $R_i := R_{\{i\}}$. We say that a vector a , a solution of an LP-relaxation of a given model, is *minimal feasible* w.r.t. this model iff there does not exist another feasible LP-solution \hat{a} such that $\hat{a} \leq a$ and $\hat{a} \neq a$.

In order to describe which customers are served and which facilities are used at each time period, we introduce binary variables y^t and z^t , respectively, for each time period (t.p.) $t \in T$. These variables are

interpreted as

$$y_j^t = \begin{cases} \leq 1; & \text{if } j \text{ is served in t.p. } t, \\ > 0; & \text{otherwise, and} \end{cases} \quad j \in R; t \in T; \quad z_i^t = \begin{cases} \leq 1; & \text{if } i \text{ is used in t.p. } t, \\ > 0; & \text{otherwise.} \end{cases} \quad i \in F; t \in T;$$

The assignment of the served customers to the open facilities and the network connecting the open facilities to the root node are modeled together by the binary arc variables x^t for all $t \in T$, which are interpreted as

$$x_{ij}^t = \begin{cases} \leq 1; & \text{if arc } (i;j) \text{ is used in t.p. } t, \\ > 0; & \text{otherwise.} \end{cases} \quad (i;j) \in A; t \in T;$$

To describe the initial opening of facilities and edges (in which case we need to pay the setup costs), we also introduce binary facility variables z^t and binary aggregated edge and arc variables \bar{x}^t :

$$\bar{z}_i^t = \begin{cases} \leq 1; & \text{if facility } i \text{ is opened for the first time in t.p. } t, \\ > 0; & \text{otherwise and} \end{cases} \quad i \in F; t \in T;$$

$$\bar{x}_e^t = \begin{cases} \leq 1; & \text{if core edge } e \text{ is opened for the first time in t.p. } t, \\ > 0; & \text{otherwise.} \end{cases} \quad e \in E_S; t \in T;$$

$$\bar{x}_{ij}^t = \begin{cases} \leq 1; & \text{if arc } (i;j) \text{ is opened for the first time in t.p. } t, \\ > 0; & \text{otherwise.} \end{cases} \quad (i;j) \in A_R; t \in T;$$

Observe that variables \bar{x}_e^t are associated with edges instead of arcs of the core network for the following reason: In the general case, a facility $i \in F$ may be opened in period $t \in T$, and closed in period $t+k \in T$ ($k > 0$). Consequently, an arc that was oriented in direction from i to j in period t , may be used in the opposite direction in period $t+k$. Since the edge opening costs need to be paid only once, we have to leave the direction of set-up variables \bar{x} unspecified. We denote by $(X; Y; Z)$ the vectors $(x^t; y^t; z^t)$ concatenated over all periods $t \in T$, i.e., $(X; Y; Z) := (x^t; y^t; z^t)_{t \in T}$.

With these variables and notations, the objective function of the incremental ConFL problem can be formulated as follows:

$$\mathbf{f}(\bar{x}; \bar{z}; X; Y; Z) = \prod_{t=1}^T (1 + \sum_{e \in E_S} c_e \bar{x}_e^t + \sum_{(i;j) \in A_R} c_{ij} \bar{x}_{ij}^t + \sum_{i \in F} g_i \bar{z}_i^t + \sum_{(i;j) \in A} m_{ij} x_{ij}^t + \sum_{i \in F} m_i z_i^t)^5 \quad (1)$$

For each period $t \in T$, the objective function comprises of the set-up cost for those elements of network in use in t that were not installed in periods $1; \dots; t-1$ plus the maintenance cost for the network used to serve selected customers. The parameter β specifies a discount factor that we multiply with the cost to obtain the net present value.

To describe a feasible solution, we notice that for each period $t \geq T$, the vector $(x^t; y^t; z^t)$ must be a feasible solution of the *ConFL with Coverage Restrictions* (ConFL-CR) problem. Solutions of this problem are binary vectors that satisfy the following properties:

- (i) For the subset of customers R^t that are served in period t , we have that $D(R^t) = D^t$.
- (ii) Each customer in R^t is connected to a facility which is used in period t .
- (iii) All used facilities are connected to the root node over arcs that are used in period t .

Let $P(D^t)$ denote the convex hull of all feasible ConFL-CR solutions (see Section 2.1). The following generic MIP formulation models then the iConFL problem:

$$\begin{aligned}
(iConFL) : \quad & \min f(\bar{x}; \bar{z}; X; Y; Z) \\
\text{s.t.} \quad & x_{ij}^t + x_{ji}^t \leq \sum_{k=1}^{|E|} x_e^k \quad e = fi; jg \in E_S; t \geq T \quad (2) \\
& x_{ij}^t \leq \sum_{k=1}^{|A|} x_{ij}^k \quad (i; j) \in A_R; t \geq T \quad (3) \\
& z_i^t \leq \sum_{k=1}^{|F|} z_i^k \quad i \in F; t \geq T \quad (4) \\
& y_j^t \leq y_j^{t-1} \quad j \in R; t \geq T \quad (5) \\
& (x^t; y^t; z^t) \in P(D^t) \quad t \geq T \quad (6) \\
& \bar{x}_e^t, \bar{z}_i^t, x_{ij}^t, y_j^t, z_i^t \in \{0, 1\} \quad e \in E; i \in F; (i; j) \in A; j \in R; t \geq T \quad (7)
\end{aligned}$$

Inequalities (2), (3) and (4) ensure that we have to *open* edges, assignment arcs and facilities, respectively, as soon as they are *used*. Inequalities (5) enforce the continuance of service for each customer (i.e., if customer j was served for the first time in period $t \geq T$, it also remains served in all periods $t+1; \dots; T$). Finally, constraint (6) states that $(x^t; y^t; z^t)$ is a feasible solution of the ConFL-CR problem in each period $t \geq T$. In the following we first provide an MIP model for the ConFL-CR problem and then, in Section 3, we study strengthening inequalities that are valid for both, ConFL-CR and iConFL.

2.1. Single-Period ConFL With Coverage Restrictions (ConFL-CR)

There are many ways to describe a MIP formulation of the ConFL-CR polytope $P(D^t)$. Each of the MIP models for standard ConFL given in Gollowitzer and Ljubić (2011) can be generalized by introducing variables associated to the set of customers to derive the corresponding models for $P(D^t)$. In the following, we restrict ourselves to the best performing model (in practice) according to Gollowitzer and Ljubić (2011) (see also Leitner et al. (2017a)), which utilizes the family of cut set constraints of exponential size to ensure the connectivity between facilities and the root node. For a single period $t \geq T$ by substituting

$(x; y; z) = (x^t; y^t; z^t)$ the following constraints define the $P(D^t)$ polytope:

$$\bigtimes_{i \in F_j} x_{ij} = y_j \quad j \in R \quad (8)$$

$$x_{ij} \leq z_i \quad (i; j) \in A_R \quad (9)$$

$$x(\delta^-(W)) \leq z_i \quad W \text{ nfrg}; i \in W \setminus F \notin; \quad (10)$$

$$\bigtimes_{j \in R} d_j y_j \leq D^t \quad (11)$$

In other words, we have

$$P(D^t) = \text{conv} \left\{ (x; y; z) \geq 0; \sum_{j \in R} d_j y_j \leq D^t; \text{for } j \in R, (x; y; z) \text{ satisfy (8)-(11)} \right\}$$

Constraints (8) model the fact that a customer is served only if there is a facility connected to it. Constraints (9) enforce that a facility is *used* if it serves a customer. Constraint set (11) expresses the minimum demand coverage requirement for each time period. Finally, the exponentially large constraint set (10) ensures that, in each time period, all used facilities are connected to the root node. The inequalities in constraint set (10) enforce that for every subset $W \subseteq S$ that includes a facility i and does not include the root node r , at least one of the arcs in the set of all incoming arcs in W must be *used* if facility i is used. The following section investigates valid inequalities of this polytope that are later on exploited in our computational framework for solving iConFL. Whenever it is clear from context, we omit the index for the time period t and refer to $P(D^t)$ as $P(D)$.

3. Valid Inequalities for the ConFL-CR Polytope

Since the ConFL-CR polytope is contained in the convex hull of all feasible iConFL solutions, valid and facet defining inequalities for $P(D^t)$ carry over to valid and strong valid inequalities for the iConFL polytope. In this section we focus on valid inequalities for $P(D^t)$ that can strengthen the previous model given by (8)-(11), and therefore also improve lower bounds for iConFL.

3.1. Connectivity Cuts

Instead of enforcing at least one arc in each directed cut that separates a chosen facility from the root node, as done by constraints (10), we may model the connectivity constraints by enforcing at least one arc in every directed cut that separates a chosen customer from the root node:

$$x(\delta^-(W)) \leq y_j \quad W \text{ nfrg}; j \in W \setminus R \notin; \quad (12)$$

This leads to the following alternative formulation for the iConFL.

Lemma 1. *Replacing constraints (10) by (12) in the MIP model (8)-(11) for $P(D^t)$ (and in the corresponding iConFL model) implies (strictly) stronger lower bounds for iConFL.*

Proof. We first show that for the $P(D^t)$ polytope, constraints (12), together with (8), (9) imply constraints (10). Consider an arbitrary inequality of type (10) defined by a cut set $W = \text{Snfrg} = V \cap (\text{frg} \cap R)$ between r and a particular $i \in F$ in time period $t \in T$. Let $(X; Y; Z)$ be a minimal LP-optimal solution of the iConFL model in which connectivity is modeled using (12). It is not difficult to see that in this case, the projected vector $(x^t; z^t)$ (further denoted by $(x; z)$) is minimal feasible, and therefore, for $i \in F$, there exists a customer $j \in R(i)$ such that $z_i = x_{ij}$. Indeed, this latter equality holds true, since, otherwise, if for all customers $j \in R(i)$ we have $z_i > x_{ij}$, the solution is not minimal feasible. Let $W_y = W \cap \{fjg\}$ be another cut set between r and j . Due to (12), we have $x(W_y) = x(W) + \sum_{i \in F_j \cap W_y} x_{ij} - y_j$. Since $y_j = \sum_{i \in F_j \cap W_y} x_{ij} + \sum_{i \in F_j \setminus W_y} x_{ij}$, we can rewrite the previous inequality as $x(W) - \sum_{i \in F_j \setminus W_y} x_{ij} - x_{ij} = z_i$.

Finally, one can easily construct a fractional solution which is feasible for constraints (10) and infeasible for (12), and problem instances which imply a strictly stronger lower bound of the latter model. \square

3.2. Set Union Knapsack Cover (SUKC) Inequalities

This is a new family of strengthening inequalities arising from coverage requirements and the fact that the bipartite graph between facilities and customers is not complete. We call a subset of facilities $I \subseteq F$ a *cover* if its complement $\bar{I} = F \setminus I$ cannot serve enough customers to satisfy the minimal demand requirement, i.e., if $D(R_{\bar{I}}) < D$. Obviously, in that case, at least one facility from each cover I has to be open in any feasible solution. Denoting the family of all covers for the demand requirement D by $COV_F \subseteq 2^F$, this implies that the *set union knapsack cover inequalities*

$$z(I) \geq 1 \quad I \in COV_F \quad (13)$$

are valid for $P(D)$. It is not difficult to construct examples where the addition of SUKC inequalities (13) strengthens the LP-relaxation of $P(D)$. However, not all SUKC inequalities are useful.

We call $I \in COV_F$ an (inclusion-wise) *minimal cover* if I itself is a cover, but no proper subset of I is a cover, i.e., if $D(R_{\bar{I}}) < D$ and $D(R_{\bar{I} \setminus \{f\}}) \geq D$, for all $f \in I$. As in the case of traditional knapsack cover inequalities, the SUKC inequality (13) for any non-minimal cover I is dominated by the SUKC inequality for any minimal cover $I_{\min} \subseteq I$ and these are the strongest possible inequalities of this type. Also note that any non-minimal cover I can be easily turned into a minimal cover by iteratively removing all those facilities, whose removal still results in a cover. However, the minimality of a cover I is not sufficient for (13) to induce a facet of $P(D)$. Even minimal SUKC inequalities (13) define facets only for faces (i.e., restrictions) of $P(D)$ in general. Strong and facet defining inequalities for the full polytope $P(D)$ may be obtained via sequential or simultaneous lifting of minimal SUKC inequalities, which is discussed in Section 4.2.

The name SUKC inequalities comes from the fact that the separation problem for (minimal and lifted minimal) SUKC requires solving the *Set Union Knapsack Problem* (SUKP) which is defined as follows:

Definition 1 (SUKP). *The Set Union Knapsack Problem (SUKP) comprises of a set of elements $U = \{1, \dots, n\}$ and a set of items $S = \{1, \dots, m\}$. Each item, $i = 1, \dots, m$, corresponds to a subset of elements that we denote by S_i . Each item has a nonnegative profit given by $p : S \rightarrow \mathbb{R}_+$ and each element has a nonnegative weight given by $w : U \rightarrow \mathbb{R}_+$. For a subset $S^0 \subseteq S$, we define the weighted union of set S^0 as $W(S^0) = \sum_{u \in \bigcup_{i \in S^0} S_i} w_u$ and its profit as $P(S^0) = \sum_{i \in S^0} p_i$. The goal is to find a subset of items $S \subseteq S$ such that $P(S)$ is maximized and $W(S) \leq B$, for a given knapsack capacity $B > 0$.*

Using the \hat{z}_i -values of a given fractional vector $(\hat{x}; \hat{y}; \hat{z})$ -vector as facility weights, the separation problem for SUKC inequalities is to find a facility set \bar{I} of maximum weight, which is the complement of the cover I , that can only serve a total demand strictly less than D , even if it serves all customers adjacent to \bar{I} . If $\hat{z}(\bar{I}) > \hat{z}(F) - 1$, then the SUKC inequality (13) for I is violated, otherwise all (non-lifted) SUKC inequalities are satisfied. Interpreting the customers $j \in R$ as elements with weights d_j and the facilities $i \in F$ as items with profits \hat{z}_i , and each item i consisting of all elements in R_i , it is not difficult to see that finding the complement of the cover I , namely the set \bar{I} of maximum profit, corresponds to solving the SUKP with the budget $B := D - 1$. In Goldschmidt et al. (1994), it has been shown that SUKP is strongly NP-hard.

In Section 4 we provide more advanced ways to utilize SUKC cuts within our solution framework. In particular, we discuss their liftings and possible exact and heuristic separation algorithms.

3.3. Other Valid Inequalities

In the sequel we present three additional types of valid inequalities: the first two are based on minimum cardinality (MC) arguments, whereas the third one is a family of valid inequalities obtained from knapsack covers.

MC Inequality on Customers. Let l_c be the lower bound on the number of customers that need to be served in order to satisfy the demand D . Then, the following *minimum cardinality inequality* is valid for $P(D)$:

$$y(R) \geq l_c \quad (14)$$

The lower bound l_c can be obtained in polynomial time by sorting the customers in non-increasing order according to d_j . The first l_c customers whose cumulative demand is greater than or equal D determine the number l_c .

MC Inequality on Facilities. Let l_f denote the lower bound on the number of facilities that need to be opened in order to serve the demand of D . Then, the following inequality is valid for $P(D)$:

$$z(F) \geq l_f \quad (15)$$

In contrast to the previous bound l_c , it is NP-hard to calculate l_f : given the bipartite graph with possible assignments between F and R , finding l_f is equivalent to determining the minimal number of facilities to

open, such that the total demand of the union of the customers that can be served by those facilities is at least D . Note however that (15) is just a single inequality so that the value of l_F can be pre-computed for a given instance without a significant computational overhead.

Knapsack Cover Inequalities on Customers. These are traditional knapsack cover inequalities that can be derived from the knapsack constraints (4): Let J be the minimal subset of customers such that its complement $\bar{J} = R \setminus J$ cannot satisfy the whole demand, i.e., $D(\bar{J}) < D$ and $d_k + D(\bar{J}) \geq D$, for any $k \in J$. The set J is called the *minimal cover* with respect to coverage requirement D . Let COV_R be the collection of all minimal covers with respect to D . Then, the following *Knapsack cover inequalities* are valid for $P(D)$:

$$y(J) \leq 1 \quad J \in COV_R \quad (16)$$

These inequalities can further be strengthened as described in Kaparis and Letchford (2010) (see also Balas and Zemel (1978)).

3.4. Strengthened Cut-Set Inequalities

We now propose two types of inequalities that are obtained by combining the cut-set inequalities with some of the cover constraints presented above.

The set of SUKC inequalities (13) implies the following, exponentially large family of cut-set inequalities, that we refer to as *cut-set-SUKC inequalities*:

$$x(I \setminus (W)) \leq 1 \quad I \in COV_F; W \subseteq \text{frg}; I \cap W = \emptyset \quad (17)$$

These inequalities state that we have to establish a path between the root and at least one of the facilities from the set I . It is not difficult to see that the cut-set-SUKC inequalities (17) are not implied by the standard cut-set inequalities (10) and (12) nor by the SUKC inequalities (13).

Similarly, from the cover inequalities on the subsets of customers (16) valid for each subset $J \in COV_R$, we can also derive the corresponding customer-based cut-set-cover inequalities:

$$x(I \setminus (W)) \leq 1 \quad J \in COV_R; W \subseteq \text{frg}; J \cap W = \emptyset \quad (18)$$

Once the set $I \in COV_F$ (respectively, $J \in COV_R$) becomes known, the separation of all three types of inequalities can be done in polynomial time by means of a maximum flow algorithm, see Section 5.4. Determining the sets I or J , however, is NP-hard.

4. Separation and Lifting of SUKC Inequalities (13)

In the following, we provide a detailed description of the separation procedures for the SUKC inequalities introduced in the previous section, and we discuss possible lifting procedures. To the best of our knowledge, these results are new and they are therefore treated separately from the (standard) algorithms for the separation of cut-set-related inequalities (whose implementation details can be found in Section 5).

4.1. Separation of SUKC Inequalities

As discussed in Section 3.2, finding a violated inequality of type (13) requires solving the NP-hard set union knapsack problem. Therefore, in the following, we propose two ways to separate these cuts: an exact approach, which relies on solving a compact ILP formulation, and an alternative heuristic approach.

Exact Separation Algorithm

Let $(\hat{x}; \hat{y}; \hat{z})$ be the current solution of the LP-relaxation of $P(D)$. To model the separation problem as an ILP, we introduce variables $z_i \in \{0, 1\}$ for all $i \in F$ indicating which facilities are contained in the cover I and variables $w_j \in \{0, 1\}$ for all $j \in R$ indicating which customers can be served by facilities not in I . Clearly, a cover I that maximizes the violation of inequality (13) corresponds to an optimal solution of the following integer program:

$$\min \sum_{i \in F} \hat{z}_i \sum_{j \in R} d_j w_j - D - 1; \quad w_j = 1 - \sum_{i \in \mathcal{S}(i; j) \cap A_R} z_i \quad (z_i; w_j) \in \{0, 1\}^{F+R} \quad (19)$$

Inequalities $w_j = 1 - \sum_{i \in \mathcal{S}(i; j) \cap A_R} z_i$ guarantee that all clients that have at least one neighboring facility not in I are served, while constraint $\sum_{j \in R} d_j w_j - D - 1$ ensures that the total demand of all served clients is strictly less than the demand required to meet the coverage constraint. Together, these constraints ensure that, for any integer solution of this problem, the set of facilities i with $z_i = 1$ forms a cover. Note that the objective value of a solution of (19) is equal to the left hand side of the corresponding SUKC inequality (13) for the current LP solution $(\hat{x}; \hat{y}; \hat{z})$. Finding a violated SUKC inequality thus is equivalent to finding a solution of (19) with objective value strictly less than 1. If $(\hat{z}; \hat{w})$ is the optimal solution of the above ILP whose optimal solution is less than one, the associated violated SUKC inequality is given as $\sum_{i \in F} \hat{z}_i z_i - 1$.

Heuristic Separation Algorithm

In practice, it may be too time consuming to exactly separate SUKC inequalities, i.e., to solve the integer linear program (19) in each separation attempt. Instead, we rely on a simple greedy heuristic that constructs potentially violated SUKC inequalities. The algorithm guarantees a constant factor approximation for the SUKP if the number of facilities serving any customer is bounded, whereas the SUKP is hard to approximate in general, see Arulselvan (2014). The pseudo-code of this heuristic is presented in Algorithm 1.

Assume we are given a fractional solution $(\hat{x}; \hat{y}; \hat{z})$ of an LP relaxation of $P(D)$. Facilities are ordered in non-increasing order according to the ratio $\hat{z}_i = D^0(R_i)$, where we define $D^0(R_i) := \sum_{j \in R_i} \frac{d_j}{w_j}$ and $w_j := \sum_{i \in \mathcal{S}(i; j) \cap A_R} z_i$. The algorithm builds the complement H of the potentially violated cover I in an iterative way, starting with the empty complement set. As long as the demand covered by the complement is less than the minimum demand requirement, it greedily adds facilities to the complement set. If finally $\sum_{i \in I} \hat{z}_i < 1$ holds, then a violated SUKC inequality is found.

Algorithm 1 Heuristic Separation of Set Union Knapsack Cover Inequalities (13)

```

1:  $H := \emptyset$ ;
2: Let  $w_j$  be the number of facilities serving customer  $j, j \in R$ .
3: while  $F \neq \emptyset$ ; do
4:   pick a facility  $i \in F$  that maximizes  $\frac{z_i d_i}{\sum_{j \in R_i} w_j}$  and remove it from  $F$ 
5:   if  $D(R_H) + D(R_i) < D$  then
6:      $H := H \cup \{i\}, R := R \setminus R_i$ 
7: return cover  $I := F \cap H$ 

```

4.2. Lifted SUKC Inequalities

Given a minimal facility cover set $I \in COV_F$, the corresponding SUKC inequality (13) does not necessarily induce a facet of $P(D)$. It can be further strengthened via sequential or simultaneous lifting. A description of the general lifting technique for integer linear programming is given in Nemhauser and Wolsey (1988). In the following, we concentrate on lifting SUKC inequalities on customer variables.

Lifting on a Single Customer from $R_{\bar{i}}$

Theorem 2. For a facility cover $I \in COV_F$, let $R_{\bar{i}}^2 \subseteq R_{\bar{i}}$ be the subset of customers that satisfies the following property:

$$j \in R_{\bar{i}}^2 \subseteq R_{\bar{i}}, \quad D(R_i \cap R_{\bar{i}}) < D - [D(R_{\bar{i}}) - d_j] \text{ for all } i \in I: \quad (20)$$

Then, the following inequalities are valid for $P(D)$:

$$z(I) - \sum_{j \in I \cap COV_F} y_j \geq \sum_{j \in R_{\bar{i}}^2} z_j: \quad (21)$$

Proof. The set $R_{\bar{i}}^2$ determines the set of customers j such that if j is not served, we need to open at least two facilities from I in order to serve the demand D . Indeed, the value $D(R_{\bar{i}})$ is an overestimate of the demand covered by \bar{I} , assuming all customers in $R_{\bar{i}}$ will be served. If this is not the case, i.e., if customer $j \in R_{\bar{i}}$ remains unserved (i.e., $y_j = 0$), the total demand covered by \bar{I} is at most $D(R_{\bar{i}}) - d_j$, and hence, facilities from I need to cover at least $D - [D(R_{\bar{i}}) - d_j]$ units of demand. On the other hand, opening a single facility from i contributes in covering only customers from $R_i \cap R_{\bar{i}}$. Hence, if the inequality (20) holds for all $i \in I$, opening a single facility from I is not sufficient for covering the residual demand, and hence, at least two facilities from I will be needed. \square

We observe that for a given $I \in COV_F$ we can determine the subset $R_{\bar{i}}^2$ for which (21) holds in *polynomial time*, as it is sufficient to check whether the relation (20) holds for all $i \in I$.

Tighter Lifting on Customers from $R_{\bar{i}}$

Inequality (21) is a special, possibly relaxed case of the lifted SUKC inequality $z(I) + \sum_j (y_j - 1) = 1$. Notice that, in general, the lifting coefficient z_j in this inequality can be greater than one. In general, the optimal lifting coefficient z_j for any $j \in R_{\bar{i}}$ is obtained by solving the following problem:

$$z_j = 1 + \min_{z(I) + \sum_j (y_j - 1) = 1} z_j(x; y; z) \geq P(D); y_j = 0 \quad j \in R_{\bar{i}}: \quad (22)$$

Theorem 3. *The optimal lifting coefficient y_j in (22) is obtained by finding the smallest number of facilities from I to open in order to serve customers from $R_I \cap R_{\bar{I}}$ so that demand $D^\theta := D - D(R_{\bar{I}}) + d_j$ is covered. If the demand D^θ cannot be covered using facilities from I , the variable y_j can be fixed to one.*

Proof. Since the core network is connected, for any subset of open facilities from I there exists a feasible way to connect them. Hence, without loss of generality, minimizing $z(I)$ boils down to finding a feasible covering on the bipartite graph determined by the set I and the respective set of customers R_I . We first observe that $\min z(I)$ can be achieved by assuming all facilities from \bar{I} are open and, correspondingly, all customers from $R_{\bar{I}}$ except j are served (i.e., $y_j = 1, j \in R_{\bar{I} \cap \bar{I}^c}$), as there always exists a feasible point from $P(D)$ satisfying this condition. This explains why the uncovered demand D^θ can be set to $D - D(R_{\bar{I}}) + d_j$. Hence, facilities from I need to be open only to cover customers that cannot be served by \bar{I} , i.e., we are looking for a smallest subset of facilities from I to serve customers from $R_I \cap R_{\bar{I}}$ in order to cover the demand D^θ .

Finally, if not serving customer j (i.e., setting $y_j = 0$) results in a total demand that cannot be covered even if all facilities from $I \cup \bar{I}$ are open, clearly, customer j must be served, and the corresponding variable y_j must be fixed to one. \square

This lifting problem is known in the literature as the (NP-hard) Partial Set Covering Location Problem, see, e.g., Cordeau et al. (2019).

Theorem 4. *The optimization problem (22) can be formulated as the following ILP:*

$$z_j := 1 + \min_{i \in I} \sum_{i \in I} x_i \quad (23)$$

$$\sum_{k \in R_I \cap R_{\bar{I}}} x_k \geq 1 \quad \forall i \in R_I \cap R_{\bar{I}} \quad (24)$$

$$\sum_{k \in R_I \cap R_{\bar{I}}} d_k x_k \geq D^\theta \quad (25)$$

$$x_i \in \{0, 1\} \quad \forall i \in I \cup R_I \cap R_{\bar{I}} \quad (26)$$

Proof. The result follows from Theorem 3. In this model, the binary variables $x_i, i \in I$, determine whether facility i is open or not. The binary variables $x_k, k \in R_I \cap R_{\bar{I}}$, correspond to customers exclusively served by facilities I and they can be set to 1 only if at least one of their serving facilities in I is open (modelled by constraint (24)). Finally, the covering demand is modeled with the constraint (25), and in the objective function we count the number of open facilities needed to cover the demand D^θ . \square

Reapplying the above technique, we can sequentially lift the coefficients of subsequent customer variables within a SUKC inequality. Let us assume that all variables from a set $L \subseteq R_{\bar{I}}$ have already been lifted (with the associated lifting coefficients $\tilde{y}_k, k \in L$) and consider lifting on the variable y_j with $j \in R_{\bar{I}} \cap L$. Starting from the valid inequality $z(I) + \sum_{k \in L} \tilde{y}_k (y_k - 1) \geq 1$, we then seek for the largest coefficient y_j such that

the resulting inequality

$$z(I) + \sum_{k \in L} \tilde{c}_k(y_k - 1) + \sum_j (y_j - 1) \leq 1 \quad (27)$$

is valid. The associated coefficient \tilde{c}_j is obtained by solving the following problem

$$\tilde{c}_j = 1 + \min_{z(I) + \sum_{k \in L} \tilde{c}_k(y_k - 1) \leq 1} z(I) + \sum_{k \in L} \tilde{c}_k(y_k - 1) \quad j(x; y; z) \geq P(D); y_j = 0 \quad (28)$$

As this procedure requires solving an NP-hard problem for lifting each single coefficient of a given SUKC inequality, we also consider a simultaneous lifting explained below.

Simultaneous Lifting on Customers from $R_{\bar{I}}$

We now propose another lifting procedure in which we try to simultaneously lift coefficients of a subset of customers. Thereby, the computational effort is reduced, but the strength of the obtained inequalities may be sacrificed. Suppose that we are given $I \geq COV_F$ and that customers from $L \subseteq R_{\bar{I}}$ are not served. Then, the demand that has to be covered by opening facilities in I is at least $D = D(R_{\bar{I}}) + D(L)$. From this fact, we can derive the following result:

Theorem 5. *For a facility cover $I \geq COV_F$ and $L \subseteq R_{\bar{I}}$, let $D^\theta := D = D(R_{\bar{I}}) + D(L)$ and let $m_{\bar{I}}^L$ denote the minimum number of facilities in I that need to be opened in order to cover the demand of D^θ by serving the customers from $R_I \cap R_{\bar{I}}$. Then the following constraints are valid for $P(D)$:*

$$z(I) + (m_{\bar{I}}^L - 1)y(L) \leq m_{\bar{I}}^L \quad I \geq COV_F; L \subseteq R_{\bar{I}}; \quad (29)$$

These inequalities state that whenever we do not serve customers from L , even if all facilities from \bar{I} are opened, we need to open at least $m_{\bar{I}}^L$ facilities from I . Whenever $y(L) = 0$ and $m_{\bar{I}}^L > 1$, these inequalities are strictly stronger than the corresponding SUKC inequalities (13). If exactly one customer from L is served, these inequalities boil down to SUKC inequalities, and if two or more customers are served, then they are not binding.

Theorem 6. *Given a cover $I \geq COV_F$ and a set $L \subseteq R_{\bar{I}}$ the value of $m_{\bar{I}}^L$ is obtained by solving the following ILP:*

$$m_{\bar{I}}^L := \min_{i \in I} \sum_{i \in I} x_i : (x; y) \text{ satisfy (24)-(26)} \quad (30)$$

Proof. It is not difficult to see that the value $m_{\bar{I}}^L$ corresponds to the optimal solution of the Partial Set Covering Location Problem with the set of facilities being equal to I , the set of customers equal to $R_I \cap R_{\bar{I}}$, and the total demand equal to D^θ . Hence, the result follows from Theorem 4.

4.3. Separation of Lifted SUKC Inequalities

Given a minimal cover $I \subseteq F$ on the facilities, for which the plain SUKC inequality (13) is violated or nearly tight, we try to generate one of the stronger (lifted) SUKC inequalities (21) or (29) to add to the

formulation, instead of (13). Inequalities of type (21) can be easily separated in polynomial time by testing for each customer $j \in R_2$ if condition (20) holds, as described in Section 3.

We also apply the simultaneous lifting procedure. Recall that finding the most violated inequality of type (29) is NP-hard, even if we are given either both I and L . In our implementation we add two types of simultaneously lifted cuts. For a given fractional solution $(\hat{x}; \hat{y}; \hat{z})$ and for a detected violated cover $I \in COV_F$, we set $L := \{j \in R_I \cap R_I^2 \mid \hat{y}_j < g\}$, where g is a parameter we choose. We chose this value to be 0.2 as this empirically provided the best results in our preliminary computations. Then we solve the MIP described by (30) to determine the value of m_I^t . If finally $\hat{z}(I) + (m_I^t - 1)\hat{y}(L) < m_I^t$, we have detected the corresponding violated inequality of type (29). In addition, we separate these cuts by assuming that $m_I^t = 2$, in which case the corresponding set L is sought for, such that no single facility in I is sufficient to satisfy the demand requirement $D = D(R_I) + D(L)$. The set L is determined by considering customers served by the facilities in \bar{I} in the non-decreasing order of their demands.

Finally, for the sequential lifting of SUKC inequalities, we utilize the ILP formulation (23)–(26). This is a computationally expensive procedure which is applied only if the weak lifting given by inequality (21) was possible.

5. Branch-and-Cut Algorithm

In this section we describe our branch-and-cut algorithm and the subroutines used to solve the separation and lifting subproblems. Whenever we deal with inequalities that are valid for $P(D^t)$, we simply omit the superscript t for simplicity.

5.1. Initialization

The integer linear programs initially contain all variables and constraints (2) – (5), (8), (9) and (11). The remaining inequalities mentioned in the previous section are applied in a standard cutting plane approach and iteratively added if they are violated by the current fractional solution.

The requirement that the root node is connected to any open facility implies the *in-arc inequalities* $x(\text{in}(i)) = z_i$, for all $i \in F$. Also, potential Steiner nodes cannot be leaves of the Steiner tree, and therefore the following inequalities are also added in the initialization phase: $x(\text{in}(i)) = x(\text{out}(i))$, for all $i \in F \setminus R$. Furthermore, we add the inequalities $x(\text{in}(i)) = 1$ for all $i \in S$, $i \neq r$ to the description of $P(D)$. These inequalities ensure that the indegree of every node except the root node is at most 1. These inequalities may cut off feasible solutions, but, as there are no capacity constraints associated with the facilities and edges, there always exists an optimal solution of incremental ConFL that satisfies these inequalities. Adding these inequalities to the formulations substantially reduced the solution times in our experiments.

5.2. Branching

Branching on the assignment variables x_{ij}^t does not affect the search space as much as branching on the facility variables z_i^t . So we give the facility variables the highest branching priority. We also provide a high priority for branching on the number of open facilities. This was done by adding integer variables denoting the number of open facilities in each period together with the corresponding equalities defining these variables to the model. In our problem, we also have uncertainty in determining the set of customers to be served at each time period. So, we provide them with the next highest priority in branching.

5.3. Primal Heuristic

Inside of the branch-and-bound tree, we apply a primal heuristic that exploits the solution structure available through a current LP-solution. The idea is based on a heuristic computation of Steiner trees in each time period (see, e.g., Fischetti et al. (2017); Takahashi and Matsuyama (1980)) from the LP relaxation values. We assume that the heuristic takes an ordering of the facilities as input. This ordering process is described later. Let $(\bar{X} ; \bar{Z} ; \bar{X} ; Y ; Z)$ be the optimal LP-solution at the current branch-and-bound node. The following steps constitute this heuristic procedure (which is applied for each time period $t \geq T$, starting from the period $t = 1$):

1. Scan the facilities in the given order until the demand requirement for the given time period t is met by the customers adjacent to the facilities.
2. Let i be the current facility. If there are unserved customers that i could serve, mark facility i and all unserved customers $j \geq R_i$.
3. Fix all marked customers and facilities to be chosen in the solution for the current and all subsequent time periods. Note that the assignment variables x (denoting assignment of a specific customer to a facility) are not fixed in this step.
4. Run a shortest path heuristic for the Steiner tree problem to connect the open facilities with the root (for the given time period t):
 - (a) For each arc $(i;j) \geq A_S$, set its cost to $(1 - x_{ij}^t)$.
 - (b) Consider the open facilities as being *terminals* and process them in some order (see below).
 - (c) Let i be the current terminal. Find the shortest path from the root node to i .
 - (d) Fix the cost of the arcs along this path to zero and repeat this procedure until all terminals are connected to the root.
 - (e) Fix all edges of the Steiner tree to the solution.
5. For the partially fixed solution, solve the residual mixed integer program. Note that the solution has been only partially fixed before, as the assignment of the customers to the open facilities has not been specified.

6. Finally, in a post-processing step, non-serving open facilities are closed. Furthermore, if such a facility is a leaf of the Steiner tree of the solution, then the path connecting it to its nearest terminal or root node in the Steiner tree is also removed.

The first call to this heuristic picks a pure greedy ordering. Here, the facilities are increasingly ordered based on the cost to demand coverage ratio. The cost is an estimate based on the opening cost, service cost and shortest path cost to the root node. For the subsequent calls, one of the following orderings is picked for the heuristics with a certain probability.

The facilities are sorted in decreasing order according to the values of \tilde{z} . The ordering is provided in the increasing value of fractional LP values of the facilities, with the preference given to earlier time periods. In other words, we sort it according to fractional \tilde{z} values corresponding to the earliest time period and ties are broken by subsequent time periods. This ordering is picked with a 25% probability.

First, a randomized set of facilities is created. Membership of facility i in the set is based on the probability $\min(1; 2\tilde{z}_i^{-T})$, where T is the last time period. The facilities in this set are decreasingly ordered based on the demand coverage. The remaining facilities are just appended to this at the end. This order is picked with a 25% probability.

A randomized greedy order is generated. An initial subset of facilities is generated, where a facility's membership is randomly determined based on a probability of $\min(1; 2\tilde{z}_i^{-T})$. The facilities within the subset are ordered based on its cost to demand coverage ratio. This ordering is picked with a 48% chance.

A pure random ordering is selected with a 2% probability

5.4. Separation of Cut-Set Inequalities

A standard way to separate inequalities of type (10) is to compute for each facility node $i \in F$ a minimum r - i -cut in the digraph $(S; A_S)$ with arc capacities set to the current optimal LP-values \hat{x} , solving the corresponding maximum flow problem.

We observed that the cuts generated from r to i tend to generate cuts that are closer to the root node with many edges being repeated in the minimum cuts generated for different facilities. In order to avoid this, we reverse the capacities and treat the root as the sink and the facilities as the source. We also perform nested cuts, wherein we resolve maximum flow for the same facility by setting the capacity of the edges in the minimum cuts already identified to 1.

The separation routines are performed for each time period independently. Ordering of facilities is handled through a scoring system. Facilities are scored based on their previous successes (a score 0.01 is assigned) and failures (a score 1 is assigned) in generating a violated cut. They are then considered in the increasing

order of this score. We stop once 200 violated cuts are added. The strengthened cut-set cuts and customer cuts are combined in the same separation routine.

5.5. Separation of Strengthened Cut-Set Inequalities

Given a cover set $I \in COV_F$, to separate the associated cut-set-SUKC inequalities of type (17), we create an artificial sink node l and connect the nodes in I to l . We then compute a maximum r - l flow in the graph $G(S \cup I; A_S \cup I \rightarrow l)$ with capacities \hat{x} for the arcs in A_S and capacity 1 for the artificial arcs in $I \rightarrow l$. If the maximum flow value is less than 1, we add the corresponding violated cut-set-SUKC inequality.

Given a customer cover $J \in COV_R$, to separate the associated cut-set-cover inequalities of type (18), we create an artificial sink node l and connect the nodes in J to l . We then compute a maximum r - l flow in the graph $G(S \cup J; A_S \cup A_J \cup J \rightarrow l)$ where $A_J = \{(i; j) \in A_R \mid j \in J\}$. Capacities for the arcs in $A_S \cup A_J$ are set to \hat{x}_{ij} and capacity 1 is set to the artificial arcs in $J \rightarrow l$. If the maximum flow value is less than 1, we add the corresponding violated cut-set-cover inequality.

We handle these separations by maintaining pools of SUKC and customer-cover inequalities. The covers generated by the heuristic presented in Algorithm 1 may not be violated by the current fractional LP-solution and adding the cut in the current iteration would get it purged. We nevertheless insert the detected SUKC/customer-covers into a pool, from which we try to generate one of the inequalities of type (17) or (18). The separation of these cuts is performed only for the covers in the pool. Note that we may obtain a violated inequality even if the corresponding SUKC/customer-cover constraint is not violated.

6. Computational Experiments

6.1. Implementation

To assess the effectiveness of the presented formulation and additional valid inequalities, we implemented a branch-and-cut algorithm using CPLEX 12.2, a commercial integer programming solver with a branch-and-cut framework, via its Python API. The experiments were performed on an Intel Core2 Quad 2.66 GHz systems with 2GB RAM. Each run was carried out on a single processor. Each test run was limited to 3 600 CPU seconds and the optimality gap at this point of time is reported in the results.

The objective of the study is to show the efficiency of the inequalities introduced in Sections 3 and 4. As the inequalities are derived exclusively from the coverage requirements of a single period, our experiments provided in Section 6.3 focus on demonstrating the efficiency of these inequalities on a single period. The original application is however for the incremental setting, so in Section 6.4 we provide a broader picture of our results.

In our computational study, we compare the performance of the following four branch-and-cut settings:

B: basic formulation given by (2)-(5), (8)-(11). Separation of the exponentially large constraint set (10) is done according to the description provided in Section 5.4.

B^c : formulation B extended by two types of cover inequalities: SUKC-cuts (13) and customer-covers (16). All cover inequalities are added within the same callback function described for B and at the same frequency. The SUKC inequalities (13) are generated using the greedy heuristics explained in Algorithm 1 and the customer cover inequalities (16) are generated by solving a knapsack problem using a standard dynamic program, which runs in pseudopolynomial time. In addition, the two cardinality inequalities (14) and (15) are added to the initial formulation.

B^+ : formulation B^c extended by strengthened cut-set inequalities (17) and (18), which are added with respect to all the cover cuts generated in B^c .

B^{++} : formulation B^+ extended by all lifted inequalities provided in Section 4.2. Weak sequentially lifted cuts of type (21) are first added, followed by the simultaneously lifted cuts (29). Finally, if weak sequential lifting was successful, strong sequential lifting is performed and cuts of type (27) are added. The ordering of the customers for both weak and strong sequential lifting is in the decreasing order of their demands.

By default, the primal heuristic was turned on in all four settings. Despite their theoretical strength, customer-connectivity constraints (12) were turned off given their negative influence on the overall performance in practice, which is also known for the single-period ConFL, see Leitner et al. (2017a) and Gollowitzer and Ljubić (2011).

6.2. Benchmark Instances

The algorithm is tested on two types of benchmark instances: 1) real-world instances that we obtained from our industry partners, and 2) instances that were used in the previous literature for ConFL.

Real-world Instances. These instances are adapted from the inputs obtained from our industry partners in the German research project FTTx-Plan. The graphs correspond to various regions in Germany representing real and potential telecommunication networks. For each instance, the potential client and facility locations were provided by the industry partner and the network itself is abstracted from the publicly available information retrieved from a geographical information system. For the purposes of our problem, we first created a larger network in which the street segments form the edges and various junctions and intersections connected by streets form the nodes. Then, shortest paths between each customer and the potential facility locations are calculated and the closest 30 facilities were picked to be the potential serving facility of that customer with a service cost proportional to the shortest path distance. We have a total of 8 real-world instances ranging from a size of 1,000 to 12,000 nodes. The number of customers in these instances is roughly 50% to 60% of the number of nodes in the core network. A complete description of the individual instances is given in Table 1.

Name	jVj	jRj	jEj	jA_{rj}	jFj
ahorn	1676	604	13211	12080	105
marburg	2272	349	8054	7028	499
vehlefan	1316	238	5961	4761	149
mitte	6751	2440	53723	49015	532
mitte-small	4228	1490	32799	29917	320
lichterfelde	4111	1670	36086	33495	231
hhi	6751	2440	53723	49222	532
tegel	11545	4275	93718	86279	891

Table 1: Instance description of real world instances

ConFL Instances. In Ljubić (2007), a set of instances for single-period ConFL was generated by combining a set of benchmark instances for the Uncapacitated Facility Location (UFL) problem from the UFLlib (see Hofer (2013)) with instances of the Steiner tree problem (STP) from the OR-library (see Beasley (2013)). The ConFL input graphs are generated in the following way: First, jFj nodes of the STP instance are selected as potential facility locations, where jFj denotes the number of facilities in the corresponding UFL instance, 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 UFL files. The STP files provide edge-costs and additional Steiner nodes. We consider a set of 32 instances obtained by combining four UFL instances mp1, mp2 and mq1, mq2 (of size 200 200 and 300 300, respectively) with eight STP instances f_c, dgn , for $n \in \{5, 10, 15, 20\}$. These instances define the core networks with between 500 and 1000 nodes and with up to 25,000 edges. In the original ConFL instances provided in Ljubić (2007), the facility set F and customers R induce a complete bipartite graph. We desire a more sparse setting for our demand satisfaction and the cover set inequalities. Therefore, we only consider the closest 30 to 40 facilities for 95% of the customers and for the remaining customers we chose the closest 3 to 7 facilities. Such obtained instances contain up to 1000 nodes and 30,000 edges.

We extend both sets of instances to include demands and time periods. We generate demands uniformly at random between 20 and 40 for each customer and we consider a time horizon $T = 1$ or 3. For a single planning period, $T = 1$, we run our experiments for three different coverage rates, namely 50%, 75% and 90%. For the three stage planning, $T = 3$, we took the incremental coverage rates to be 50%, 75% and 90%.

6.3. Computational results for the single-period ConFL-CR

Our major methodological and theoretical contribution is related to the new families of valid inequalities introduced in Sections 3 and 4. In this section we therefore study the effect of these inequalities from the computational perspective. We focus on the quality of the LP-relaxation when these inequalities are added to the basic model, and on the overall computational performance, which is measured by the total computing time or the final gap obtained after a given time limit if an instance is not solved to optimality. Four different

instance	B				B^c			
	root gap	Final Gap	Cuts	nodes	root gap	Final Gap	Cuts	nodes
marburg	77.6	32.7	17409	313	81.0	77.2	9275/1075	1111
ahorn	71.3	67.5	12128	488	77.3	58.0	5937/220	240
vehlefanzen	67.1	59.4	12976	1651	68.8	60.4	8121/1076	1123
mitte	57.4	57.4	8982	0	63.5	63.5	3710/69	20
lichterfeld	63.9	63.9	10171	5	64.9	64.9	4894/90	47
mitte-small	64.9	64.9	12756	20	67.7	67.7	5455/100	52
hhi	57.2	57.2	7353	0	63.2	63.2	3703/69	20
tegel	78.9	78.6	1432	0	78.1	78.1	1939/36	0

instance	B^+				B^{++}			
	root gap	Final Gap	Cuts	nodes	root gap	Final Gap	Cuts	nodes
marburg	4.0	1.6	6528/411/3410	385	4.0	2.4	4265/184/1965/6370	145
ahorn	2.4	1.2	1844/411/851	400	2.4	1.2	1644/267/854/2849	250
vehlefanzen	4.0	0.2	5183/153/1654	98	4.4	0.3	3055/410/2280/13058	370
mitte	23.1	23.1	6199/32/442	0	25.9	25.9	1611/30/476/974	0
lichterfeld	12.4	12.4	7513/39/587	2	12.5	12.5	2561/47/1133/0	0
mitte-small	37.5	37.5	10403/53/1454	5	40.6	40.6	2744/50/1105/417	0
hhi	24.0	24.0	6140/32/557	0	26.1	26.1	1544/29/458/1059	0
tegel	63.8	63.8	3675/18/111	0	69.5	69.5	898/16/88/647	0

Table 2: Comparison of four branch-and-cut settings for the ConFL-CR with 50% coverage. The best obtained gaps are shown in bold.

settings, denoted by B , B^c , B^+ and B^{++} (cf. Section 6.1) are computationally compared. We first study the computational behaviour of these settings for the single-period ConFL-CR, and in the following section we report results obtained for iConFL.

ConFL-CR results for real-world instances

Tables 2, 3 and 4 present detailed results obtained on the real-world instances for coverage rates equal to 50%, 75% and 90%, respectively. For each of the settings we report: the LP-gap after processing the root node (incl. separation of cuts, “root gap”) calculated as $\frac{bestUB - rootLB}{bestUB}$; the gap obtained after the time limit of 3600 seconds was reached (“final gap”), calculated as $\frac{bestUB - finalLB}{bestUB}$; the number of detected violated cuts (“Cuts”), and the number of processed branch-and-bound nodes (“nodes”). For the setting B we report the cumulative number of inserted cuts of type (10). For B^c the second number of cuts that we report is the cumulative number of detected violated cover inequalities (13) and (16). For B^+ we in addition report on the number of detected violated cut-set-cover constraints (17) and (18), and finally for B^{++} we also provide the total number of detected violated lifted constraints.

From the obtained results we conclude that a significant amount of the gap for a majority of instances was closed by the violated cut-set-cover constraints (17) and (18). For the smaller instances of this data set, the final gap drops down from 30%-70% (for the B setting) to less than 2% (for the setting B^+). Surprisingly, the setting B^c performs worse than B , indicating the trade-off between the separation time needed for the pure SUKC cuts and customer covers and their overall improvement of the lower bounds. Only in combination with cut-set covers (setting B^+) we are able to truly benefit from SUKC cuts. The effect of cut-set-SUKC cuts (17) and cut-set covers (18) on the quality of lower bounds can be easily seen by comparing the root gaps between B^c and B^+ . Finally, our results also indicate that lifting procedures provide a computational

instance	B				B^c			
	root gap	Final Gap	Cuts	nodes	root gap	Final Gap	Cuts	nodes
marburg	70.4	10.4	18857	220	74.7	71.2	10573/332	326
ahorn	64.4	61.8	13159	396	63.6	60.1	6750/387	354
vehlefanzen	62.0	55.7	15523	1071	63.4	55.7	8415/1358	1352
mitte	47.8	47.8	8066	0	53.6	53.6	3519/65	15
lichterfeld	53.8	53.8	10392	5	55.9	55.4	3973/74	30
mitte-small	56.7	56.7	12209	13	59.9	59.9	5887/109	60
hhi	49.1	49.1	8062	0	53.9	53.9	3700/69	20
tegel	67.6	67.6	5285	0	72.1	72.0	2176/41	0
instance	B^+				B^{++}			
	root gap	Final Gap	Cuts	nodes	root gap	Final Gap	Cuts	nodes
marburg	7.9	5.2	11148/221/3030	195	7.9	6.9	6363/203/3228/1625	173
ahorn	5.7	5.6	2622/520/1057	496	5.5	5.5	1588/502/694/3216	465
vehlefanzen	2.3	0.0	2334/94/419	68	2.1	0.3	2374/96/420/101	61
mitte	22.5	22.5	6646/33/336	0	24.5	24.5	2009/37/762/379	0
lichterfeld	13.4	13.4	8863/56/811	10	13.6	13.6	3049/58/1568/246	12
mitte-small	35.7	35.7	12128/60/1766	10	37.9	37.9	3508/65/2021/1705	17
hhi	21.8	21.7	5788/29/465	0	25.0	25.0	1860/35/688/440	0
tegel	60.2	60.2	4300/21/246	0	66.7	66.7	1262/24/266/1322	0

Table 3: Comparison of four branch-and-cut settings for the ConFL-CR with 75% coverage. The best obtained gaps are shown in bold.

instance	B				B^c			
	root gap	Final Gap	Cuts	nodes	root gap	Final Gap	Cuts	nodes
marburg	67.6	59.4	21032	127	69.9	66.6	9776/806	767
ahorn	59.3	56.4	14730	240	62.7	56.9	5167/496	478
vehlefanzen	56.5	51.0	15631	1046	58.0	52.1	7936/1016	1008
mitte	47.3	47.3	6734	0	51.0	51.0	2692/50	0
lichterfeld	46.6	46.6	10346	5	49.8	49.8	4300/78	30
mitte-small	49.6	49.6	12246	11	52.7	52.7	5108/94	45
hhi	48.7	48.7	7356	0	51.9	51.9	2664/50	0
tegel	61.1	61.1	5304	0	65.6	65.6	2373/44	0
instance	B^+				B^{++}			
	root gap	Final Gap	Cuts	nodes	root gap	Final Gap	Cuts	nodes
marburg	7.7	5.6	14597/174/9666	142	10.2	7.2	6510/177/10285/685	150
ahorn	2.6	2.4	2607/508/685	286	2.6	2.5	2813/532/578/8570	500
vehlefanzen	4.0	0.4	4237/115/1225	94	4.0	0.0	2711/87/773/5	68
mitte	25.5	25.5	5704/28/588	0	28.2	28.2	2057/38/1030/7	0
lichterfeld	10.0	10.0	8378/46/543	8	9.8	9.8	3055/58/1238/164	15
mitte-small	28.2	28.2	10654/53/1334	7	28.8	28.8	3488/65/2010/565	15
hhi	27.2	27.2	5112/25/397	0	29.5	29.5	1882/35/879/7	0
tegel	55.0	55.0	4079/20/307	0	61.0	61.0	1376/26/484/989	0

Table 4: Comparison of four branch-and-cut settings for the ConFL-CR with 90% coverage. The best obtained gaps are shown in bold.

overhead that does not help improving the lower bounds within the given time limit. For more challenging instances of this data set, the algorithm gets stuck at the root node, spending too much time on the expensive lifting procedures.

Similar results can be observed for the single-period ConFL-CR when the coverage rate is set to 75% or 90% (cf. Tables 3 and 4).

ConFL-CR results for ConFL instances

Following the results reported on real-world instances, we decided to focus in the rest of our study on two settings: the basic setting B and the more advanced setting B^+ .

We first focus on the root gaps and the final gaps obtained on the set of ConFL instances. For the coverage rate of 50%, 75% and 90%, Figures 2, 3 and 4, respectively, depict the quality of lower bounds at the root node and the quality of global lower bounds obtained after the time limit is reached. The gaps are calculated with respect to the best upper bound. In these cumulative charts, a point with coordinates $(x; y)$ indicates that for $y\%$ of the instances, the obtained gap was $x\%$.

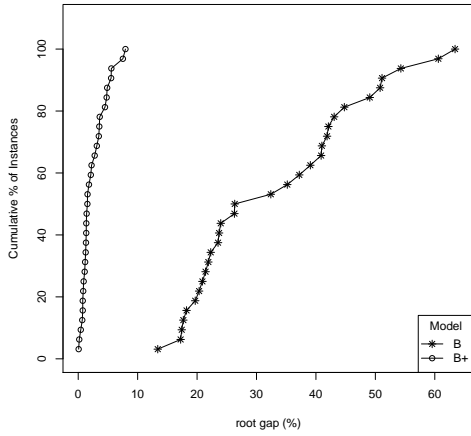
Independently on the coverage rate, we observe that the root bounds of the B^+ setting are significantly stronger than those obtained by the basic setting. Whereas for the lower coverage rate of 50%, basic setting manages to overcome this drawback by branching (ending with very small final gaps in 80% of the cases), for higher coverage rates, it is consistently overperformed by the setting B^+ . For example, for the coverage rate of 75%, there exist instances for which the LP-bounds of the basic formulation are as far as 50%, and with our new sets of inequalities this gap is never worse than 13%. Furthermore, even after reaching the time limit with the basic setting, the final gap remains $> 13\%$ for a quarter of the instances.

Figure 5 summarizes the obtained gaps over all ConFL instances and over all three coverage rates. We notice that instances with lower coverage rates are easier to solve than the ones with the higher coverage rates. Regarding the quality of lower bounds, we could observe that for all coverage rates, the root gap is significantly reduced in the presence of the new inequalities introduced in this paper and that these inequalities help in providing a more stable performance and in reducing the final gaps.

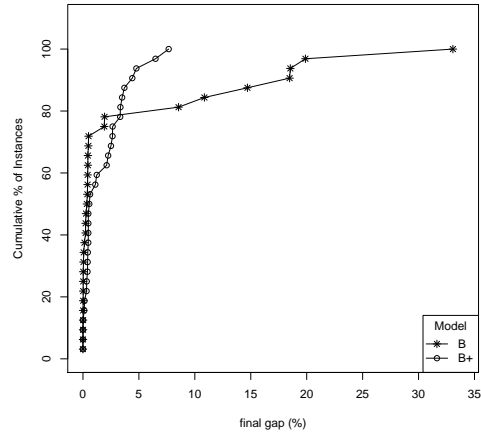
6.4. Computational results for the incremental ConFL

The focus of our experiments was predominantly on the single period network design problem, as the valid inequalities we introduced were targeted towards exploiting the polytope structure corresponding to the coverage restriction arising from a single period. To make the study complete, we performed our experiments on the ConFL benchmark instances and for the planning period with $T = 3$.

In Figure 6, we present computational results obtained for three periods, with the incremental coverage requirements set to 50%, 75% and 90%, respectively. As we can observe, this problem variant is much more challenging for exact solvers than its single-period counterpart. The root gap ranges between 12% and 32% for the setting B^+ , whereas for the basic setting, these values are between 17% and 47%. Over 40%

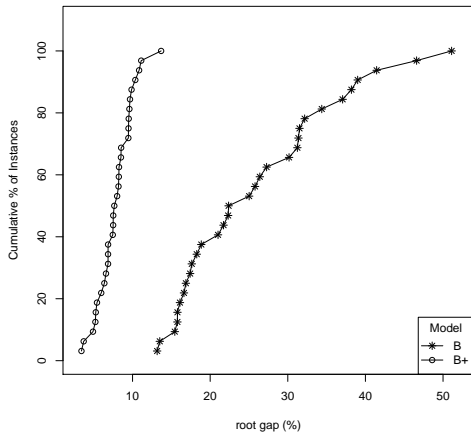


(a)

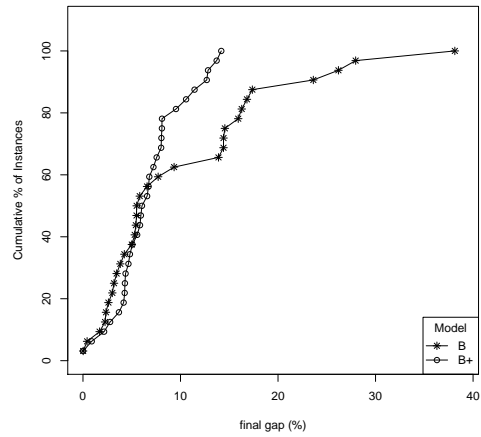


(b)

Figure 2: Coverage rate: 50%. ConFL instances.



(a)



(b)

Figure 3: Coverage rate: 75%. ConFL instances.

of instances have a root gap of more than 30%, and this value was brought down significantly using the proposed inequalities.

Comparing the charts depicting the root and the final gap, we notice that there are few differences between them, which is due to the fact that no branching was performed before reaching the time limit. This is not surprising, given the fact that the problem size grows with the number of planning periods, and hence, solving a single LP takes significantly more time.

We may conclude that iConFL still poses significant challenges for exact solution methods and that more has to be invested in future studies to be able to provide provably optimal solutions for instances of realistic

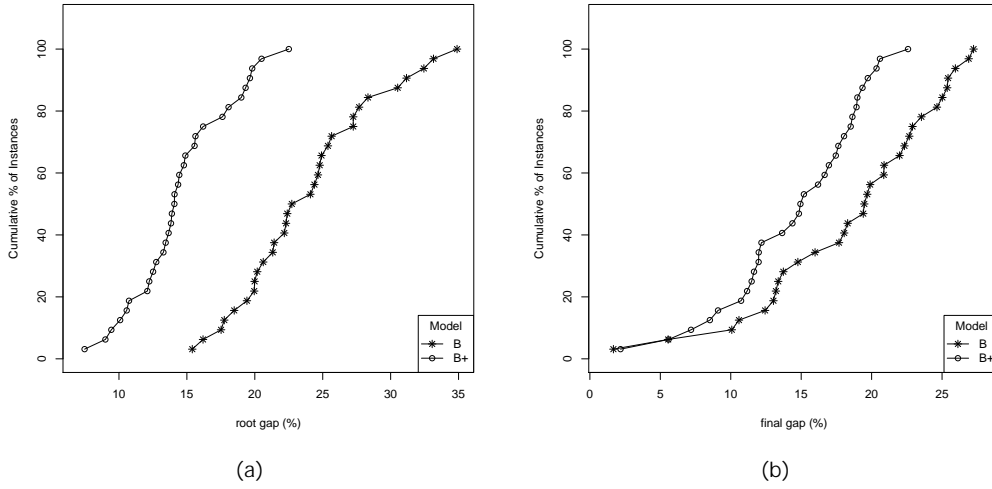


Figure 4: Coverage rate: 90%. ConFL instances.

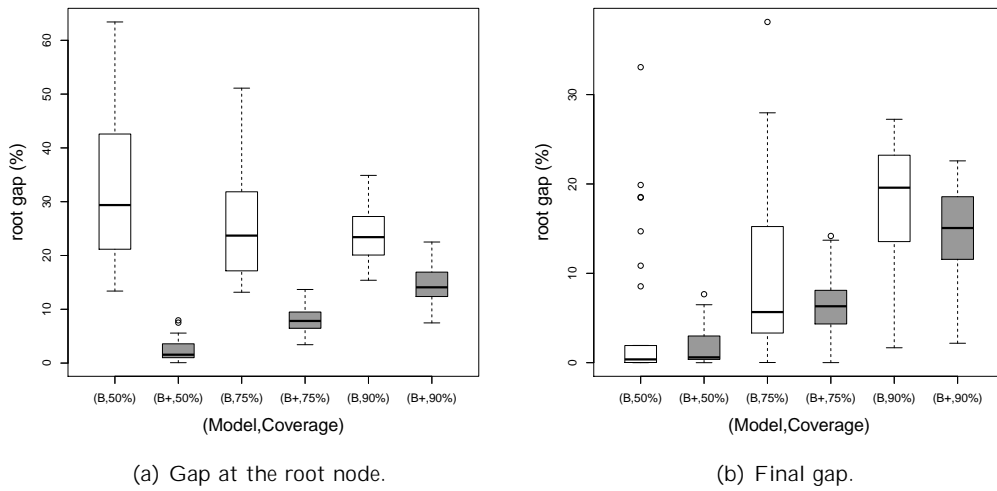


Figure 5: Comparison of root- and final gaps with increasing coverage rates. ConFL instances.

size. Nevertheless, our study provides new families of inequalities that can be used for improving the dual bounds and significantly reducing the final gaps, even for the instances with hundreds of nodes and tens of thousands of edges.

6.5. Effects of the primal heuristic to the overall performance

In order to demonstrate the need for problem-tailored primal heuristic, we ran tests on the real world instances for the best setting B^+ (with all the cover cuts but without lifting, as lifting takes considerable computational time) and compared our runs with and without the primal heuristic implemented for all time

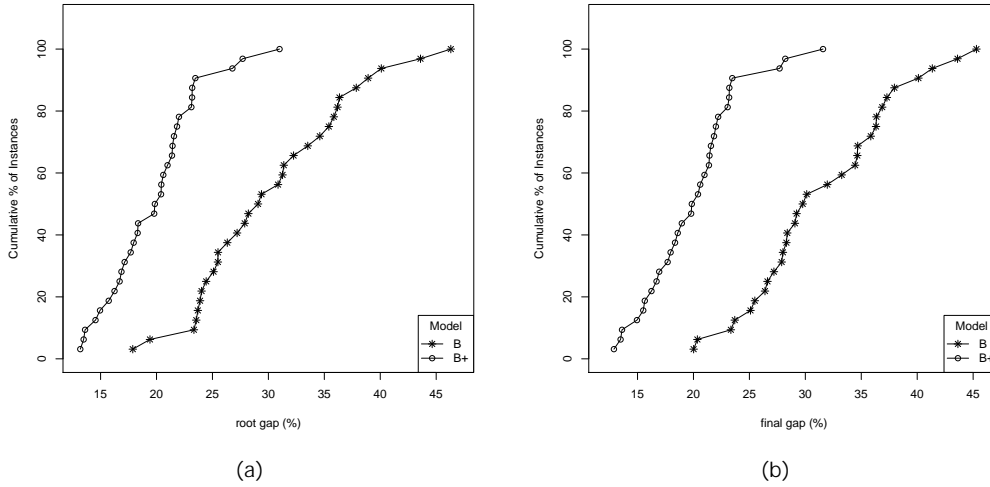


Figure 6: Three-period iConFL results for coverage rates: 50%, 75% and 90%. ConFL instances.

periods and we provide the results in Table 5. As we could see, the default CPLEX heuristics could only find a solution as one goes deep in the branch-and-bound tree and this happens only for one of the smaller instances, while our primal heuristic is providing reasonable solutions already at the root node. However, the computation time involved in the heuristic also leads to a smaller number of nodes explored. One could potentially employ a strategy that adjusts the frequency of calls to the heuristic callback according to the current gap to address this issue. However, we are presenting the results for the default strategy described in the implementation section in order to keep the tests uniform and simple.

Figure 7 provides a more detailed look on the results for the single period ConFL with and without our primal heuristic for the city of Ahorn. The bold points indicate that an incumbent was found. The implementation without the heuristic gets a solution as we delve deeper into the Branch-and-Bound tree. We also notice that with the primal heuristic we immediately get a good solution, but we do not make significant improvement as we go down the branch-and-bound tree. Ideally, we need to employ an adaptive strategy, where we call the primal heuristic at the early iterations when it pays off, but reduce the frequency of calls to the heuristic as we go down the branch-and-bound tree in order to give more time for exploring the nodes of the branch-and-bound tree. This also can be verified by the fact that the runs without the primal heuristic explore more nodes in the same time. We report the results from the setting where we make the heuristic call at the same rate throughout the run to keep the results uniform.

To conclude our study, we provide the results obtained for the Ahorn instance from the real-world data set. In Figure 8, we present the snapshots of the solutions obtained for three periods with coverage rates 50%;75% and 90%, when viewed on the Google Maps. This example shows that with already 3 open facilities, 50% of the demand can be covered, and that in the subsequent periods one, respectively two more facilities

instance	B^+ with Primal Heuristics			B^+ without Primal Heuristics		
	root gap	Final Gap	nodes	root gap	Final Gap	nodes
marburg	13.72	12.63	15	-	-	19
ahorn	3.37	3.29	73	-	-	127
vehlefan	12.53	9.91	118	-	14.52	740
mitte	62.08	62.08	0	-	-	0
lichterfeld	41.91	41.91	0	-	-	0
mitte-small	45.02	45.02	0	-	-	0
hhi	61.39	61.39	0	-	-	0
tegel	86.60	86.60	0	-	-	0

Table 5: Comparison of gaps with and without primal heuristics implemented for all time periods

have to be opened to reach the demand of 75%, respectively 90%.

This example demonstrates how our solution approach can be a useful tool for decision makers, not only for taking decisions concerning the infrastructure planning and scheduling, but also for running simulations and sensitivity analyses with respect to different coverage rates and the number of periods.

7. Conclusions

In this work we introduced a new combinatorial optimization problem that models the design of fiber-to-the-curb networks over time. The problem is a multi-period version of the connected facility location problem, which has been intensively studied in the literature in the last decade. We introduced several new families of valid inequalities derived from the incremental coverage constraints over time. We provide separation algorithms needed to detect these new coverage-related inequalities within a cutting plane framework. The problem is then solved by means of a branch-and-cut algorithm that makes use of the cut-set inequalities and the new coverage-related constraints. In the computational study we analyzed the influence of the minimum coverage rate D^t to the quality of lower bounds of the proposed models. We also showed that the new inequalities are useful for small and/or sparser instances, where the obtained duality gaps can be significantly reduced. For larger instances, there is sometimes a trade-off between the separation time of the coverage-related family of inequalities and the obtained improvement of the dual lower bounds. In general, our detailed computational results indicate that it is beneficial to consider SUKC inequalities, along with their combination with cut-set cuts, as they are detrimental for significantly reducing both, the LP-relaxation gaps and the overall computing times. On the other hand, our study shows that lifting SUKC inequalities is not beneficial from the computational perspective, due to computational overhead imposed by the respective separation algorithms.

The main questions regarding the iConFL and its single-period variant, ConFL-CR, that remain open are related to the complexity and approximability of these problems. Furthermore, efficient heuristics involving advanced local search procedures may prove to be useful for solving larger instances. Finally, we believe that decomposition-based methods (like Benders or Lagrangian decomposition as in Cordeau et al. (2019); Bley et al. (2013), respectively) could significantly help in further improving the computational performance of

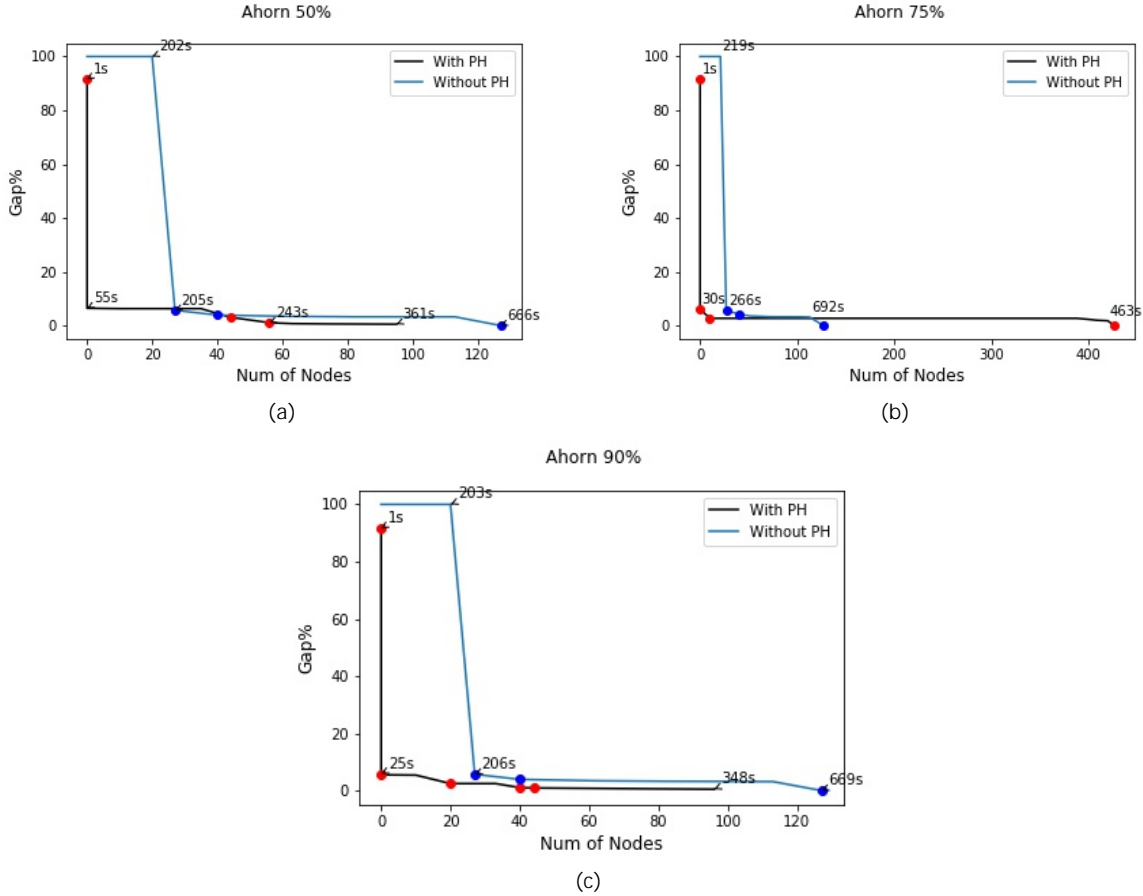


Figure 7: Three-period iConFL results for coverage rates for Ahorn: 50%, 75% and 90% with and without the Primal Heuristic (PH) implemented. The bold points indicate when an incumbent was found.

the proposed MIP-based approach.

Acknowledgments. This research has partially been done while A. Arulsevan, A. Bley and I. Ljubić stayed at COGA, TU Berlin. A. Arulsevan and A. Bley were supported by the DFG research center Matheon Mathematics for key technologies in Berlin and I. Ljubić was supported by the APART Fellowship of the Austrian Academy of Sciences. This support is greatly acknowledged.

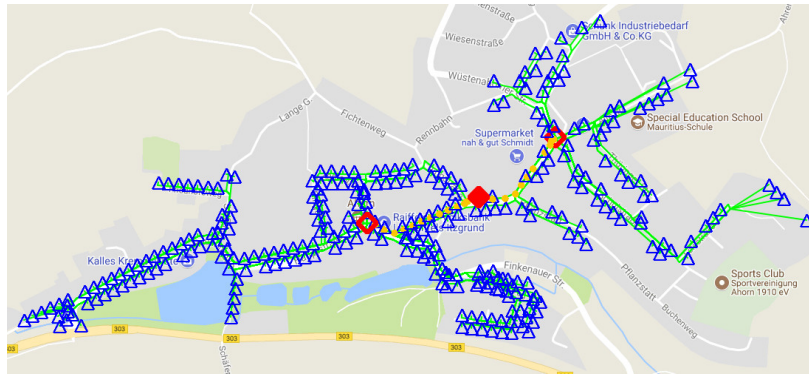
References

- M. Albareda-Sambola, E. Fernández, Y. Hinojosa, and J. Puerto. The multi-period incremental service facility location problem. *Computers & Operations Research*, 36:1356–1375, May 2009.
- A. Arulsevan. A note on set union knapsack problem. *Discrete Appl. Math.*, 169:214–218, 2014.
- A. Arulsevan, A. Bley, S. Gollowitzer, I. Ljubić, and O. Maurer. MIP modeling of incremental connected

- facility location. In *Proceedings of INOC 2011*, volume 6701 of *Lecture Notes in Computer Science*, pages 490–502. 2011.
- A. Arulselvan, O. Maurer, and M. Skutella. An incremental algorithm for the uncapacitated facility location problem. *Networks*, 65(4):306–311, 2015.
- I. Averbakh and J. Pereira. Network construction problems with due dates. *European Journal of Operational Research*, 244(3):715 – 729, 2015.
- E. Balas and E. Zemel. Facets of the knapsack polytope from minimal covers. *SIAM Journal on Applied Mathematics*, 34(1):119–148, 1978.
- 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.
- M. Baxter, T. Elgindy, A.T. Ernst, T. Kalinowski, and M.W.P. Savelsbergh. Incremental network design with shortest paths. *European Journal of Operational Research*, 238(3):675–684, 2014.
- J.E. Beasley. OR-library, 2013. URL <http://people.brunel.ac.uk/~mastjjb/jeb/orlib/steininfo.html>.
- D. Bienstock, O. Raskina, I. Saniee, and Q. Wang. Combined network design and multiperiod pricing: Modeling, solution techniques, and computation. *Operations Research*, 54(2):261–276, 2006.
- A. Bley, I. Ljubić, and O. Maurer. Lagrangian decompositions for the two-level FTTx network design problem. *EURO Journal on Computational Optimization*, 1(3):221–252, 2013.
- B. Cavdaroglu, E. Hammel, J.E. Mitchell, T.C. Sharkey, and W.A. Wallace. Integrating restoration and scheduling decisions for disrupted interdependent infrastructure systems. *Annals of Operations Research*, 203(1):279–294, 2013.
- Moses Charikar, Samir Khuller, David M. Mount, and Giri Narasimhan. Algorithms for facility location problems with outliers. In *Proceedings of the Twelfth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '01, pages 642–651, 2001.
- N. Christofides and B.P. Nicos. Optimal expansion of an existing network. *Mathematical Programming*, 6: 197–211, 1974.
- J.F. Cordeau, F. Furini, and I. Ljubić. Benders decomposition for very large scale partial set covering and maximal covering problems. *European Journal of Operational Research*, 2019. to appear.
- P.J. Doulliez and M.R. Rao. Optimal network capacity planning: A shortest-path scheme. *Operations Research*, 23(4):810–818, 1975.

- F. Eisenbrand, F. Grandoni, T. Rothvoß, and G. Schäfer. Connected facility location via random facility sampling and core detouring. *Journal of Computer and System Sciences*, 76:709–726, 2010.
- K. Engel, T. Kalinowski, and M.W.P. Savelsbergh. Incremental network design with minimum spanning trees. *J. Graph Algorithms Appl.*, 21(4):417–432, 2017.
- M. Fischetti, M. Leitner, I. Ljubić, M. Luipersbeck, M. Monaci, M. Resch, D. Salvagnin, and M. Sinnl. Thinning out steiner trees: a node-based model for uniform edge costs. *Math. Program. Comput.*, 9(2): 203–229, 2017.
- O. Goldschmidt, D. Nehme, and G. Yu. Note: On the set-union knapsack problem. *Naval Research Logistics (NRL)*, 41(6):833–842, 1994.
- S. Gollowitzer and I. Ljubić. MIP models for connected facility location: A theoretical and computational study. *Computers & Operations Research*, 38(2):435–449, 2011.
- S. Gollowitzer, B. Gendron, and I. Ljubić. A cutting plane algorithm for the capacitated connected facility location problem. *Computational Optimization and Applications*, 55(3):647–674, 2013.
- F. Havet and M. Wennink. The push tree problem. *Networks*, 44(4):281–291, 2004.
- M. Hoefer. UflLib, 2013. URL <http://www.mpi-inf.mpg.de/departments/d1/projects/benchmarks/UflLib/>.
- T. Kalinowski, D. Matsypura, and M.W.P. Savelsbergh. Incremental network design with maximum flows. *European Journal of Operational Research*, 242(1):51–62, 2015.
- K. Kaparis and A.N. Letchford. Separation algorithms for 0-1 knapsack polytopes. *Math. Program.*, 124(1-2):69–91, 2010.
- B. Lardeux and D. Nace. Multi-period network design of optical transmission networks. In *12th IEEE Symposium on Computers and Communications (ISCC 2007)*, pages 935–940, 2007.
- M. Leitner and G.R. Raidl. Variable neighborhood search for a prize collecting capacity constrained connected facility location problem. In *Proceedings of 2008 International Symposium on Applications and the Internet (SAINT)*, pages 233–236. IEEE Computer Society, 2008. ISBN 978-0-7695-3297-4.
- 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, 2011a.
- M. Leitner and G.R. Raidl. Variable neighborhood and greedy randomized adaptive search for capacitated connected facility location. In Roberto Moreno-Díaz, Franz Pichler, and Alexis Quesada-Arencibia, editors, *EUROCAST (1)*, volume 6927 of *Lecture Notes in Computer Science*, pages 295–302. Springer, 2011b.

- M. Leitner, I. Ljubić, M. Sinnl, and A. Werner. *fILP*g heuristics and a new exact method for bi-objective 0/1 ilps: Application to ftx-network design. *Computers & Operations Research*, 72:128 – 146, 2016.
- M. Leitner, I. Ljubić, and M. Sinnl J.J. Salazar-González and. An algorithmic framework for the exact solution of tree-star problems. *European Journal of Operational Research*, 261(1):54 – 66, 2017a.
- M. Leitner, I. Ljubić, J.J. Salazar-González, and M. Sinnl. The connected facility location polytope. *Discrete Applied Mathematics*, 2017b. To appear.
- I. Ljubić. *A hybrid VNS for connected facility location*, volume 4771 of *Lecture Notes in Computer Science*, pages 157–169. Springer, 2007.
- G.L. Nemhauser and L.A. Wolsey. *Integer and combinatorial optimization*. Wiley-Interscience, New York, NY, USA, 1988. ISBN 0-471-82819-X.
- S.H. Owen and M.S. Daskin. Strategic facility location: A review. *European Journal of Operational Research*, 111(3):423–447, 1998.
- U.H. Suhl and H. Hilbert. A branch-and-cut algorithm for solving generalized multiperiod Steiner problems in graphs. *Networks*, 31(4):273–282, 1998.
- H. Takahashi and A. Matsuyama. An approximate solution for the steiner problem in graphs. *Math. Japon.*, 24:573–577, 1980.
- A. Tomazic and I. Ljubić. A GRASP algorithm for the connected facility location problem. In *Proceedings of 2008 International Symposium on Applications and the Internet (SAINT)*, pages 257–260, 2008.
- S.V. Ukkusuri and G. Patil. Multi-period transportation network design under demand uncertainty. *Transportation Research Part B: Methodological*, 43(6):625 – 642, 2009.
- B. Wassermann. *Operations research in action: a project for designing telecommunication access networks*. PhD thesis, University of Vienna, 2012.
- N. Zadeh. On building minimum cost communication networks over time. *Networks*, 4(1):19–34, 1974.



(a)

(b)

(c)

Figure 8: Three-period iConFL results for coverage rates for Ahorn: 50%, 75% and 90%. ConFL instances. Maps courtesy of Google Maps.