New Formulations for Two Location Problems with Interconnected Facilities

Yerlan Kuzbakov[1], Ivana Ljubić[2]

ESSEC Business School of Paris, Cergy-Pontoise, France

Abstract

This paper studies two location problems with interconnected facilities. In the first problem, all customers need to be served by open facilities, and in the second (covering) variant a penalty is imposed for customers that cannot receive the service. Compared to the standard facility location setting, an additional constraint is imposed asking that all open facilities are interconnected, i.e., all open facilities need to be within a given radius of each other. These problems combine classical facility location aspects with network design, and we exploit this link to derive new mixed integer programming models. The strength of these models is investigated both theoretically and empirically. An extensive computational study is conducted on a set of benchmark instances from the literature, in which branch-and-cut, Benders decomposition and compact models are assessed in terms of the runtime and the resulting gaps.

Keywords: (O) Combinatorial Optimization, Discrete Location Problems, Network Design, Branch-and-Cut, Benders decomposition

1. Introduction

This paper investigates two discrete location problems involving interconnected facilities. In both problems, the goal is to minimize costs for opening facilities and allocating customers to these facilities while satisfying an additional connectivity constraint. This constraint requires that all open facilities should be located within a certain radius of each other and they should include a pre-specified root node. This interconnectivity property is particularly relevant for modeling communication in e.g., sensor networks [35] or radio networks [12]. The problems have been recently introduced by Cherkesly et al. [7], as the *Median Problem with Interconnected Facilities* (MPIF) and the *Covering Problem with Interconnected Facilities* (CPIF). In the MPIF, the goal is to cover all customers using the open facilities, while in the CPIF, if it is not profitable to cover all customers, a penalty is imposed for each customer who does not receive service.

[1]yerlan.kuzbakov@essec.edu
[2]ivana.ljubic@essec.edu

MPIF/CPIF are closely related to problems studied in the network optimization literature under the common name of *Generalized Steiner Tree-Star* problems (GSTS), see, e.g., [24] and further references therein. The major difference between MPIF/CPIF and GSTS lies in the cost function. In MPIF/CPIF there is a cost associated to opening a facility, and a cost for assigning a customer to a facility or penalty for not satisfying the demand. Contrary to the GSTS, we do not incur costs for the edges connecting facilities. This assumption can be explained by the nature of the connection between open facilities, which is rather informational and not physical. We can further require facilities to serve only customers located within a radius R. Figure 1a illustrates an input instance: triangles correspond to potential facility locations, customers are shown as dots, the root node is shown as a square, and circles with smaller radius ($r > 0$) centered around a potential facility node represent the range of communication with the neighboring facilities. A possible solution is shown in Figure 1b, where open facilities (black triangles) together with the given root node build a connected network. Two open facilities can be connected by an edge if the distance between them does not exceed r. Since there is no cost incurred for such edges, it is sufficient to build a tree that connects the root with open facilities, and one such tree is shown in Figure 1b.

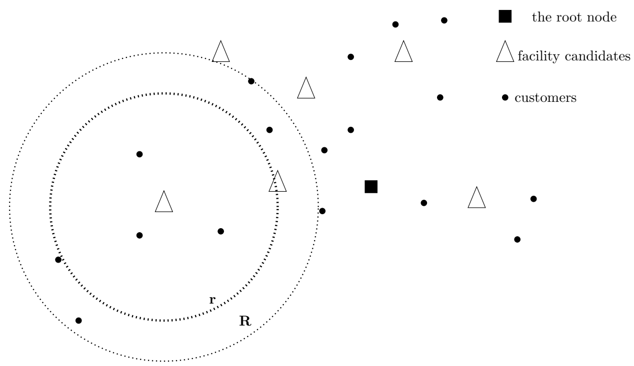
Our contribution

The article contributes to the literature of discrete location problems as follows:

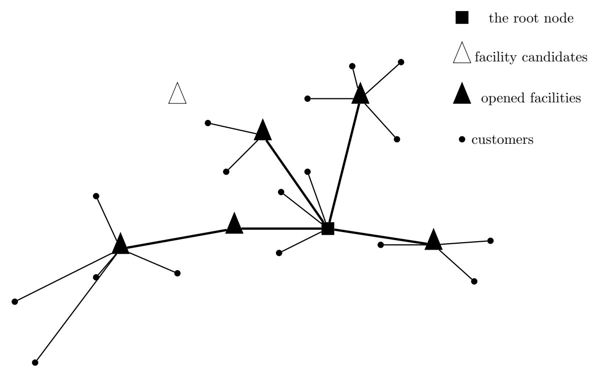
- We show that MPIF/CPIF can be decomposed into two modeling components that can be handled separately: one is related to imposing the interconnectivity between open facilities, and the other is related to assignment, respectively, covering of customers through open facilities. We provide both compact and non-compact ways to deal with both modeling aspects. For each of the two problems, six different formulations are proposed. They facilitate flow-based or node-separator-based constraints to ensure connectivity between the facilities. At the assignment level, we show that standard compact formulations can be replaced by respective Benders cuts.

- From the theoretical perspective, we explore the theoretical quality of LP-relaxation bounds between two flow-based formulations to model interconnectivity, one based on single-commodity flows (proposed earlier in [7]) and the other based on multi-commodity flows. We provide worst-case examples, proving that the quality of the LP-bounds of the single-commodity flow model can be arbitrarily bad (the gap between the LP-relaxation and the optimal value grows with the size of the input graph). In addition, we derive a surprising result showing that the multi-commodity-flow-based models can be as weak as their single-commodity-based counterparts.

- From the empirical perspective, we implement several tailored branch-and-(Benders)-cut algorithms and test them on benchmark instances from the literature with up to 100,000 customers. We conduct an extensive computational study in which we compare our branch-and-(Benders)-cut approaches against a state-of-the-art MIP solver Cplex. The results show that our tailored approaches always outperform Cplex when the latter is used as an off-the-shelf-solver (no matter if compact formulations or their Benders counterparts are solved – using the advanced *automatic Benders decomposition* of Cplex). The obtained computational results also support our theoretical findings concerning



(a) *Input*



(b) *Solution*

Figure 1: Sample network of interconnected facilities with customers. Open facilities have to be connected to the root node via network where the nodes are open facilities and the length of edges does not exceed the radius r. Customers laying within radius R from an open facility are considered as covered or served.

the quality of lower bounds of the flow-based formulations. Finally, the results indicate that the most critical modeling aspect is the interconnectivity of facilities for which the branch-and-cut methods based on node-separator cuts exhibit the best performance.

The paper is organized as follows. In Section 1.1, the formal definition of the two problems is given. In Section 1.2, related facility location problems are presented in a detailed literature review. Then, in Sections 2 and 3 we introduce different ways to model the MPIF and the CPIF, respectively. We compare the tightness of two flow-based formulations to model the network of open facilities in Section 4. In Section 5, we elaborate on the algorithms employed in our computational experiments. In Section 6 we assess the performance of our exact methods and we derive final conclusions in Section 7.

1.1. Problem definitions and notation

The MPIF and CPIF are defined on a set of potential facility locations I (which also includes a root node 0), and a set of customers J. In addition, we are given an undirected graph $G_I = (I, E_I)$ whose nodes are potential facility locations, and such that any two facilities

$i,\ k \in I,\ k \neq i$ are connected by an edge $\{i,k\} \in E_I$ if and only if their distance is not greater than a given radius $r > 0$. We also consider a directed counterpart of G_I, denoted by $D_I = (I, A_I)$ in which to each edge $\{i,k\} \in E_I$ we associate two reverse arcs (i,k), $(k,i) \in A_I$.

Opening a facility at node $i \in I$ incurs fixed opening cost $g_i \geq 0$. Moreover, let $c_{ji} \geq 0$ be the transportation cost per unit of demand between facility $i \in I$ and customer $j \in J$. The demand of customer $j \in J$ is denoted as $d_j \geq 0$. There are no capacities on facilities, and hence, demand of each customer can be fully allocated to the closest open facility. In the MPIF variant, allocating a customer $j \in J$ to facility $i \in I$ incurs allocation cost equal to $d_j c_{ji}$. In the CPIF variant, not serving a customer $j \in J$ incurs a penalty which is proportional to its demand (d_j/α, for a given parameter $\alpha > 0$).

For a given $j \in J$, let $I(j)$ be the subset of facility nodes $i \in I$ such that $c_{ji} \leq R$ (we will also say, j is within the covering radius R of i). For a given $i \in I$, let $J(i)$ be the subset of customers $j \in J$ such that $c_{ji} \leq R$. Similarly, for a given $I_F \subseteq I$, let $J(I_F) \subseteq J$ denote all customers that are within the radius R from at least one of the facilities from I_F, i.e., $J(I_F) = \cup_{i \in I_F} J(i)$. Let J_s be the subset of customers that can be served by a single facility only, i.e., $J_s = \{j \in J : |I(j)| = 1\}$. Finally, for a given subset of open facilities $\tilde{I} \subseteq I$ we define $\tilde{I}_j$ as the number of open facilities from $\tilde{I}$ that can serve customer $j \in J$, i.e., $\tilde{I}_j = |\tilde{I} \cap I(j)|$.

Definition 1 (Interconnected facilities)**.** *A subset of facilities $I_F \subseteq I$ including the root node is called interconnected if there is a path in the graph G_I from the root node to each of the facilities from I_F.*

Definition 2 (The Median Problem with Interconnected Facilities, MPIF)**.** *The MPIF seeks for a subset of interconnected facilities I_F such that each customer $j \in J$ is assigned to a facility $i(j) \in I_F \cap I(j)$ and such that the cost required to open the facilities from I_F and to allocate the customers to them, defined as*

$$\sum_{i \in I_F} g_i + \sum_{j \in J} d_j c_{ji(j)} \tag{1}$$

is minimized.

Definition 3 (The Covering Problem with Interconnected Facilities, CPIF)**.** *The CPIF seeks for a subset of interconnected facilities I_F such that the incurred facility opening cost and the penalty incurred by customers that lie further than R from any open facility from I_F, defined as*

$$\alpha \sum_{i \in I_F} g_i + \sum_{j \in J \setminus J(I_F)} d_j \tag{2}$$

is minimized.

The problems MPIF and CPIF are NP-hard even if all facilities are pairwise interconnected, i.e., even if r is sufficiently large so that G_I becomes a complete graph. Indeed, in this case, the MPIF boils down to the classical facility location problem, and the CPIF becomes the maximum covering location problem, which are known to be NP-hard [22, 23].

1.2. Related literature

Location problems with interconnected facilities are introduced by Cherkesly et al. [7]. The authors propose a compact formulation for MPIF/CPIF and design a metaheuristic algorithm based on iterative local search to solve these problems. Their compact formulation is based on a single-commodity flow formulation (SCF) whose details are provided in Section 2.1.1. In our paper, along with the SCF model, we explore alternative ways to model network connectivity and assignment constraints. We also compare the quality of LP-relaxations of the SCF formulation with an alternative flow model proposed in this paper.

In the following, we review several facility location and network design problems closely related to the MPIF and the CPIF:

- The classical Facility Location Problem (FLP), the p-median problem, and the p-center problem: The literature on the FLP is vast, and the most recent advancements in exact methods for the FLP can be found in [17], where two Benders reformulations are given to project out assignment variables. For a more comprehensive literature overview on the FLP, see also [15]. For the most recent studies on p-median, see, e.g., [1, 11, 14]. State-of-the-art methods for the p-center problem and the capacitated facility location problem are based on branch-and-(Benders-)-cut strategies, see [18] and [36], respectively. A survey of metaheuristic approaches is given in [30]. Exact methods can be found in [14, 19, 13].
- Connected Facility Location (ConFL): This is a minimization problem requiring that open facilities are connected via a Steiner tree. The sum of facility opening *and connectivity cost* together with the customer assignment cost is minimized. Unlike the MPIF, in the ConFL, the edge cost of the Steiner tree connecting open facilities is part of the objective function. In addition, for ConFL, no costs are incurred for facilities that do not serve customers, even if they are part of the interconnected network, whereas in the MPIF opening costs need to be paid for all facilities building the interconnected network. The ConFL problems are studied in, e.g., [4, 20, 24, 26].
- The Prize-Collecting Steiner Tree Problem (PCSTP). In the PCSTP, we are looking for a connected subgraph that maximizes the sum of node weights minus the edge costs needed to connect the nodes trough a tree. To model the MPIF as the PCSTP, let us assume for simplicity that $I \cap J = \emptyset$. We assign a big-M node-weight to every customer, and a node-weight $-g_i$ to every facility $i \in I$. In addition, edge-costs are set to zero for every edge connecting two facilities, and they are set to $d_j c_{ji}$ to every edge connecting $i \in I$ and $j \in J$, assuming their distance is within the radius R. The PCSTP is studied in e.g., [16, 25, 28, 32]. In [24] a general algorithmic framework is proposed for the ConFL and the PCSTP. In particular, in [16] the PCSTP is modeled using node variables only (assuming uniform edge-costs). Node-separator constraints used in [16] (among others) to ensure connectivity between nodes, will also be used to model interconnectivity between facilities when solving the MPIF/CPIF.
- The Maximum (Node-) Weight Connected Subgraph Problem (MWCS) is a maximization problem asking to find a connected subgraph with the maximum weight of nodes. The objective of the MWCS resembles that of the CPIF in that it has to cover as much

demand at the nodes as possible and no edge costs are incurred. Recent studies on the MWCS can be found in [2, 5, 25, 31, 33].

- The Set Covering Location Problem (SCLP). In the SCLP the goal is to find a subset of facilities of minimum opening cost so that every customer is covered at least once. A customer is said to be covered by a facility if it lies within the given coverage radius of this facility. In [10] two types of the SCLP related to our paper are considered: the maximal covering location problem (MCLP) and the partial set covering location problem (PSCLP). In the MCLP, we are given a budget on the cost of open facilities and have to cover as many customers as possible. In the PSCLP, a minimum percentage of demand needs to be covered while minimizing the cost of open facilities. In [10], for both problems polynomial time procedurs are proposed to derive Benders cuts and project out customer variables. One of our models for the CPIF will also employ this type of Benders cuts. The major difference between the CPIF and the PSCLP, besides the interconnectivity constraints, is that we do not impose a minimum demand to be covered, but penalize the unserved customers in the objective function instead. As for modeling the MPIF, starting from the PSCLP, we need to add interconnectivity constraints and require that 100% of the demand is covered.

Finally, the tree of hubs location problem is another (more distantly) related location problem on trees, see, e.g., [9]. A more comprehensive list of references can be found in a recent survey on Steiner trees and related network design problems [27].

2. Mathematical models for the MPIF

We start by presenting a generic model for the MPIF in the natural space of variables. In the following, binary node variables y_i are set to 1 if and only if facility at node $i \in I$ is open. We also introduce binary assignment variables w_{ji} which are defined for each $j \in J$ and $i \in I \setminus \{j\}$, to indicate whether customer $j \in J$ is assigned to facility $i \in I$. For a node $i \in I \cap J$, it will be assumed that whenever $y_i = 1$, the node is assigned to itself at zero cost (and hence, it is not needed to explicitly deal with w_{ii} variables). A generic MIP model for the MPIF can now be stated as follows:

$$\text{minimize} \sum_{i \in I} g_i y_i + \sum_{j \in J} \sum_{i \in I(j) \setminus \{j\}} d_j c_{ji} w_{ji} \tag{3}$$

$$\text{s.t. } y \in \mathcal{Y}, y_0 = 1 \tag{4}$$

$$w \in \mathcal{W}(y) \tag{5}$$

where $\mathcal{Y} \subseteq \{0,1\}^{|I|}$ represents a set of incidence vectors of possible subsets of interconnected facilities, and with $y_0 = 1$ we ensure that the root facility is always open. Similarly, $\mathcal{W}(y)$ refers to a set of binary incidence vectors corresponding to feasible assignments of customers to open facilities determined by the vector y. The objective function (3) is the cost-minimization variant of the MPIF as proposed in [7]. Moreover, we will also consider the cardinality-constrained variant of the problem (under the same name, MPIF, as in [7]), in which the number of open facilities must be equal to p, i.e., in which we add to the model the following constraint:

$$\sum_{i \in I} y_i = p. \tag{6}$$

In the latter case, the first term of the objective function (3) becomes a constant and can be ignored, assuming that $g_i = g$, for all $i \in I$.

The following constraints are determining the set $\mathcal{W}(y)$:

$$w_{ji} \leq y_i \qquad i \in I, j \in J(i) \tag{7a}$$

$$y_j + \sum_{i \in I(j) \setminus \{j\}} w_{ji} = 1 \qquad j \in J \cap I \tag{7b}$$

$$\sum_{i \in I(j)} w_{ji} = 1 \qquad j \in J \setminus I \tag{7c}$$

$$w_{ji} \in \{0, 1\} \qquad j \in J, i \in I(j), i \neq j \tag{7d}$$

Constraints (7a)-(7c) ensure that each customer is assigned to an open facility located within the radius R from it. The objective function (3) minimizes fixed costs of open facilities as well as costs to connect customers to facilities. It is well-known (due to the total unimodularity property of the constraint matrix) that we can replace (7d) by

$$w_{ji} \geq 0 \quad j \in J, i \in I(j), i \neq j \tag{8}$$

In the following, we will present several alternative ways to model interconnectivity constraints, and in Section 2.2 we will also use a Benders reformulation to model the assignment constraints.

2.1. Modelling the set $\mathcal{Y}$ (the network design component)

The interconnectivity constraint requires that the set of open facilities builds a connected subgraph with the root node as one of its nodes. To achieve this we can use flow formulations. There are two main types of flow formulations typically used to impose connectivity: the first one relies on a single-commodity flow (SCF) and the second one on multiple-commodity flows (MCF). For some network design problems, the MCF-based formulations are known to provide better LP-relaxation bounds than their SCF-based counterparts (see, e.g., [27]). We will investigate these relationships for the MPIF and CPIF in Section 4.

2.1.1. Single-commodity flow formulation SCF

In [7], the single-commodity flow-based formulation (9) was used to describe the network for the MPIF and the CPIF. Similar ideas have been used earlier in, e.g., [2, 3, 6]. The formulation uses auxiliary flow variables f_{ik}, for all $(i, k) \in A_I$. We assume that the flow represented by variables f is coming from the root node to each open facility via other open facilities. The amount of flow terminating at node $i \in I$ is required to be y_i.

$$\sum_{(0,k)\in A_I} f_{0k} - \sum_{(k,0)\in A_I} f_{k0} = \sum_{i\in I\setminus\{0\}} y_i \tag{9a}$$

$$\sum_{(i,k)\in A_I} f_{ik} - \sum_{(k,i)\in A_I} f_{ki} = y_k \qquad k \in I \setminus \{0\} \tag{9b}$$

$$f_{\ell k} + f_{k\ell} \leq (|I| - 1) y_\ell \qquad \ell \in I, \{\ell, k\} \in E_I \tag{9c}$$

$$f_{\ell k} \geq 0 \qquad (\ell, k) \in A_I \tag{9d}$$

Constraints (9a)-(9c) ensure that there is a path in G_I along which the flow can pass from the root node to each of the open facilities, thus guaranteeing connectivity of the open facilities. Constraint (9a) ensures that a sufficient amount of flow is sent out from the root node to reach all open facilities. Constraints (9b) guarantee that the amount of flow terminating at a facility is equal to y_i. Constraints (9c) limit the flow along the arcs between two open facilities. Flow can exist only on arcs of A_I as stated by (9d).

Contrary to standard flow-based formulation from the literature, where the flow variables $f_{\ell k}$, $(\ell, k) \in A_I$ are linked to arcs of the underlying network D_I (see, e.g., [29]), this model links flows with node variables only. That way, a sparser formulation is obtained which exploits the fact that variables associated to $(i, j) \in A_I$ do not appear in the objective function. Potential disadvantage of this model is related to the big-M constraints (9c) which may render its LP-relaxation very weak, as we will discuss it in Section 4.

2.1.2. Multi-commodity flow formulation MCF

We now propose an alternative way to use flow variables to model connectivity. In this formulation a separate flow h^i destined to facility $i \in I \setminus \{0\}$ is coming from the root node and passes through other open facilities. This time we don't involve big-M constants as each commodity flow can be modeled separately. As a result, we have constraints (10b) for each commodity and each edge of E_I, where we avoid using arc variables and provide a link between the flows and facility variables directly.

$$\sum_{(\ell,k)\in A_I} h^i_{\ell k} - \sum_{(k,\ell)\in A_I} h^i_{k\ell} = \begin{cases} y_i & \text{if } k = i \\ -y_i & \text{if } k = 0 \\ 0 & \text{otherwise} \end{cases} \qquad i, k \in I \tag{10a}$$

$$h^i_{k\ell} + h^i_{\ell k} \leq y_\ell \qquad i \in I, \ell \in I, \{\ell, k\} \in E_I \tag{10b}$$

$$h^i_{\ell k} \geq 0 \qquad i \in I, (\ell, k) \in A_I \tag{10c}$$

Constraints (10a) ensure that one unit of flow for commodity $i \in I$ (provided that $y_i = 1$) departs from the root node 0 and terminates at node i. As with the SCF, the flow is limited to arcs in the graph G_I, due to constraints (10b) and (10c).

This model should give us possibly tighter LP-relaxation bounds at the cost of increasing the number of variables and constraints. In Section 4 we compare the quality of the LP-relaxation of models with connectivity constraints based on the SCF and the MCF.

2.1.3. Connectivity cuts

Yet another way to impose connectivity between facilities is obtained with a non-compact formulation in the natural space of y variables. The connectivity is ensured by an exponential number of constraints using the notion of node separators.

Definition 4 (node separators)**.** *For any pair of distinct and non-adjacent facilities $k, \ell \in I$, the set $N \subset I \setminus \{k, \ell\}$ is called a (k, ℓ)-separator iff after the elimination of all nodes from N from the graph G_I there is no (k, ℓ)-path in the resulting graph.*

Let $\mathcal{N}(0, i)$ denote the family of all $(0, i)$-separators for a given facility $i \in I \setminus \{0\}$ (obviously, if $\{0, i\} \in E_I$ then $\mathcal{N}(0, i) = \emptyset$). Then in our generic model (3)-(5) we can replace the SCF or MCF constraints (9), respectively (10), with the following exponential family of cuts:

$$\sum_{j \in N} y_j \geq y_i \quad i \in I \setminus \{0\} : \{0, i\} \notin E_I, \quad N \in \mathcal{N}(0, i). \tag{11}$$

These constraints ensure that for any open facility $i \in I$, which is not directly adjacent to the root node 0,there exists a $0 - i$ path in which all facilities along this path are open. We observe that for facilities i such that $\{0, i\} \in E_I$, no connectivity constraints need to be imposed (as, by default, the root facility is always open).

The notion of node separators has been successfully used to develop branch-and-cut methods for the MWCS in e.g., [2, 3, 16]. These constraints are separated in a dynamic fashion. Given a vector y with integer coordinates, one can check in polynomial time if this vector satisfies all constraints of type (11). For example, if there is a connected component $C \subset I$ of open facilities that does not contain the root node, then the neighboring nodes of such component in G_I provide $|C|$ node separators $N \in \mathcal{N}(0, i)$, one for each $i \in C$. Hence, for each $i \in C$, an associated cut (11) can be generated, in order to enforce that at least one of these neighboring nodes has to be open in a feasible solution, in case facility at node i is open.

It is not difficult to see that tighter cuts can be obtained if *minimal* (with respect to node-inclusion) node separators are used instead. We have implemented an efficient separation procedure for finding minimal node separator cuts, based on the following proposition that has been shown for Steiner trees in [16]:

Proposition 1 (based on [16])**.** *Let $N \in \mathcal{N}(0, i)$ be a $(0, i)$-separator for some $i \in I \setminus \{0\}$, and let C_0 and C_i be connected components of G_I obtained after removing nodes from N, such that $0 \in C_0, i \in C_i$. Then N is a minimal $(0, i)$ node separator if and only if every node in N is adjacent to at least one node in C_0 and to at least one node in C_i.*

More details on how these constraints are integrated into our branch-and-(Benders)-cut procedure are provided in Section 5.

2.2. Benders decomposition of the assignment constraints

At the beginning of this section we proposed to explicitly model the assignment of customers to open facilities using w variables. However, we can also project out w variables and just trace the cost of assignment for each customer $j \in J$ instead. Thus, to calculate the cost of assignment of customer $j \in J$ we introduce new continuous variables $v_j \geq 0$ representing the cost to serve customer $j \in J$ given a vector y of open facilities.

The new generic model for the MPIF can be restated in the space of (y, v) variables:

$$\text{minimize} \sum_{i \in I} g_i y_i + \sum_{j \in J} d_j v_j \tag{12}$$

$$\text{s.t. } v_j \geq c_{jk} - \sum_{i \in I(j): c_{ji} < c_{jk}} (c_{jk} - c_{ji}) y_i \qquad j \in J, k \in I(j) \tag{13a}$$

$$\sum_{i \in I(j)} y_i \geq 1 \qquad j \in J \tag{13b}$$

$$v_j \geq 0 \qquad j \in J \tag{13c}$$

$$y \in \mathcal{Y}, y_0 = 1$$

Constraints (13a) can be derived using Benders decomposition and LP-duality theory, see, e.g., [8, 17]. We briefly sketch this idea in the following. For any given $y \in \mathcal{Y}$, $y_0 = 1$, satisfying constraints (13b), the assignment subproblem is given as

$$\min\{\sum_{j \in J} \sum_{i \in I(j) \setminus \{j\}} d_j c_{ji} w_{ji} : \text{ (7a)-(7c) } \}. \tag{14}$$

Constraints (13b) guarantee that for any choice of $y \in \mathcal{Y}$ s.t. $y_0 = 1$, the linear program (14) is always feasible, i.e., they act as Benders feasibility cuts. Hence the LP (14) is well defined and attains its optimum. This LP is separable per $j \in J$, and can be solved by $|J|$ independent LPs, one for each j. The optimal solution value of the LP associated to customer j expresses the cost needed to serve this customer, is a function of the vector y. Starting from LP-duals of these $|J|$ LPs, constraints (13a), which correspond to Benders optimality cuts, are derived. We notice that for $j \in I \cap J$, if $y_j = 1$, constraints (13a) impose $v_j \geq 0$.

Although there is a polynomial number of constraints (13a), very few of them are actually binding in an optimal solution. Hence, it pays off to dynamically separate them in a branch-and-cut fashion. In this case, we can initialize the lower bounds on v variables as follows:

$$v_j \geq \left(\min_{i \in I(j), i \neq j} c_{ji} \right) (1 - y_j) \qquad j \in J \cap I \tag{15}$$

$$v_j \geq \min_{i \in I(j)} c_{ji} \qquad j \in J \setminus I \tag{16}$$

These constraints state that the assignment cost will be at least the cost to the closest facility (and in the case of node $j \in J \cap I$, if $y_j = 0$, this facility must be different than j).

2.3. Overview of models for the MPIF

In this section we have proposed three ways to model interconnectivity between open facilities: two compact approaches based on SCF and MCF, respectively, and an approach that uses an exponential number of node-separator inequalities (11). We also proposed two ways to model assignment between customers and facilities: the first one is the standard model (7) with w variables, and the second one uses Benders reformulation with variables v. Thus, by combining the network design and assignment parts, we obtain six different MIP models. They are summarized in Table 1. When referring to these models, letters S, M and N stand for SCF, MCF and node-separator network design subformulation, respectively. Letters W and V stand for assignment subformulation with w and v variables, respectively.

Model	Objective	Network Design	Assignment	Variables
MPIF SW	(3)	Single-commodity flow (9)	Compact (7)	y, f, w
MPIF MW	(3)	Multi-commodity flow (10)	Compact (7)	y, h, w
MPIF NW	(3)	Node-separator cuts (11)	Compact (7)	y, w
MPIF SV	(12)	Single-commodity flow (9)	Compact (13)	y, f, v
MPIF MV	(12)	Multi-commodity flow (10)	Compact (13)	y, h, v
MPIF NV	(12)	Node-separator cuts (11)	Compact (13)	y, v

Table 1: Classification of models for the MPIF with references to the equations/inequalities of the objective functions and constraints, respectively

3. Mathematical models for the Covering Problem with Interconnected Facilities (CPIF)

The objective function of the CPIF models a trade-off between the cost of opening facilities, and the revenues that can be collected from offering a service to customers. This time we do not incur costs for assigning customers, hence we do not need to know which facility the customer is assigned to, but rather whether the customer is covered or not. Recall that we are still limited by the radius $R > 0$ denoting the customer covering radius of each facility. To model the CPIF, we introduce binary covering variables z, which are defined for each customer $j \in J$ and which are equal to 1 if and only if for customer j there exists an open facility within radius R from j. Binary facility location variables y are the same as for the MPIF.

We now minimize the cost of open facilities plus the penalties for the customers who remain outside of radius R of any open facility. Hence, a generic way of modeling the problem is given as follows:

$$\text{minimize} \quad \alpha \sum_{i \in I} g_i y_i + \sum_{j \in J} d_j (1 - z_j) \tag{17}$$

$$\begin{aligned} \text{s.t. } & y \in \mathcal{Y}, y_0 = 1 \\ & z \in \mathcal{Z}(y) \end{aligned} \tag{18}$$

In the objective function, the second term determines the penalty (which is the sum of the demand of uncovered customers), and the coefficient $\alpha > 0$ is used to balance the facility opening cost and the penalty. In the definition of the objective function, it is assumed that the revenue generated from each customer and hence its significance is proportional to the demand of the customer. This is a customary assumption in the covering location literature (see also related problems described in [10, 21, 37]).

As for the MPIF, with $y \in \mathcal{Y}, y_0 = 1$ we model all possible feasible configurations of interconnected facilities. The set $\mathcal{Z} \subseteq \{0,1\}^{|J|}$ is a set of incidence vectors corresponding to feasible coverage of customers by open facilities. Moreover, as for the MPIF, there exists a cardinality-constrained variant of the problem (under the same name, CPIF as in [7]), in which the number of open facilities is fixed to p :

$$\text{minimize} \quad \{\sum_{j\in J} d_j\,(1 - z_j) : \sum_{i\in I} y_i = p, y \in \mathcal{Y}, y_0 = 1, z \in \mathcal{Z}(y)\}. \tag{19}$$

The feasible region $\mathcal{Y}$ for facilities (variables y) in the CPIF can be defined by the same set of constraints as for the MPIF (cf. Section 2.1). The feasible region $\mathcal{Z}(y)$ for covering variables z is defined by (20b) and (20a).

$$z_j \le \sum_{i\in I(j)} y_i \qquad j \in J \tag{20a}$$

$$z_j \in \{0,1\} \qquad j \in J \tag{20b}$$

Constraints (20a) ensure that $z_j = 1$ (i.e., customer $j \in J$ is covered) only if there is at least one open facility within the radius R from j.

We can model penalty term in the objective function explicitly, using the covering variables z in a compact way with constraints (20). Alternatively, we can use a Benders decomposition approach to project out z variables and track the total demand of uncovered customers in a non-compact formulation, as described below.

3.1. Benders approach to model the non-covering penalty

If we introduce variable $\theta = \sum_{j\in J} z_j$ representing the total covered demand, the objective of the CPIF (17) becomes:

$$\text{minimize} \quad \alpha \sum_{i\in I} g_i y_i + \sum_{j\in J} d_j - \theta \tag{21}$$

where the value of θ can be bounded as

$$0 \le \theta \le \sum_{j\in J} d_j. \tag{22}$$

The Benders subproblem associated with a solution $\tilde{y} \in \mathcal{Y}, \tilde{y}_0 = 1$ corresponds to

$$\max\{\sum_{j\in J} d_j z_j : z_j \le \sum_{i\in I(j)} \tilde{y}_i, z_j \le 1, j \in J\}. \tag{23}$$

This subproblem is always feasible, it is separable per each $j \in J$, and its optimal solution is determined as $z_j = \min\{1, \sum_{i\in I(j)} \tilde{y}_i\}$. Using this observation, for a given solution $\tilde{y}$ of the master problem, let $\tilde{I}_j := \sum_{i \in I(j)} \tilde{y}_i$. Then, the following Benders optimality cuts can be derived for the CPIF, in a similar fashion as it has been done for the related partial set covering problem in [10]:

$$\theta \leq \sum_{j \in J : \tilde{I}_j \geq 1} d_j + \sum_{j \in J : \tilde{I}_j < 1} d_j (\sum_{i \in I(j)} y_i) + \sum_{j \in J_s : \tilde{I}_j = 1} d_j (y_{i(j)} - 1). \tag{24}$$

We separate these cuts within a branch-and-Benders-cut procedure. A detailed description of this separation procedure is given in Section 5.

3.2. Overview of models for the CPIF

Depending on how we model the network design or the assignment component of the MIP formulation, we again obtain six different models. As with the MPIF models, we can formulate the interconnectivity in a compact or non-compact way. As shown above, the coverage can also be formulated in a compact way or using Benders cuts. The summary of these models is given in Table 2 where as before, S, M and N stand for SCF, MCF and node-separator formulations, respectively. In addition, letters Z and T denote two ways to model covering, using z variables or θ, respectively.

Model name	Objective	Network Design	Coverage	Variables
CPIF SZ	(17)	Single-commodity flow (9)	Compact (20)	y, f, z
CPIF MZ	(17)	Multi-commodity flow (10)	Compact (20)	y, h, z
CPIF NZ	(17)	Node-separator cuts (11)	Compact (20)	y, z
CPIF ST	(21)	Single-commodity flow (9)	Exponential #constraints (24)	y, f, θ
CPIF MT	(21)	Multi-commodity flow (10)	Exponential #constraints (24)	y, h, θ
CPIF NT	(21)	Node-separator cuts (11)	Exponential #constraints (24)	y, θ

Table 2: Classification of models for the CPIF with references to the equations/inequalities of the objective functions and constraints, respectively

4. Comparing the tightness of flow-based subformulations modeling interconnectivity

In this section, we focus on two compact ways of modeling the interconnectivity. We compare the tightness of underlying MIP models when using the SCF-based constraints, as proposed in [7], versus MCF-based constraints proposed in Section 2.1.2. For a given formulation F and a given input instance, we denote the value of its LP-relaxation as $v_{\text{LP}}(F)$ and the value of the optimal solution as OPT. For two MIP formulations F_1 and F_2, we will say that F_1 is strictly stronger than F_2 if for all problem instances we have $v_{\text{LP}}(F_1) \geq v_{\text{LP}}(F_2)$ and there exist instances for which the strict inequality holds. We will use the following two examples to derive our theoretical results.

Example 1. *Let us consider an example depicted in Figure 2 with the set of facility candidates*

$I = \{0, \dots, n\}$ and customers $J = \{n+1, \dots, n+m\}$ ($n \geq m$). We assume:

$$\begin{aligned}
&E_I = \{0,1\} \cup \{\{1,i\} : i \in \{2,\dots,n\}\} && \\
&I(j) = \{2,\dots,n\} && j \in J \\
&g_1 = L, g_i = K && i \in I \setminus \{0,1\} \\
&d_j = 1 && j \in J \\
&c_{ji} = \varepsilon && j \in J, i \in \{2,\dots,n\}
\end{aligned}$$

where $L \gg K \geq 0$ and $\varepsilon \geq 0$ is close to zero. Due to the definition of $I(j)$, we have to make sure that $\sum_{i\in\{2,\dots,n\}} y_i \geq 1$ so that all customers could be served. Consequently, by the definition of E_I, we have to open a facility at node 1, in order to build a feasible network.

Example 2. *For the set of facility candidates $I = \{0, \dots, n\}$ and single customer $J = \{n+1\}$ shown in Figure 3, we have*

$$\begin{aligned}
&E_I = \{\{i,i+1\} : i \in \{0,\dots,n-1\}\} && \\
&I(j) = \{n\} && j \in J \\
&g_i = K && i \in I \setminus \{0\} \\
&d_j = 1 && j \in J \\
&c_{n+1\ n} = \varepsilon &&
\end{aligned}$$

To obtain a feasible solution we have to open all facilities so that there is a path from the root node to facility n so that customer $n+1$ can be served.

In the following we compare the quality of LP-relaxation bounds of the formulations SW and MW on these two examples. While we focus on the MPIF, we point out that the results can be easily translated into related ones for the CPIF if we assume big-M demand for the customer nodes, since we assume negligible transportation costs.

It is known in the network design literature (see, e.g., [20, 29]) that SCF-based formulations can produce very weak lower bounds. In our case this is caused by the bigM-constraints (9c). On the other hand, MCF-based formulations with capacities on arc variables are known to provide much tighter lower bounds. For example, for the Steiner tree problem in graphs and some of its variants, the value of the LP-relaxation of the MCF-based formulation is never below 1/2 of the optimal value (see, e.g., [27, 29]). The following proposition shows a surprising result that MCF-based formulations with capacities on node variables do not necessarily improve the quality of lower bounds obtained by the SCF-based model. Specifically, we show that there exist MPIF instances for which the LP-relaxation value of the MW model can be arbitrarily bad. Since in all models summarized above, the assignment component is modeled independently, without loss of generality we can focus on models SW and MW, and the results remain the same when comparing SV against MV.

Proposition 2. *We have:*

$$\inf \frac{v_{\mathrm{LP}}(\mathit{MPIF\ MW})}{\mathrm{OPT}} \leq \frac{1}{|I|-2} \ \text{and} \ \inf \frac{v_{\mathrm{LP}}(\mathit{MPIF\ SW})}{\mathrm{OPT}} \leq \frac{1}{|I|-2}$$

where the infimum is calculated over all possible MPIF instances.

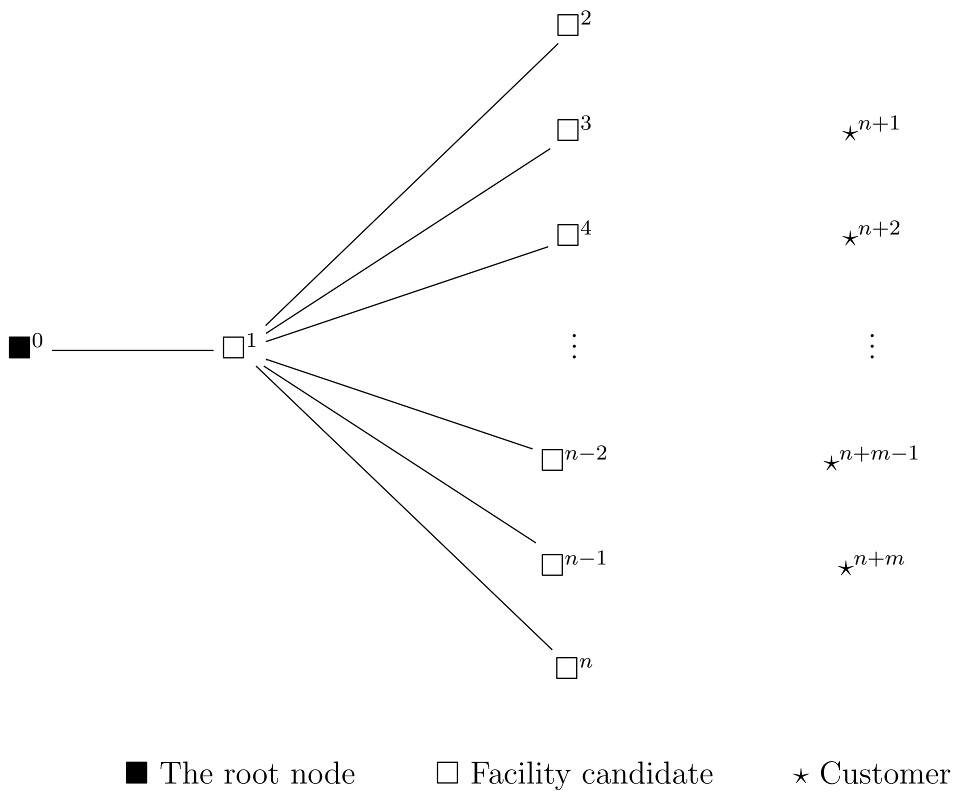


Figure 2: Example instance 1. Squares represent candidate facility nodes, stars are customers. Node 0 is the root node. Node 1 can be connected as a facility to both node 0 and nodes $\{2, ..., n\}$, but cannot serve customers. Thus we force at least one of nodes $\{2, ..., n\}$ to be open. The cost to build a facility at node 1 is L, at nodes $\{2, ..., n\}$ is K.

Proof. We will use the previously introduced Example 1, where $|I| = n + 1$, to illustrate an instance for which the above ratios are attained. We have:

$$\text{OPT} = L + K + m \cdot \varepsilon \approx L \text{ under our assumption } L \gg K \geq 0 \text{ and } \varepsilon \text{ close to } 0.$$

$$v_{\text{LP}}(\text{MPIF SW}) = \frac{1}{n-1}L + K + m \cdot \varepsilon \approx \frac{1}{n-1}L$$

$$v_{\text{LP}}(\text{MPIF MW}) = \frac{1}{n-1}L + K + m \cdot \varepsilon \approx \frac{1}{n-1}L.$$

We explain in Appendix A, how the corresponding LP-optimal solutions are obtained. Hence, this example shows that the value of the LP-relaxation of the two models can be the same, and moreover, the ratio to the optimal solution value is $1/(n-1)$. □

The following result shows that the LP-relaxation of the SW formulation can be arbitrarily worse than the respective value of the MW formulation.

Proposition 3. *It holds:*

$$\inf \frac{v_{\text{LP}}(\textit{MPIF SW})}{v_{\text{LP}}(\textit{MPIF MW})} \leq \frac{e}{|I| - 1},$$

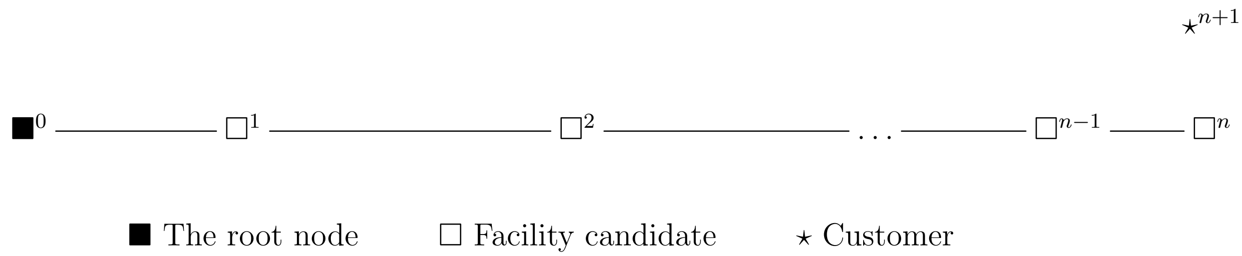


Figure 3: Example instance 2.

where the infimum is calculated over all possible MPIF instances.

Proof. For Example 2, where $|I| = n + 1$, we have:

$$\text{OPT} = nK{+}\varepsilon \approx nK$$

$$v_{\text{LP}}(\text{MPIF MW}) = nK{+}\varepsilon \approx nK$$

$$v_{\text{LP}}(\text{MPIF SW}) = \left(\frac{n}{n-1}\right)^{n-2} K{+}\varepsilon \approx eK$$

We explain in Appendix B, how the corresponding LP-optimal solutions are obtained. □

Corollary 1. *Formulation MPIF MW is strictly stronger than the formulation MPIF SW.*

Proof. We show how to transform an arbitrary LP-solution $(\bar{y}, \bar{h}, \bar{w})$ of the MPIF MW formulation into a feasible LP-solution of the MPIF SW formulation, say $(\tilde{y}, \tilde{f}, \tilde{w})$, without changing the value of the objective function. Indeed, we have $\tilde{y} := \bar{y}$, $\tilde{w} := \bar{w}$ and for each $(k, \ell) \in A_I$ we set $\tilde{f}_{k\ell} = \sum_{i \in I, i \neq 0} \bar{h}^i_{k\ell}$. Hence $v_{\text{LP}}(\text{MPIF SW}) \leq v_{\text{LP}}(\text{MPIF MW})$. Together with the counter example from the proof of Proposition 3, the result follows. □

5. Implementation Details

In the following, we provide implementation details along with pseudo-codes for the separation of constraints used in our branch-and-(Benders)-cut implementations. Algorithm 1 shows the general framework applied to models MPIF NW/NV and CPIF NZ/NT, in case no cardinality constraints are imposed by the input data. Within this framework, we precompute small cardinality solutions by assuming that at most two facilities (the root node together with one of its neighbors) are open, which allows us to obtain an initial feasible solution. The value of this solution is then passed as a cutoff value to Cplex together with an additional constraint that at least three facilities must be open. General purpose MIP solvers have very powerful techniques for finding feasible solutions which work particularly well for compact MIP formulations (when the full model is given to the solver). However, these techniques are way less effective for branch-and-cut approaches (when only a small subset of valid inequalities is provided to the solver, and the remaining ones are dynamically generated). Hence, OPT1 technique allows us to circumvent these difficulties in two ways: we obtain an initial feasible solution and we restrict the search space for the branching tree.

We recall that the model MPIF NW is solved using a branch-and-cut, whereas the model MPIF NV is solved using a branch-and-Benders-cut. Algorithm 2 shows the cut separation process for intermediate (and potentially infeasible) solution vectors for both models. Algorithm 2 is called after solving the relaxed master problem, or after finding an "incumbent" solution (i.e., a solution that satisfies integrality conditions together with all constraints of the relaxed master problem). For each such solution of the master problem, we generate a set `Cuts` of violated constraints, that are used to discard the solution, in case it is not valid. We first separate node-separator constraints (11), and only if there are no more of these cuts violated, we resort to the separation of Benders cuts (13a). For the MPIF NW model, the steps 17-24 are skipped. Minimal node-separator cuts (11) are separated at binary points only. Given a binary vector $\tilde{y}$ indicating a set of open facilities, we build a support graph $G_{\tilde{I}}$ induced by the nodes i such that $\tilde{y}_i = 1$. In our notation, sets $C_i \subseteq I$ refer to the subset of facilities reachable from node i in $G_{\tilde{I}}$ (directly or through other opened facilities for a given binary solution). Similarly, C_0 refers to the connected component in $G_{\tilde{I}}$ containing the root node. The set $A(C_0)$ refers to the set of closed facilities which are not in the connected component of 0, but are direct neighbors of some open facility from C_0. Finally, we also consider R_i as the set of nodes reachable from i, after removing from the original graph G_I the set of edges connecting nodes in $C_0 \cup A(C_0)$. Minimal node-separators between two components, C_0 and C_i, are obtained in the intersection of sets R_i and $A(C_0)$.

Similarly, the model CPIF NZ is solved using a branch-and-cut, whereas the model CPIF NT is solved using a branch-and-Benders-cut. Algorithm 3 shows the cut separation process for intermediate (and potentially infeasible) solution vectors for both models. In the implementation of the CPIF NZ model, the steps 16-23 are skipped. Also for the CPIF, we resort to separation of Benders cuts, only if no more violated cuts of type (11) can be found.

Algorithm 1: Algorithm for MPIF NV/NW, the CPIF NT/NZ models (for instances without the cardinality constraint)

Data: Instance of the MPIF or the CPIF problem without the cardinality constraint
Result: Optimal solution, or best found solution (if timelimit is reached)

Find by inspection the optimal solution OPT_0 when only the facility at the root node is open
Find by inspection the optimal solution OPT_1 when only two facilities are open: the facility at the root node and a single additional facility reachable from the root
Add constraint $\sum_{i \in I} y_i \geq 3$ to the respective model
Set the upper cutoff tolerance of CPLEX solver to $\min(\mathrm{OPT}_0, \mathrm{OPT}_1)$
Call Branch-and-(Benders)-Cut algorithm to find OPT
return $\min(\mathrm{OPT}_0, \mathrm{OPT}_1, \mathrm{OPT})$

Algorithm 2: Separation algorithm for the MPIF NV/NW model

Data: Solution vector $(\tilde{y}, \tilde{v}) \in [0,1]^{|I|} \times \mathbb{R}_+^{|J|}$ (MPIF NV model), or
$(\tilde{y}, \tilde{w}) \in [0,1]^{|I|} \times [0,1]^{|I| \times |J|}$ (MPIF NW model)

Result: `Cuts`: Set of violated cuts

`Cuts` $\leftarrow \emptyset$
if $\tilde{y} \in \{0,1\}^{|I|}$ **then**
  Let $\tilde{I} = \{i \in I : \tilde{y}_i = 1\}$ and let $G_{\tilde{I}}$ be the subgraph of G_I obtained after removing from G_I all nodes i such that $\tilde{y}_i = 0$
  Let C_0 be the nodes of the connected component in $G_{\tilde{I}}$ containing the root node
  Let $A(C_0) \subseteq I \setminus \tilde{I}$ be the neighboring nodes of C_0 in G_I
  Using BFS on $G_{\tilde{I}}$ and G_I, find C_0 and $A(C_0)$, respectively
  Delete all edges connecting nodes of $C_0 \cup A(C_0)$ from G_I
  if $C_0 \neq \tilde{I}$ **then**
    $O \leftarrow \tilde{I} \setminus C_0$
    for $i \in O$ **do**
      Using BFS on $G_{\tilde{I}}$, find C_i, the subset of nodes reachable from i
      Using BFS on G_I, find R_i, the subset of nodes reachable from i
      Let $\tilde{N}_i = R_i \cap A(C_0)$
      for $j \in C_i$ **do**
        Add $\sum_{i' \in \tilde{N}_i} y_{i'} \geq y_j$ to `Cuts`.
        $O \leftarrow O \setminus j$
  else if *MPIF NV Model* **then**
    for $j \in J, k \in I(j)$ **do**
      if $\tilde{v}_j < c_{jk} - \sum_{i \in I(j)} (c_{jk} - c_{ji})^+ \tilde{y}_i$ **then**
        add $v_j \geq c_{jk} - \sum_{i \in I(j): c_{ji} < c_{jk}} (c_{jk} - c_{ji}) y_i$ to `Cuts`.
else if *MPIF NV Model* **then**
  for $j \in J, k \in I(j)$ **do**
    if $\tilde{v}_j < c_{jk} - \sum_{i \in I(j)} (c_{jk} - c_{ji})^+ \tilde{y}_i$ **then**
      add $v_j \geq c_{jk} - \sum_{i \in I(j): c_{ji} < c_{jk}} (c_{jk} - c_{ji}) y_i$ to `Cuts`.
return `Cuts`

6. Computational results

In this section we assess the empirical performance of our models and the respective solution methods for solving benchmark instances originally introduced by Cherkesly et al. [7]. To test the scalability of the proposed approaches, we also consider an additional dataset, used for solving the related covering location problems. This dataset, used by [10] and based on the procedures originally proposed by [34], contains up to 100,000 customers. In what follows, we first describe the datasets and the tested algorithmic frameworks. After that, we provide a detailed analysis of the obtained results for both MPIF and CPIF instances.

The code was implemented in C++. Experiments were run on a computer equipped with an Intel Core i7 (4790) 3.6 GHz CPU and 16 GB of RAM running on Linux. CPLEX 20.1.0.0

Algorithm 3: Separation algorithm for the CPIF NT/NZ model.

Data: Solution vector $(\tilde{y}, \tilde{\theta}) \in [0,1]^{|I|} \times \mathbb{R}_+$ (CPIF NT model), or $(\tilde{y}, \tilde{z}) \in [0,1]^{|I|} \times [0,1]^{|J|}$ (CPIF NZ model)

Result: `Cuts`: Set of violated cuts

if $\tilde{y} \in \{0,1\}^{|I|}$ **then**
    Let $\tilde{I} = \{i \in I : \tilde{y}_i = 1\}$ and let $G_{\tilde{I}}$ be the subgraph of G_I obtained after removing from G_I all nodes i such that $\tilde{y}_i = 0$
    Let C_0 be the nodes of the connected component in $G_{\tilde{I}}$ containing the root node
    Let $A(C_0) \subseteq I \setminus \tilde{I}$ be the neighboring nodes of C_0 in G_I
    Using BFS on $G_{\tilde{I}}$ and G_I, find C_0 and $A(C_0)$, respectively
    Delete all edges connecting nodes of $C_0 \cup A(C_0)$ from G_I
    if $C_0 \neq \tilde{I}$ **then**
        $O \leftarrow \tilde{I} \setminus C_0$
        for $i \in O$ **do**
            Using BFS on $G_{\tilde{I}}$, find C_i, the subset of nodes reachable from i
            Using BFS on G_I, find R_i, the subset of nodes reachable from i
            Let $\tilde{N}_i = R_i \cap A(C_0)$
            for $j \in C_i$ **do**
                Add $\sum_{i' \in \tilde{N}_i} y_{i'} \geq y_j$ to `Cuts`.
                $O \leftarrow O \setminus j$
    else if *CPIF NT model* **then**
        $\bar{I}_j \leftarrow \sum_{i \in I(j)} \tilde{y}_i$
        if $\tilde{\theta} > \sum_{j \in J: \bar{I}_j \geq 1} d_j + \sum_{j \in J: \bar{I}_j < 1} d_j \bar{I}_j$ **then**
            add $\theta \leq \sum_{j \in J: \bar{I}_j \geq 1} d_j + \sum_{j \in J: \bar{I}_j < 1} d_j (\sum_{i \in I(j)} y_i) + \sum_{j \in J_s: \bar{I}_j = 1} d_j (y_{i(j)} - 1)$ to `Cuts`
else if *CPIF NT model* **then**
    $\bar{I}_j \leftarrow \sum_{i \in I(j)} \tilde{y}_i$
    if $\tilde{\theta} > \sum_{j \in J: \bar{I}_j \geq 1} d_j + \sum_{j \in J: \bar{I}_j < 1} d_j \bar{I}_j$ **then**
        add $\theta \leq \sum_{j \in J: \bar{I}_j \geq 1} d_j + \sum_{j \in J: \bar{I}_j < 1} d_j (\sum_{i \in I(j)} y_i) + \sum_{j \in J_s: \bar{I}_j = 1} d_j (y_{i(j)} - 1)$ to `Cuts`
return `Cuts`

was used as a general-purpose MIP solver. We used callback functions and limited all runs to a single thread. The time limit was set to 3600s. All other CPLEX parameters are kept at their default values, except for the number of threads (which was set to one). For the branch-and-cut methods, reduction and dual reduction techniques performed by the CPLEX were turned off as well. The implementation and the results are also publicly available at `https://github.com/b00750186/PIF`.

6.1. Benchmark set of instances

Our algorithms are applied to the following two datasets which were used in [7]:

- *Euclidean instances*: For 29 instances from this set (see `http://www.math.nsc.ru/AP/benchmarks/UFLP/Engl/uflp_eucl_eng.html`), each customer node can be used as a facility as well, and we have $|I| = |J| = 100$. The values c_{ji} were calculated as Euclidean distances between each pair of nodes (rounded to the nearest integer). The fixed cost of each facility is equal to 3000 and no cardinality p on the number of open facilities is imposed. The values for r (as used in [7]) are chosen from $\{500, 1500, 2500\}$. For the MPIF, $R = \infty$, whereas for the CPIF we have $R \in \{1000, 1500, 2000\}$. Figure 4 illustrates optimal solutions for one of the Euclidean instances with different radii. Overall, there are 87 MPIF and 261 CPIF instances in this group.

- *p-median instances*: This is a set of 40 instances with 100 to 900 nodes ($|I| = |J|$), with the values c_{ji} and possible connections between facilities given as input graphs (see `http://people.brunel.ac.uk/~mastjjb/jeb/info.html`). The number p of facilities that need to be open ranges between five and 200. As proposed in [7], for instances 1 to 6 from this set, we choose $r \in \{80, 100, 150\}$, for instances 7 to 9, we have $r \in \{50, 80, 100\}$ and for instances 10 to 40, we have $r \in \{25, 50, 80\}$. For the MPIF, $R = \infty$, whereas for the CPIF the values of R are chosen from $\{8, 10, 12, 15, 20\}$. Overall, there are 75 MPIF instances and 555 CPIF instances in this group.

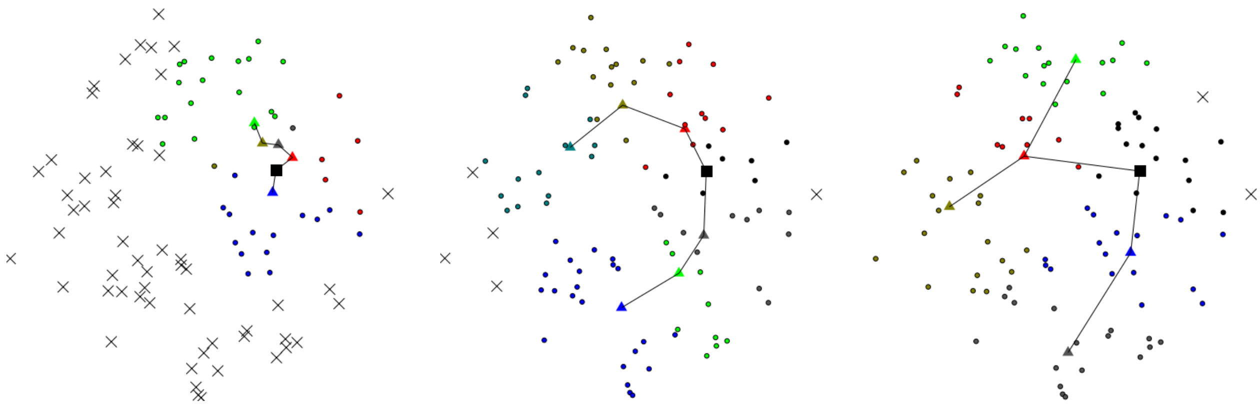
For both sets of instances, the demand of each single customer is set to 1 and the value of α is set to 0.001.

While the former two datasets contain the same number of customers and facility locations, in some applications (when facilities correspond to locations of antennas, sensors, or similar telecommunication devices), the number of potential facility locations can be fairly limited when compared to the number of customers that need to be covered. Hence, to test the scalability of the proposed approaches, we also consider an additional dataset from [10], see also [34].

- *Covering dataset*: Starting with 20 instances from the dataset available at `https://github.com/fabiofurini/LocationCovering`, we generate 50 MPIF and 100 CPIF instances as follows. The coordinates of the potential facility locations (together with their opening cost) and the set of customers (together with their demand) are given. The number of facilities is $|I| = 100$, and the number of customers is $|J| \in \{1000, 5000, 10000, 50000, 100000\}$. The values c_{ji} were calculated as Euclidean distances between each pair of nodes. The fixed cost of each facility is between 10 and 100 and no cardinality p on the number of open facilities is imposed. The demand of each customer is an integer number between 1 and 100. We choose the values for r from $\{3.4, 5.5\}$ as in the preliminary tests, we have observed that these radii are more difficult to solve. For the MPIF, we set $R = \infty$, and for the CPIF we set $R \in \{4, 6\}$. The value of α is equal to 1.

6.2. Comparing the computational performance of proposed models

To compare the models and algorithms proposed in this paper, we consider the following two indicators: the runtime needed to solve benchmark instances to optimality, and the final gaps at termination (in case the time limit of 3600 seconds was reached). For the latter gap, we use formula from CPLEX node report, namely gap[%] := (UB − LB) · 100/UB, where UB and LB refer to the global upper, respectively, lower bound of the model at the termination.



(a) $r = 500$ (b) $r = 1500$ (c) $r = 2500$

Figure 4: Optimal solutions for an instance of the CPIF for three different values of r. The black square represents the root node. Triangles represent open facilities. Colored circles represent customers covered by the facility of the same color. Crosses stand for uncovered customers. Lines connect open facilities that lie within the radius r. Instance file: Euclidean 2111, $R = 2000$.

We use default MIP tolerance gap of 0.01%. Hence, whenever the obtained gap is < 0.01%, it is reported as 0% in our experiments.

In case a method could not obtain any integer solution (i.e., UB $= \infty$) we recalculate UB by assuming that: (a) for the MPIF, all customers are assigned to the root node (which is always possible, since $R = \infty$ for the given benchmark set), and (b) for the CPIF, no customers are served and no facilities are open. If a method does not solve the LP-relaxation within the time limit, we set $LB = 0$ and gap = 100%.

6.2.1. MPIF

We compare five different ways to solve models from Table 1:

- **SW (compact):** This is the MPIF SW model solved by the default CPLEX MIP algorithm.
- **SW (auto Benders):** This is the MPIF SW model solved using the automatic Benders decomposition available in CPLEX.
- **MW (compact):** This is the MPIF MW model solved by the default CPLEX MIP algorithm.
- **NW Branch-and-Cut:** This is the MPIF NW model solved by a branch-and-cut in which node-separator cuts (11) are dynamically generated (using LazyCallback function of CPLEX).
- **NV Branch-and-Cut:** This is the MPIF NV model solved by a branch-and-cut in which node-separator cuts (11) are separated (using LazyCallback) and assignment cuts (13a) are separated (using both LazyCallback and UserCallback functions).

Due to the size of the MPIF MW model, we could solve only instances with 100 nodes on a computer with 16 GB RAM, so we did not consider other MCF-based models in this computational study. As explained in Algorithm 1, for instances without the cardinality constraints on the number of open facilities (namely, the Euclidean and the Covering dataset), we pre-calculate optimal solutions with a single open facility (apart from the root node) or with only one open facility at the root node, and then we run the respective MIP models with additional constraint $\sum_{i \in I} y_i \geq 3$ and with the lower cut-off value set to OPT_1.

A summary of obtained results for the Euclidean and p-median datasets is provided in Table 3, whereas Table 4 summarizes the results for the Covering dataset. We report the number of instances solved to optimality within the time limit out of the total number of instances of the respective dataset (column Optimal), the number of instances for which the gap at termination remains positive (column Suboptimal), the average gap at termination over all instances (column gap[%]), and the average runtime in seconds over all instances (column Time (s)). The best performance is achieved by two methods: the NV Branch-and-Cut and the NW Branch-and-Cut. In the NV Branch-and-Cut method, we use node-separator cuts to ensure interconnectivity and use variables v_j to represent the cost of serving customer $j \in J$. In the NW Branch-and-Cut, we have the same approach to model facility network as in NV but use explicit assignment variables w. All 162 Euclidean and p-median and all 50 Covering instances were solved within the time limit of 3600 seconds by each of the two methods. In terms of the average runtime, on the Euclidean and p-median dataset, NV Branch-and-Cut exhibits a slightly better performance (12 secs on average versus 19 secs for the NW Branch-and-Cut method). Such a similar performance between these two models can be explained by the fact that $|I| = |J|$ in these instances. However, in terms of runtime on the Covering dataset, NV Branch-and-Cut significantly beats NW Branch-and-Cut. Indeed, this result confirms the fact that the real benefit of projecting out w variables can be seen on instances for which $|J| \gg |I|$. In that case, the Benders reformulation allows to reduce the number of assignment variables by orders of magnitude, whereas when $|I| = |J|$, the reduction in the number of variables is negligible, and hence the two models (with and without Benders decomposition) perform similarly. The remaining three solution methods from this experiment faced more difficulties with the considered benchmark sets. More precisely, the two SW-based methods are an order of magnitude slower. Indeed, no matter if the SW model is solved using a compact formulation, or automatic Benders decomposition, at least 7 instances out of 162 Euclidean and p-median, and 10 out of 50 Covering instances, remained unsolved. Covering instances seem to be particularly challenging for this single-commodity-flow-based model where the average gaps at termination are higher than 20%. The compact model MW exhibits the worst performance, mostly due to the fact that for instances with more than 100 nodes not even the LP-relaxation was solved within the time limit.

In the following, we report the empirical cumulative distribution functions (ECDFs) w.r.t. the runtimes and percentage gaps at termination, respectively. The ECDFs with e.g., runtimes can be interpreted as the number of instances (shown in y-axis) that can be solved within a certain amount of time (depicted in the x-axis).

In Figures 5 and 6 the number of MPIF instances versus the time needed to solve them to optimality is given. From Figures 5 and 6 we can see that both NW and NV Branch-and-Cut methods dominate the others, but there are instances that are solved faster by one and other instances that are solved faster by the other method. Figure 5 illustrates the performance of our models in terms of runtime on the first dataset that we tested with Euclidean and

p-median instances. The NV method spends less than 500 seconds on each instance, NW spends more time on one instance but stays within 1000 seconds limit. The MW model cannot resolve about 50% of the instances within 3600 seconds time limit due to the size of the instances. The SW model solved by the automatic Benders method is slightly dominated by the SW model solved by the default CPLEX MIP solver. All models except the MW show comparable results on easier 100 instances, while the hardest 62 instances really separate them in terms of computational performance. Similarly, Figure 6 summarizes the performance of our models on the second dataset that we tested with Covering instances. NV and NW models outperform other models spending less than 2000 seconds on each instance. This dataset clearly differentiates models based on their computational performance. The MW model and the SW with automatic Benders seem to show comparable results for some easier instances.

Figures 7 and 8 summarize the performance of our models on the first dataset (Euclidean and p-median instances) and the second dataset (Covering instances), respectively, in terms of the gap at termination. NV and NW models solve all instances to optimality with 0 % gap. The gaps obtained by the MW model and the SW model were reported as 100 % for some instances, as for these instances these methods cannot find any feasible integer solution.

Model	Method	Optimal	Suboptimal	gap[%]	Time (s)
MPIF NV	**NV Branch-and-Cut**	**162/162**	**0/162**	**0.00**	**12**
MPIF NW	**NW Branch-and-Cut**	**162/162**	**0/162**	**0.00**	**19**
MPIF SW	SW (compact)	155/162	7/162	0.1	345
	SW (auto Benders)	148/162	14/162	8.3	427
MPIF MW	MW (compact)	87/162	75/162	27.1	1796

Table 3: MPIF instances solved by different methods within 3600 s. (Euclidean and p-median instances)

Model	Method	Optimal	Suboptimal	gap[%]	Time (s)
MPIF NV	**NV Branch-and-Cut**	**50/50**	**0/50**	**0.0**	**118**
MPIF NW	**NW Branch-and-Cut**	**50/50**	**0/50**	**0.0**	**346**
MPIF SW	SW (compact)	40/50	10/50	20.0	997
	SW (auto Benders)	34/50	16/50	30.7	1483
MPIF MW	MW (compact)	38/50	12/50	24.0	1316

Table 4: MPIF instances solved by different methods within 3600 s. (Covering dataset)

When it comes to the quqlity of LP-relaxations, for the Euclidean instances, on average, solution values of the LP-relaxed MPIF MW model are 0.41% higher than that of the LP-relaxed MPIF SW model.

6.2.2. CPIF

We compare five solution methods applied to CPIF instances:

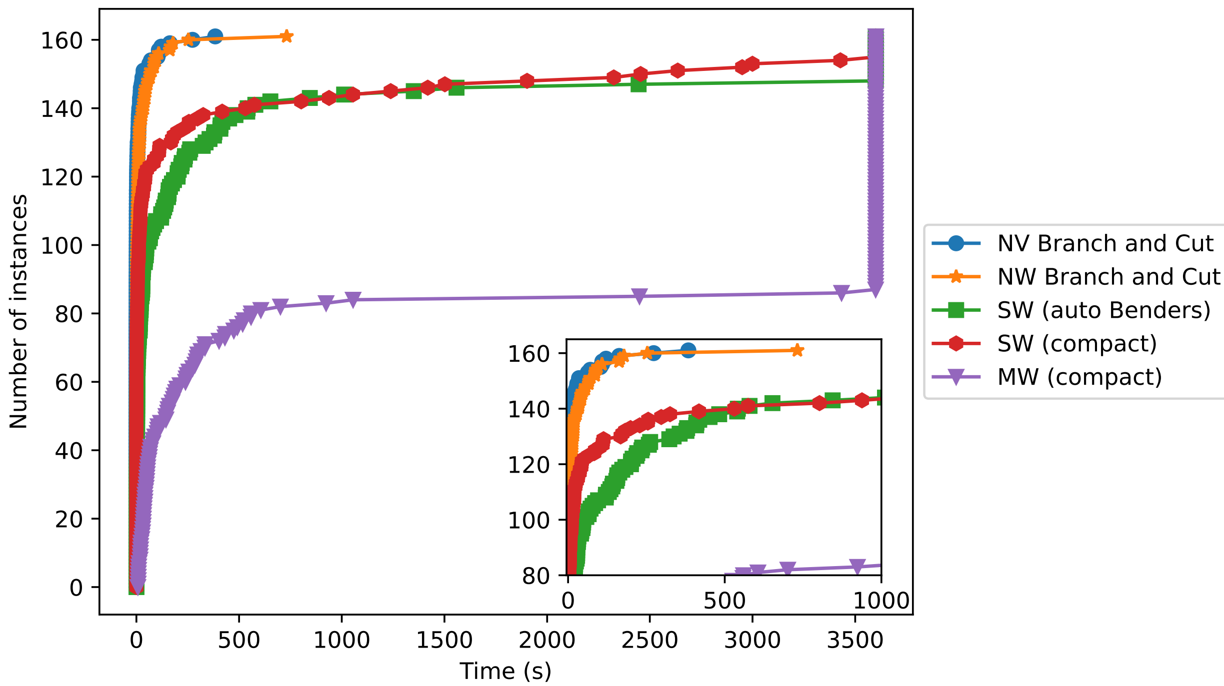


Figure 5: ECDF comparing runtime (in seconds) spent by different methods to solve 162 MPIF instances (Euclidean and p-median dataset).

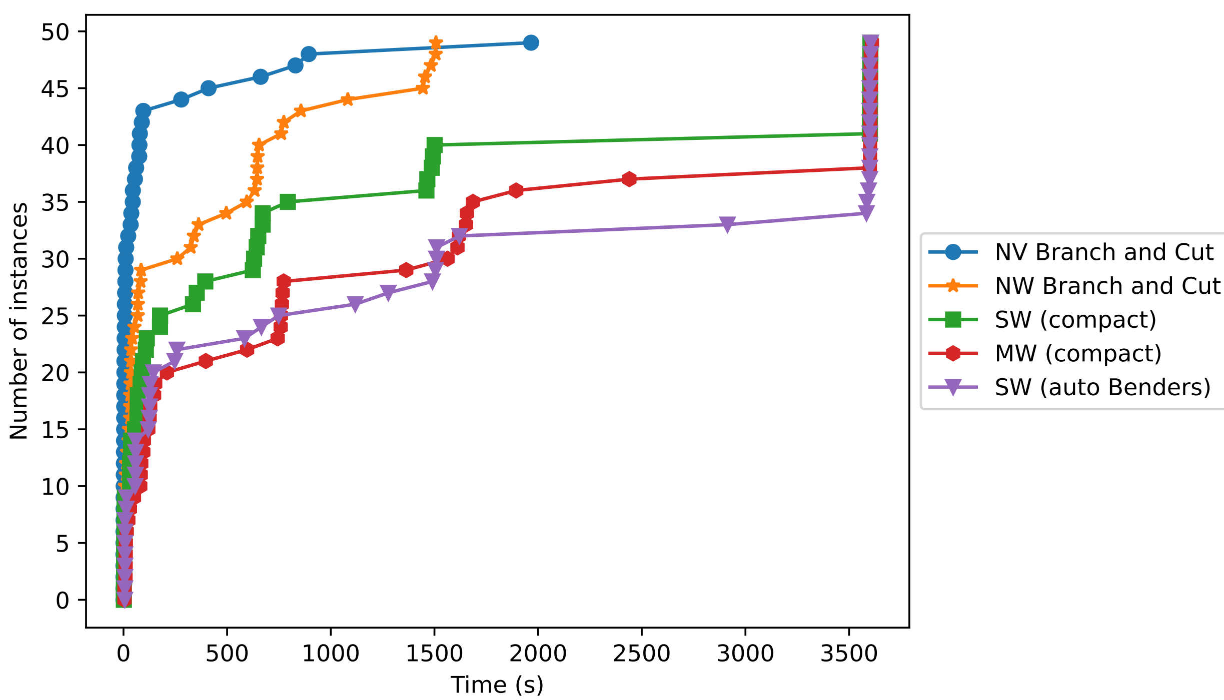


Figure 6: ECDF comparing runtime (in seconds) spent by different methods to solve 50 MPIF instances (Covering dataset).

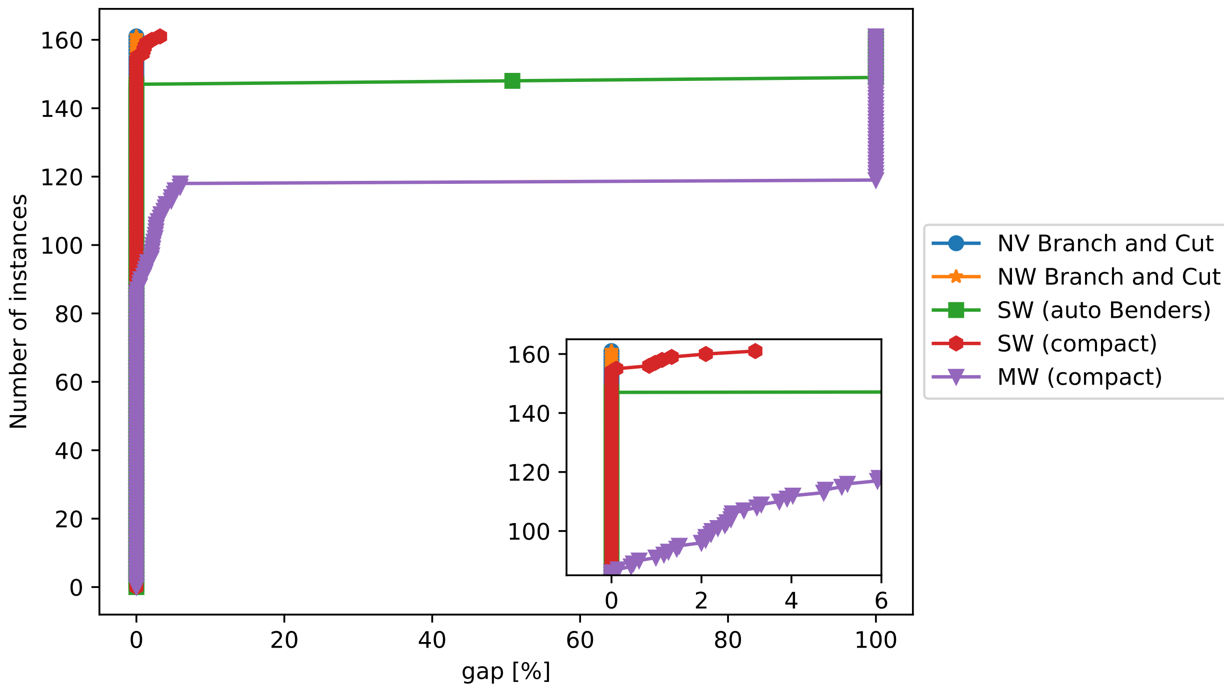


Figure 7: ECDF comparing gaps at termination (gap[%]) for 162 MPIF instances (Euclidean and p-median dataset).

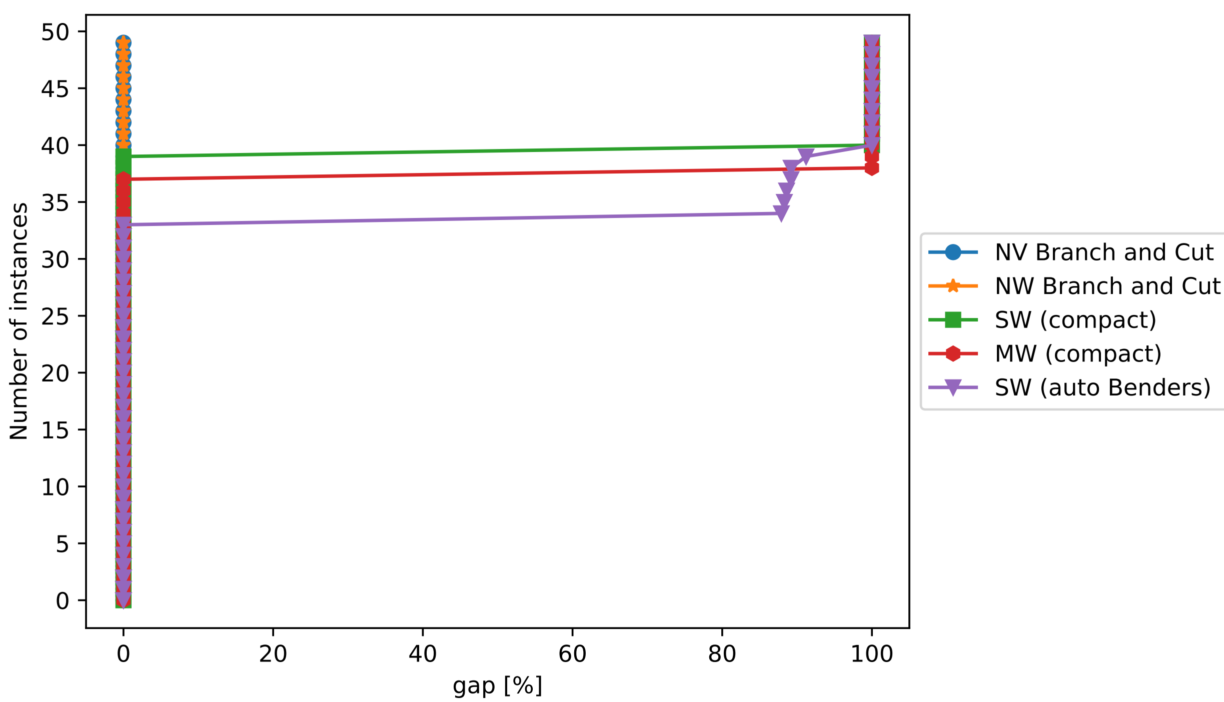


Figure 8: ECDF comparing gaps at termination (gap[%]) for 50 MPIF instances (Covering dataset).

- **SZ (compact):** This is the CPIF SZ model solved by the default CPLEX MIP algorithm.

- **SZ (auto Benders):** This is the CPIF SZ model solved using automatic Benders decomposition available in CPLEX.
- **MZ (compact):** This is the CPIF MZ model solved by the default CPLEX MIP algorithm.
- **NZ Branch-and-Cut:** This is the CPIF NZ model solved by a branch-and-cut in which node-separator cuts (11) are dynamically generated (using LazyCallback function of CPLEX).
- **NT Branch-and-Cut:** This is the CPIF NT model solved by a branch-and-cut with dynamically generated node-separator cuts (using LazyCallback function of CPLEX), Benders cuts (24) (using UserCallback and LazyCallback functions of CPLEX). The latter cuts are added if the violation threshold of 5% is exceeded.

Tables 5-6 provide a summary of obtained results, where we report the same indicators as in Tables 3-4. The best performing method is NZ Branch-and-Cut, where we use node-separator constraints to impose interconnectivity, and model the covering constraints in a compact way with z variables. With this method, all 816 Euclidean and p-median and all 100 Covering instances were solved to optimality within the time limit. However, expressing the covering constraints using Benders cuts as in the NT Branch-and-Cut method worsens the computational performance. This can be explained by the fact that in the benchmark set where we have $|I| = |J|$, potential savings in terms of the reduction of the number of z variables are negligible, and the (computational) cost of introducing a potentially large number of dynamically separated constraints is high. For the Covering dataset (where $|I| \ll |J|$), the performance of the NT Branch-and-Cut method is much better, but still not comparable with the NZ Branch-and-Cut method.

Figures 9 and 10 provide ECDFs reporting the runtime for the Euclidean and p-median instances, and Covering instances, respectively. Similarly, Figures 11 and 12 provide ECDFs reporting the gap at termination for the Euclidean and p-median instances, and Covering instances, respectively. From Figure 9, we observe that only NZ model stays within the time limit, spending less than 3600 seconds on each instance from the set of Euclidean and p-median instances. The second and third best-performing approaches are based on the SZ model. The NT model is only the fourth best, with the corresponding ESDF line going significantly below that of the NZ and the SZ models. The MZ model cannot solve around 60 % of the Euclidean and p-median instances. Moreover, Figure 11 indicates that for around 50% of instances of this dataset, no feasible solution could be found by the MZ model. From Figure 10 we observe that the NZ model solves even faster the instances from the Covering dataset. The NZ model stays within 2500 second limit for each instance, closely followed by the SZ model with automatic Benders that has only one instance left unsolved within 3600 seconds limit. The NT model is the third best. With Covering instances, the NT shows comparable results with the SZ model with automatic Benders. All models can solve more than 80 % of all instances from this benchmark set. For Covering instances, unlike for the Euclidean and p-median instances, all models can find feasible integer solutions (cf. Figure 12).

Model	Method	Optimal	Suboptimal	gap[%]	Time(s)
CPIF NZ	**NZ Branch-and-cut**	**816/816**	**0/816**	**0.0**	**23**
CPIF SZ	SZ (auto Benders)	791/816	25/816	1.2	182
	SZ (compact)	773/816	43/816	0.6	306
CPIF NT	NT Branch-and-cut	584/816	232/816	5.8	1129
CPIF MZ	MZ (compact)	318/816	498/816	51.1	2279

Table 5: CPIF instances solved by different methods within 3600 s. (Euclidean and p-median dataset)

Model	Method	Optimal	Suboptimal	gap[%]	Time(s)
CPIF NZ	**NZ Branch-and-cut**	**100/100**	**0/100**	**0.0**	**55**
CPIF SZ	SZ (auto Benders)	99/100	1/100	0.03	128
	SZ (compact)	89/100	11/100	0.51	516
CPIF NT	NT Branch-and-cut	95/100	5/100	0.15	261
CPIF MZ	MZ (compact)	87/100	13/100	0.79	636

Table 6: CPIF instances solved by different methods within 3600 s. (Covering dataset)

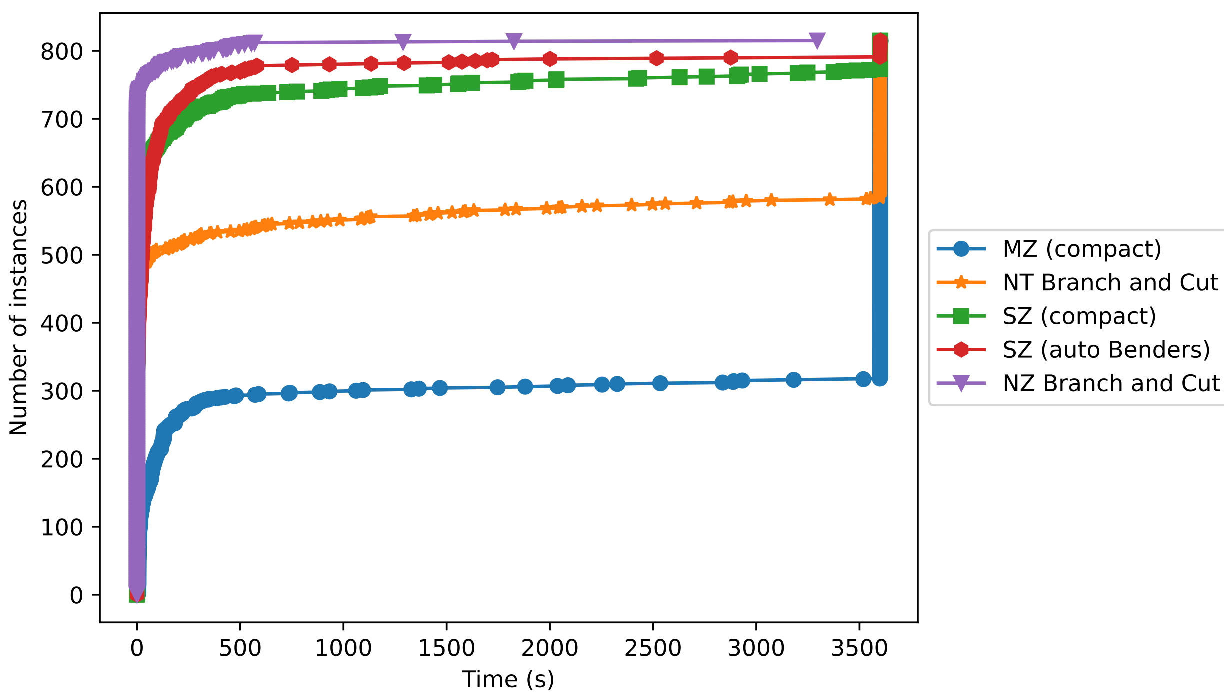


Figure 9: ECDF comparing runtime (in seconds) spent by different methods to solve 816 CPIF instances (Euclidean and p-median dataset).

6.3. Sensitivity analysis

In this section we investigate sensitivity of the empirical performance of the studied methods with respect to the value of r (the interconnectivity radius between open facilities). The

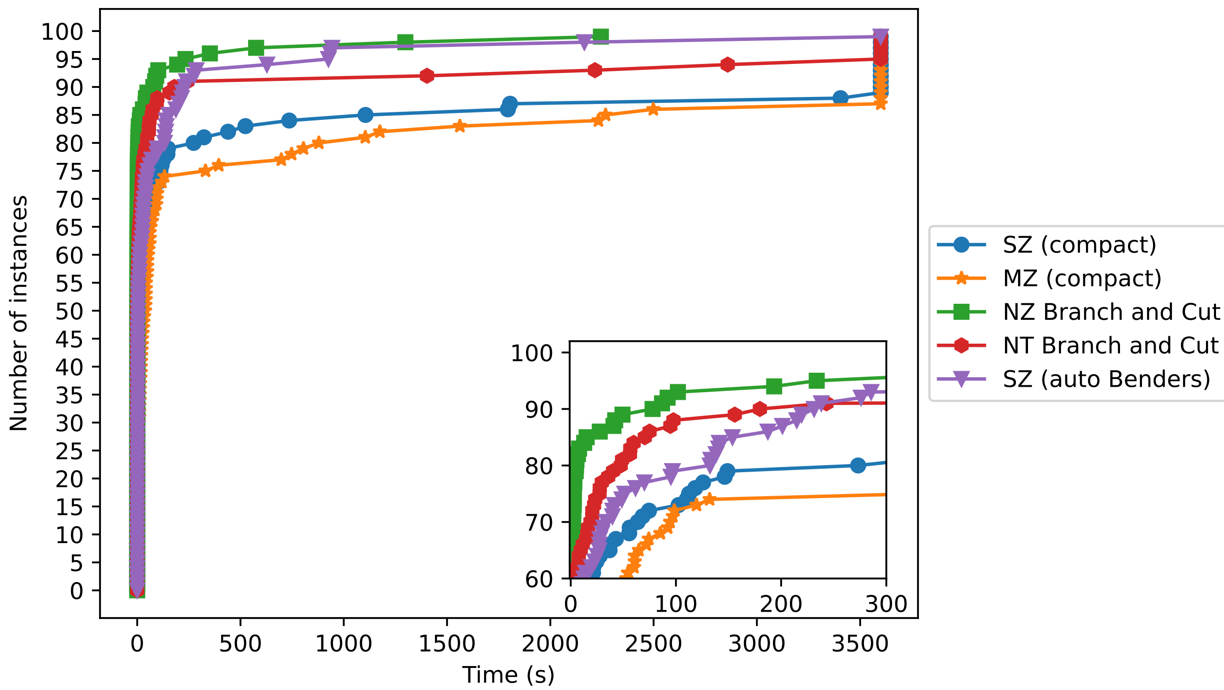


Figure 10: ECDF comparing runtime (in seconds) spent by different methods to solve 100 CPIF instances (Covering dataset).

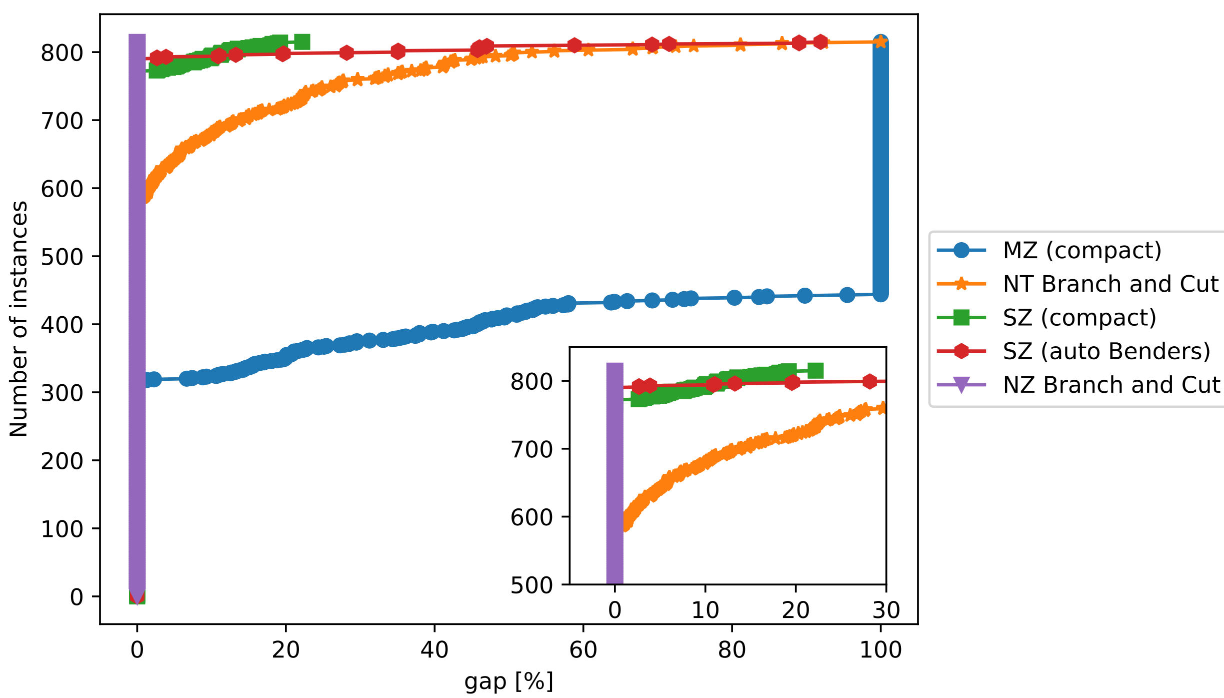


Figure 11: ECDF comparing gaps at termination for 816 CPIF instances (Euclidean and p-median dataset).

analysis is based on the fist benchmark set (Euclidean and p-median instances by Cherkesly et al. [7]).

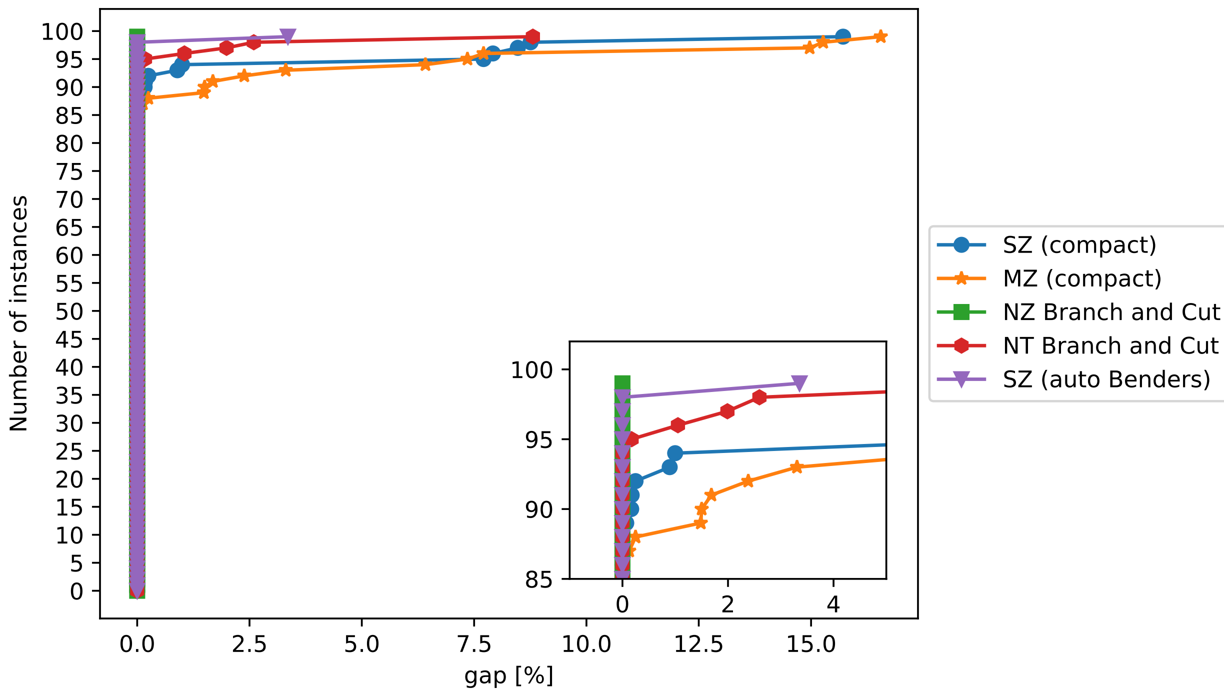


Figure 12: ECDF comparing gaps at termination (gap[%]) for 100 CPIF instances (Covering dataset).

Table 7 reports the number of instances solved to optimality and suboptimality and the respective average runtimes for 87 MPIF instances from the Euclidean dataset, divided into two groups: those with either very small or very large radius, namely $r \in \{500, 2500\}$ and the remaining ones with $r = 1500$. We notice that for all five studied methods, the most challenging instances are those with the interconnectivity radius equal to 1500. Indeed, a small radius implies a limited amount of possible combinations of open facilities (and hence a relatively simple and sparse network G_I of facilities that can be reached from the root node). On the other hand, a large radius implies a densely connected network G_I, and hence a small number of e.g., node-separator cuts that need to be imposed in order to ensure interconnectivity. Recall that if we set r to a sufficiently large value, we obtain the classical facility location problem.

Similarly, the (empirical) difficulty of instances of the CPIF depends primarily on the interconnectivity radius r as we can see in Table 8. Also here, among the Euclidean instances, the most difficult ones are those with $r = 1500$. On the other hand, when we disaggregate the instances based on the value of R, we notice that there are no significant variations in the runtime when it comes to different values of R. Hence, we conclude that the structure of the underlying graph G_I, reflecting possible connections between open facilities, highly affects the difficulty of underlying instances.

7. Conclusions

We introduced new MIP formulations for the MPIF and the CPIF problems using non-compact ways to model interconnectivity, assignment and/or covering constraints. We tested

$r \in \{500, 2500\}$

Model	Method	Optimal	Suboptimal	Total	Time (s)
MPIF NW	NW Branch-and-cut	58	0	58	0.02
MPIF NV	NV Branch-and-cut	58	0	58	0.08
MPIF SW	SW (auto Benders)	58	0	58	2.82
	SW (compact)	58	0	58	0.12
MPIF MW	MW (compact)	58	0	58	134.5

$r = 1500$

Model	Method	Optimal	Suboptimal	Total	Time (s)
MPIF NW	NW Branch-and-cut	29	0	29	49
MPIF NV	NV Branch-and-cut	29	0	29	48
MPIF SW	SW (auto Benders)	29	0	29	475
	SW (compact)	22	7	29	1692
MPIF MW	MW (compact)	1	28	29	3594

Table 7: Sensitivity of different MPIF methods with respect to the interconnectivity radius r (Euclidean dataset)

R	8	10	12	15	20
r = 25	998	1007	1037	1064	1062
r = 50	998	1030	1028	980	1053
r = 80	851	860	856	824	851
r = 100	22	19	98	98	112
r = 150	10	10	13	8	10

R	1000	1500	2000
r = 500	2	2	2
r = 1500	1639	2106	1413
r = 2500	13	175	739

Table 8: Average runtime (in seconds) spent by all five methods reported in Table 5 for CPIF instances with interconnectivity radius r and covering radius R (Euclidean dataset on the right and p-median dataset on the left)

these new models on datasets from the literature and provided some theoretical insights on the quality of lower bounds of these formulations. We implemented multiple branch-and-cut procedures and showed that these tailored approaches outperform off-the-shelf solution methods provided by CPLEX. We also conducted a sensitivity analysis, showing that the values of the interconnectivity radius r strongly affect the empirical difficulty of underlying instances.

The obtained results show that the modeling component that affects the computations the most is related to enforcing the interconnected facilities. To ensure interconnectivity, we have tested two flow-based formulations and a formulation based on node-separator cuts. Our results indicate that dynamical separation of node-separator cuts significantly reduces the runtimes when compared to flow-based compact models. We have also tested Benders decomposition to model the assignment/covering of customers. On the benchmark set in which the potential facility locations coincide with customer locations, no significant computational benefits could be drawn from Benders reformulations. This can be explained by the fact that in such situations no significant reduction in the number of variables could be achieved. However, on the benchmark set in which the number of customers is order(s) of magnitude larger than the number of potential facility locations, Benders decomposition of the assignment component shows its advantages, in particular for the MPIF problem.

Interesting directions for future research include: 1) studying the problem variants under uncertainty (e.g., with respect to uncertain customer demands) in combination with data-driven approaches, 2) taking into consideration capacities on the facilities, or 3) mitigating possible congestion at the facilities.

Acknowledgements

The authors are grateful to Marilène Cherkesly for sharing benchmark instances and detailed information about the computational experiments reported in their paper [7].

References

[1] Z. Alès and S. Elloumi. Compact MILP formulations for the p-center problem. In J. Lee, G. Rinaldi, and A. R. Mahjoub, editors, *Combinatorial Optimization - 5th International Symposium, ISCO 2018, Marrakesh, Morocco, April 11-13, 2018, Revised Selected Papers*, volume 10856 of *Lecture Notes in Computer Science*, pages 14–25. Springer, 2018.

[2] E. Álvarez-Miranda, I. Ljubić, and P. Mutzel. The maximum weight connected subgraph problem. In M. Jünger and G. Reinelt, editors, *Facets of Combinatorial Optimization*, pages 245–270. Springer, 2013.

[3] E. Álvarez-Miranda, I. Ljubić, and P. Mutzel. The rooted maximum node-weight connected subgraph problem. In C. Gomes and M. Sellmann, editors, *Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, pages 300–315. Springer, 2013.

[4] 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.

[5] A. Buchanan, Y. Wang, and S. Butenko. Algorithms for node-weighted Steiner tree and maximum-weight connected subgraph. *Networks*, 72(2):238–248, 2018.

[6] S. Chen, I. Ljubić, and S. Raghavan. The generalized regenerator location problem. *INFORMS Journal on Computing*, 27(2):204–220, 2015.

[7] M. Cherkesly, M. Landete, and G. Laporte. Median and covering location problems with interconnected facilities. *Computers & Operations Research*, 107:1–18, 2019.

[8] M. Conforti, G. Cornuéjols, G. Zambelli, et al. *Integer programming*, volume 271. Springer, 2014.

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

[10] J.-F. Cordeau, F. Furini, and I. Ljubić. Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research*, 275(3):882–896, 2019.

[11] M. S. Daskin and K. L. Maass. The p-median problem. In G. Laporte, S. Nickel, and F. Saldanha da Gama, editors, *Location science*, pages 21–45. Springer, 2015.

[12] E. D. Demaine, M. T. Hajiaghayi, H. Mahini, A. S. Sayedi-Roshkhar, S. Oveisgharan, and M. Zadimoghaddam. Minimizing movement. *ACM Transactions on Algorithms (TALG)*, 5(3):1–30, 2009.

[13] C. Duran-Mateluna, Z. Alès, and S. Elloumi. An efficient benders decomposition for the p-median problem. *European Journal of Operational Research*, 308(1):84–96, 2023.

[14] S. Elloumi. A tighter formulation of the p-median problem. *Journal of combinatorial optimization*, 19(1):69–83, 2010.

[15] E. Fernández and M. Landete. Fixed-charge facility location problems. In G. Laporte, S. Nickel, and F. Saldanha da Gama, editors, *Location science*, pages 47–77. Springer, 2015.

[16] 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. *Mathematical Programming Computation*, 9(2):203–229, 2017.

[17] M. Fischetti, I. Ljubić, and M. Sinnl. Redesigning Benders decomposition for large-scale facility location. *Management Science*, 63(7):2146–2162, 2017.

[18] E. Gaar and M. Sinnl. A scaleable projection-based branch-and-cut algorithm for the p-center problem. *European Journal of Operational Research*, 303(1):78–98, 2022.

[19] S. García, M. Labbé, and A. Marín. Solving large p-median problems with a radius formulation. *INFORMS Journal on Computing*, 23(4):546–556, 2011.

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

[21] N. Gupta and S. Gupta. Approximation algorithms for capacitated facility location problem with penalties. *arXiv preprint arXiv:1408.4944*, 2014.

[22] K. Jakob and P. M. Pruzan. The simple plant location problem: Survey and synthesis. *European Journal of Operational Research*, 12:36–81, 1983.

[23] O. Kariv and S. L. Hakimi. An algorithmic approach to network location problems. ii: The p-medians. *SIAM Journal on Applied Mathematics*, 37(3):539–560, 1979. ISSN 00361399.

[24] M. Leitner, I. Ljubić, J.-J. Salazar-González, and M. Sinnl. An algorithmic framework for the exact solution of tree-star problems. *European Journal of Operational Research*, 261(1):54–66, 2017.

[25] M. Leitner, I. Ljubić, M. Luipersbeck, and M. Sinnl. A dual ascent-based branch-and-bound framework for the prize-collecting Steiner tree and related problems. *INFORMS Journal on Computing*, 30(2):402–420, 2018.

[26] M. Leitner, I. Ljubić, J.-J. Salazar-González, and M. Sinnl. The connected facility location polytope. *Discrete Applied Mathematics*, 234:151–167, 2018.

[27] I. Ljubić. Solving Steiner trees: Recent advances, challenges, and perspectives. *Networks*, 77(2):177–204, 2021.

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

[29] T. L. Magnanti and L. A. Wolsey. Optimal trees. *Handbooks in operations research and management science*, 7:503–615, 1995.

[30] N. Mladenović, J. Brimberg, P. Hansen, and J. A. Moreno-Pérez. The p-median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, 179 (3):927–939, 2007.

[31] D. Rehfeldt and T. Koch. Combining NP-hard reduction techniques and strong heuristics in an exact algorithm for the maximum-weight connected subgraph problem. *SIAM Journal on Optimization*, 29(1):369–398, 2019.

[32] D. Rehfeldt and T. Koch. On the exact solution of prize-collecting Steiner tree problems. *INFORMS Journal on Computing*, 34(2):872–889, 2022. ISSN 1091-9856, 1526-5528.

[33] D. Rehfeldt, T. Koch, and S. J. Maher. Reduction techniques for the prize collecting Steiner tree problem and the maximum-weight connected subgraph problem. *Networks*, 73(2):206–233, 2019.

[34] C. ReVelle, M. Scholssberg, and J. Williams. Solving the maximal covering location problem with heuristic concentration. *Computers & Operations Research*, 35(2):427–435, 2008.

[35] A. Romich, G. Lan, and J. C. Smith. A robust sensor covering and communication problem. *Naval Research Logistics (NRL)*, 62(7):582–594, 2015.

[36] D. Weninger and L. A. Wolsey. Benders-type branch-and-cut algorithms for capacitated facility location with single-sourcing. *European Journal of Operational Research*, 2023. ISSN 0377-2217.

[37] G. Xu and J. Xu. An improved approximation algorithm for uncapacitated facility location problem with penalties. *Journal of combinatorial optimization*, 17(4):424–436, 2009.

Appendix A. Calculation for Example 1

It suffices to open one facility out of $\{2, \ldots, n\}$ to serve all customers J, and facility 1 to keep inter-connectivity, hence:

$$\text{OPT} = L + K + m\varepsilon.$$

Solution of the LP-relaxations of models MPIF SW and MPIF MW

For Example 1, an optimal LP-solution of the MPIF SW model is

$$y_i = 1, i \in \{0, 2\}, y_1 = \frac{1}{n-1}, w_{j2} = 1, j \in J \text{ and } y_i = 0, i = 3, \ldots, n$$

which gives the objective function value equal to

$$v_{LP}(\text{MPIF SW}) = K + \frac{1}{n-1}L + m\varepsilon.$$

On the other hand, an optimal solution of the MPIF MW model is

$$y_0 = 1, y_i = \frac{1}{n-1}, i = 1, \ldots, n, w_{ji} = \frac{1}{n-1}, j \in J, i = 2, \ldots, n$$

with gives the objective function value equal to

$$v_{LP}(\text{MPIF MW}) = K + \frac{1}{n-1}L + m\varepsilon.$$

Hence, for this particular instance we have $v_{LP}(\text{MPIF MW}) = v_{LP}(\text{MPIF SW})$.

Appendix B. Calculation for Example 2

To obtain a feasible solution there must be a path from the root node to facility n, and hence all facilities must be open, so we have:

$$\text{OPT} = nK + \epsilon.$$

For the LP-relaxation of the MPIF MW, we have to open facility n. Due to constraints (10b), we also have to fully open all other facilities, i.e.

$$v_{LP}(\text{MPIF MW}) = nK + \varepsilon.$$

For the LP-relaxation of the MPIF SW, we have to open facility n. Due to constraints (9):

$$
\begin{aligned}
y_n &= 1 \\
f_{n-1\ n} &= 1 \\
y_{n-1} &= \frac{1}{n-1} \\
f_{n-2\ n-1} &= \frac{n}{n-1} \\
&\dots \\
y_2 &= \frac{n^{n-3}}{(n-1)^{n-2}}
\end{aligned}
$$

$$
\sum_{i \in I \setminus \{0\}} y_i = 1 + \frac{1}{n-1} \sum_{m=0}^{n-2} \left(\frac{n}{n-1} \right)^m = \left(\frac{n}{n-1} \right)^{n-2}
$$

$$
v_{LP}(\text{MPIF SW}) = \left(\frac{n}{n-1} \right)^{n-2} K + \varepsilon.
$$