European Journal of Operational Research 303 (2022) 14–36

Contents lists available at ScienceDirect

European Journal of Operational Research

journal homepage: www.elsevier.com/locate/ejor

Discrete Optimization

Upgrading edges in the maximal covering location problem

Marta Baldomero-Naranjo [a,*], Jörg Kalcsics [b], Alfredo Marín [c], Antonio M. Rodríguez-Chía [a]

[a] Departamento de Estadística e Investigación Operativa, Universidad de Cádiz, Facultad de Ciencias, Puerto Real (Cádiz), 11510, Spain
[b] School of Mathematics, University of Edinburgh, Edinburgh, EH9 3FD, United Kingdom
[c] Departamento de Estadística e Investigación Operativa, Universidad de Murcia, Facultad de Matemáticas, Murcia, 30100, Spain

ARTICLE INFO

Article history:
Received 11 June 2021
Accepted 1 February 2022
Available online 5 February 2022

Keywords:
Location
Covering problems
Networks
Upgrading problems
Integer programming

ABSTRACT

We study the upgrading version of the maximal covering location problem with edge length modifications on networks. This problem aims at locating p facilities on the vertices (of the network) so as to maximise coverage, considering that the length of the edges can be reduced at a cost, subject to a given budget. Hence, we have to decide on: the optimal location of p facilities and the optimal edge length reductions.

This problem is NP-hard on general graphs. To solve it, we propose three different mixed-integer formulations and a preprocessing phase for fixing variables and removing some of the constraints. Moreover, we strengthen the proposed formulations including valid inequalities. Finally, we compare the three formulations and their corresponding improvements by testing their performance over different datasets.

1. Introduction

The maximal covering location problem was first introduced by Church & ReVelle (1974). Given a set of clients, each with their own demand, the aim is to locate a fixed number of facilities so as to maximise the amount of covered demand. A client is hereby considered to be covered if their distance to a facility is smaller than or equal to a given coverage radius. Since its origins, this model has been widely studied in the literature under different perspectives. One of the most distinguishing aspects is the solution domain of the problem: continuous (Bansal & Kianfar, 2017; Church, 1984; Plastria, 2002), discrete (Avella, Boccia, & Vasilyev, 2009; Church & ReVelle, 1974; Cordeau, Furini, & Ljubić, 2019; García & Marín, 2019), or on networks (Berman, Kalcsics, & Krass, 2016; Church & Meadows, 1979; Fröhlich, Maier, & Hamacher, 2020). Furthermore, the maximal covering location problem has been solved dealing with alternative coverage assumptions, like gradual coverage (Berman & Krass, 2002) and cooperative coverage (Averbakh, Berman, Krass, Kalcsics, & Nickel, 2014; Karatas & Eriskin, 2021), and with uncertainty, for example uncertainty in the customer demand (Baldomero-Naranjo, Kalcsics, & Rodríguez-Chía, 2021; Berman & Wang, 2011), in the availability of facilities to provide coverage (Daskin, 1983; Marín, Martínez-Merino, Rodríguez-Chía, & da Gama, 2018; Vatsa & Jayaswal, 2021), or in a combination of these and other parameters (Arana-Jiménez, Blanco, & Fernández, 2020; Guzmán, Masegosa, Pelta, & Verdegay, 2016; Zhang, Peng, & Li, 2017).

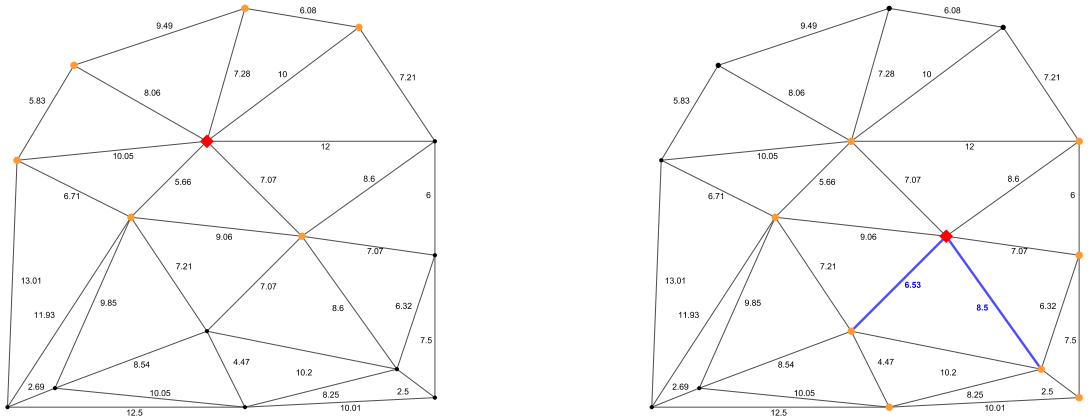
Common to all those problems is, however, that the parameters of the network and the problem are not decision variables of the model. In this work, we propose a different approach dealing with the maximal covering location problem on networks assuming that edges can be *upgraded* and the total cost of all upgrades is subject to a budget constraint. Upgrading an edge hereby means reducing its length, usually within certain limits, at a given cost which is proportional to the extent of the upgrade. In what follows, we give an example to illustrate the proposed problem.

Example 1. Consider the single facility upgrading version of the maximal covering location problem in the graph depicted in Fig. 1a. The numbers next to the edges are their lengths, the coverage radius is 11, all the nodes have the same demand, all the edges have a reduction cost of 1 unit per unit, and the maximum reduction is 25% of the edge length. The problem has been solved for different values of the budget: 0 (Fig. 1a), 2.5 (Fig. 1b), 5 (Fig. 1c), and 10 (Fig. 1d). The facility is represented as a red diamond, the covered nodes are colored in orange, and the upgraded edges are shown as thicker blue edges.

The pictures show that the location of the facility changes when the budget grows, covering more demand with each increase. In

* Corresponding author.
E-mail addresses: marta.baldomero@uca.es (M. Baldomero-Naranjo), joerg.kalcsics@ed.ac.uk (J. Kalcsics), amarin@um.es (A. Marín), antonio.rodriguezchia@uca.es (A.M. Rodríguez-Chía).

https://doi.org/10.1016/j.ejor.2022.02.001



(a) Without upgrading (budget=0)

(b) Small budget (budget=2.5)

(c) Medium budget (budget=5)

(d) Large budget (budget=10)

Fig. 1. Illustrative example.

fact, if we fix the optimal location of the problem without upgrading and apply the most beneficial upgrades (i.e., the location of the facility is given and only the upgrade of each edge should be determined), we get that this facility would cover one node less for each of the other three cases. These results illustrate the usefulness of the upgrading version of this problem.

1.1. Related work

There are three main types of problems in the literature in which two key parameters of the network, demand weights and edge lengths, can be adjusted, i.e., they are decision variables of the model:

- In *inverse problems*, the objective is to modify one of the two key parameters at minimum cost such that a given feasible solution becomes optimal, see e.g. Heuberger (2004), Burkard, Pleschiutschnig, & Zhang (2004b), Bonab, Burkard, & Gassner (2011), Wu, Lee, Zhang, & Wang (2013), Alizadeh & Etemad (2016), Yang & Zhang (2008), Nguyen & Sepasian (2016), Gassner (2012).
- In *reverse problems*, the goal is to maximise/minimise the objective value of a given solution by modifying one of the key parameters subject to a given budget. Basically, in reverse problems the roles of variables and input parameters are interchanged, i.e., the variables are considered as parameters and the parameters become variables, see e.g. Burkard, Gassner, & Hatzl (2006, 2008), Wang & Bai (2010), Zhang, Yang, & Cai (1999).
- In *up/downgrading problems*, an actor modifies the parameters of the network and then a reactor takes a decision. In upgrading problems, actor and reactor have the same goal; in downgrading problems, their objectives are conflicting.

Summing up, the main difference between inverse (reverse) problems and up/downgrading problems is that in the former there is a given solution that we want to improve, while in the latter there is not. In this paper, we will focus on the upgrading maximal covering location problem with variable edge lengths. In the following, we denote problems where the edge lengths (demand weights) can be changed as *edge upgrading* (*node upgrading*) problems.

Next, we briefly review the literature of upgrading problems. The upgrading version of many classical problems has been studied during the last decades, e.g. for the spanning tree problem (Álvarez-Miranda & Sinnl, 2017), for the min-max spanning tree problem (Sepasian & Monabbati, 2017), for bottleneck problems (Burkard, Lin, & Zhang, 2004a), for minimum flow cost problems (Demgensky, Noltemeier, & Wirth, 2002), for the shortest path problem (Dilkina, Lai, & Gomes, 2011), for the maximal shortest path interdiction problem (Zhang, Guan, & Pardalos, 2021), or for communication and signal flow problems (Paik & Sahni, 1995).

Table 1
Summary of literature review of upgrading problems.

Field	Upgrading	N	C	Problem and reference
Location problems	Nodes	X		1-center: Gassner (2009).
		X		Hub-location: Blanco & Marín (2019).
		X		1-median: Gassner (2007).
			X	Euclidean 1-median: Plastria (2016).
		X		p-median: Sepasian & Rahbarnia (2015).
	Arcs/edges	X		1-center: Sepasian (2018).
		X		Maximal covering: our paper.
		X		Obnoxious p-median: Afrashteh et al. (2020).
Others	Nodes	X		Communication and signal flow problems: Paik & Sahni (1995).
		X		Shortest path: Dilkina et al. (2011).
		X		Spanning tree: Álvarez-Miranda & Sinnl (2017).
	Arcs/edges	X		Maximal shortest path interdiction problem: Zhang et al. (2021).
		X		Min-max spanning tree: Sepasian & Monabbati (2017).
		X		Minimum flow cost: Demgensky et al. (2002).

In the context of node upgrading location problems (the weight of the vertices can be modified subject to a prespecified budget), the following problems have been analysed: the 1-median problem (Gassner, 2007), the 1-center problem (Gassner, 2009), the Euclidean 1-median problem (Plastria, 2016), the p-median problem (Sepasian & Rahbarnia, 2015), and the hub-location problem (Blanco & Marín, 2019), among others.

In the context of edge upgrading location problems, we are aware of only two directly related publications: upgrading the 1-center problem (Sepasian, 2018) and upgrading the obnoxious p-median problem on trees (Afrashteh, Alizadeh, & Baroughi, 2020). Somewhat related are the models in Melkote & Daskin (2001a,b) which consider the possibility of adding new edges to the network. This can be interpreted as an edge upgrading problem where an edge is in one of two states: non-upgraded with a length of infinity, and upgraded with a finite length. This, however, differs from typical upgrading models where an upgrade can be any fraction of the edge length (or node weight). The models consider the minimization of the overall cost. So far, no results are known for covering problems. Therefore, the main aim of this paper is to fill this gap in the literature by studying the upgrading maximal covering location problem with variable edge lengths.

For sake of clarity, we summarise the cited literature of upgrading problems in Table 1. We add two columns labelled N and C for network and continuous problems.

1.2. Applications

This problem has several interesting applications in real-life. Note that two decisions are made at the same time. On the one hand, decide where to locate the p facilities, and on the other hand, determine which edges to upgrade and by how much.

One application of this problem arises when a public administration wants to improve the accessibility of public services for citizens, e.g. for health centres, educational facilities or social welfare facilities. As the improvement is closely linked with distances (Ensor & Cooper, 2004), one way to achieve this is to invest in the infrastructure in order to reduce travel times to those services. Such an investment is often a combination of building new facilities and improving the means to get to them, for example by upgrading roads (developing a road into a highway, adding new lanes, etc.) and enhancing public transport (incorporating high-speed lines, adding dedicated bus lines, increasing the frequency of service along links, etc.).

An interesting application in the private sector is for telecommunication companies. To improve their transmission rates and broadband coverage, they will have to increase the bandwidth on existing network links as well as build new or extend existing switching centers. Similar problems are faced by gas and electricity companies who wish to increase their coverage.

Finally, we would like to highlight another useful application in shopping centers, airports, etc. The aim is to locate services such as defibrillators and information posts, in combination with building additional passenger conveyors or escalators to make sure that as many people as possible are within a fixed walking distance of these facilities.

1.3. Overview

In this work, we derive three mixed-integer linear programming formulations for the maximal covering location problem with edge upgrades. Furthermore, we develop an effective preprocessing phase that allows us to reduce the dimension of the proposed formulations, allowing us to solve instances faster and also solve larger instances than without preprocessing. Besides, we include several sets of valid inequalities in order to eliminate symmetries and even further improve the solution times of the formulations.

The rest of the paper is structured as follows. In Section 2 the problem is introduced. Section 3 presents the first formulation for the problem based on flow variables. Moreover, a preprocessing phase and valid inequalities are developed. Next, in Section 4 and Section 5 two new formulations are proposed. In addition, several valid inequalities to enhance them are presented. Section 6 contains computational experiments in which we compare the three formulations. We also test the efficiency of the developed valid inequalities. Finally, our conclusions and some future research topics are included in Section 7.

2. Definitions and problem description

Let $N = (V, E, \ell)$ be an undirected network with node set $V = \{1, \dots, n\}$ and edge set E, where $|E| = m$. Every edge $e = [k, q] = [q, k] \in E$, $k, q \in V$, has a positive length $\ell_e = \ell_{[k,q]}$ and is assumed to be rectifiable. For $i, j \in V$, $d(i, j)$ is the length of the shortest path connecting i with j. Furthermore, we are given a fixed coverage radius $R > 0$. We say that a node $i \in V$ is *covered* by a facility at node j if $d(i, j) \leq R$. Finally, for each node $i \in V$ we are given a non-negative amount w_i that specifies the demand at the node.

The length ℓ_e of each edge $e \in E$ can be reduced by an amount lower than or equal to $u_e \in [0, \ell_e)$, $e \in E$. Without loss of generality, we assume that $\ell_e - u_e \leq R$, for $e \in E$ (if that were not the case, i.e., there were an edge $e \in E$ such that $\ell_e - u_e > R$, then e can be removed from the network without affecting the optimal solution). Moreover, any unit of reduction of the length of the edge e

Table 2
Notation used in the paper.

A	Set of all arcs in the induced directed network.
B	Budget.
c_e	Unit cost of reducing the length of edge e, $e \in E$.
$d(i, j)$	Distance between nodes i and j before upgrading, $i, j \in V$.
$d(i, j, \delta)$	Distance between nodes i and j after the edge length reductions δ_e, $e \in E$. In particular, $d(i, j, \delta^{ij})$ represents the distance after the most favourable feasible edge length reductions in the path from i to j and $d(i, j, u)$ the distance in a network with edge lengths $\ell_e - u_e$, $e \in E$.
Γ_i	Set of edges incident to node i for each $i \in V$.
Γ_i^+ (Γ_i^-)	Set of outgoing (incoming) arcs for each $i \in V$.
m	Number of edges.
n	Number of nodes.
$N = (V, E, \ell)$	Network with node set V, edge set E, where $e \in E$ has length ℓ_e.
p	Number of facilities.
R	Coverage radius.
u_e	Maximum amount that edge e can be reduced, $e \in E$.
$\hat{V}_i$	Set of nodes whose distance to i before upgrading is lower than or equal to R, i.e., $\{j \in V \setminus \{i\} : d(i, j) \leq R\}$.
w_i	Demand of node i, for $i \in V$.

comes at a cost of c_e and there is a budget constraint B on the overall cost of reduction. Again without loss of generality, we assume that $c_e u_e \leq B$, for $e \in E$ (if that were not the case, i.e., there were a cost c_e for $e \in E$ such that $c_e u_e > B$, then u_e can be substituted by $u_e = B/c_e$ without affecting the optimal solution). Finally, we assume that facilities can only be located at nodes. The upgrading maximal covering location problem (Up-MCLP) aims to locate p service facilities covering the maximum demand taking into account that the total cost for the edge length reductions is within the given budget.

Let $\delta = (\delta_e)_{e \in E}$ denote a vector of edge length reductions, $0 \leq \delta_e \leq u_e$, for $e \in E$. Moreover, let $d(i, j, \delta)$ be the length of a shortest path between nodes i and j after the edge length reductions δ have been applied, i.e., a shortest path in the network $(V, E, \ell(\delta))$ where $\ell_e(\delta) = \ell_e - \delta_e$, for $e \in E$. Finally, for $p \in \mathbb{N}$ let $X_p \subseteq V$ denote a set of p nodes and let $C(X_p, \delta) = \{i \in V \mid \exists j \in X_p : d(i, j, \delta) \leq R\}$ denote the set of all nodes covered by a facility in X_p after the edge upgrades. Then, Up-MCLP can be formulated as:

$$\max \left\{ \sum_{i \in C(X_p, \delta)} w_i \;\middle|\; \sum_{e \in E} c_e \delta_e \leq B, X_p \subseteq V, |X_p| = p, 0 \leq \delta_e \leq u_e, e \in E \right\}.$$

Table 2 summarizes the notation used in this paper.

Observe that this problem is NP-hard because the maximal covering location problem (MCLP) is a particular case of Up-MCLP (setting $u_e = 0$ for all $e \in E$). The NP-hardness of the maximal covering location problem is proved in Hochbaum (1997).

3. Flow coverage formulation

In this section, we propose the first of our three Mixed-Integer Programming (MIP) formulations for Up-MCLP. Using flow variables, the idea of this formulation is to model a path between those pairs of nodes for which the distance between them is smaller than or equal to R after the edge length reductions have been applied. That is, if $d(i, j, \delta) \leq R$, then this will be reflected in the formulation by a unit flow between nodes i and j. If, however, $d(i, j, \delta) > R$, then the flow between i and j will be zero. We note that in the former case, any path of length $\leq R$ will do to assert coverage of i (j) by a service facility located at site j (i), so we do not insist on finding the shortest path.

To facilitate the use of flow variables, we consider a directed network $N_D = (V, A, \ell)$ with node set $V = \{1, \ldots, n\}$ and arc set A containing arcs (i, j) and (j, i) for each edge $[i, j] \in E$. We denote $e_a \in E$ the undirected edge corresponding to $a \in A$ and we define Γ_i^+ (Γ_i^-) as the set of outgoing (incoming) arcs for each $i \in V$. The set of variables used in the formulation is summarized below.

Decision variables

x_j	1, if there is a facility at node j, and 0, otherwise, for $j \in V$.
y_{ij}	1, if node i is assigned to a facility at node j, and 0, otherwise, for $i, j \in V, i \neq j$.
δ_e	The amount of reduction of the length of edge e, for $e \in E$.
f_a^{ij}	1, if a path of length $\leq R$ from i to j traverses arc a, and 0, otherwise, for $i, j \in V, i < j$, $a \in A \setminus \left(\Gamma_i^- \cup \Gamma_j^+\right)$.
α_a^{ij}	The length of arc a, if this arc belongs to a path of length $\leq R$ from node i to node j ($\alpha_a^{ij} = 0$ otherwise), for $i, j \in V, i < j$, $a \in A \setminus \left(\Gamma_i^- \cup \Gamma_j^+\right)$.

Observe that in the definition of the y-variables, we use the term "assign to" instead of "covered by". A node can potentially be covered by more than one service facility and we decided to resolve this ambiguity by explicitly assigning a node to a facility as this simplifies the explanations of the formulations. Taking into account the notation presented above, the flow coverage formulation for Up-MCLP is:

$$\text{(Flow-Cov)} \quad \max \sum_{i \in V} w_i \left(\sum_{j \in V \setminus \{i\}} y_{ij} + x_i \right)$$

$$\text{s.t.} \quad \sum_{j \in V} x_j = p, \tag{1}$$

$$\sum_{j \in V \setminus \{i\}} y_{ij} + x_i \leq 1, \qquad i \in V, \tag{2}$$

$$y_{ij} \leq x_j, \qquad i, j \in V, i \neq j, \tag{3}$$

$$\sum_{e \in E} c_e \delta_e \leq B, \tag{4}$$

$$0 \leq \delta_e \leq u_e, \qquad e \in E, \tag{5}$$

$$\sum_{a \in A \setminus \left(\Gamma_i^- \cup \Gamma_j^+\right)} \alpha_a^{ij} \leq R, \qquad i, j \in V, i < j, \tag{6}$$

$$\alpha_a^{ij} \geq f_a^{ij} \ell_{e_a} - \delta_{e_a}, \qquad i, j \in V, i < j, a \in A \setminus \left(\Gamma_i^- \cup \Gamma_j^+\right), \tag{7}$$

$$\sum_{a \in \Gamma_k^+, a \notin \Gamma_i^-} f_a^{ij} - \sum_{a \in \Gamma_k^-, a \notin \Gamma_j^+} f_a^{ij} = 0, \qquad i, j \in V, i < j, k \in V \setminus \{i, j\}, \tag{8}$$

$$\sum_{a \in \Gamma_i^+} f_a^{ij} = y_{ij} + y_{ji}, \qquad i, j \in V, i < j, \tag{9}$$

$$\sum_{a \in \Gamma_j^-} f_a^{ij} = y_{ij} + y_{ji}, \qquad i, j \in V, i < j, \tag{10}$$

$$0 \leq \alpha_a^{ij} \leq \ell_{e_a} - \delta_{e_a}, \qquad i, j \in V, i < j, a \in A \setminus \left(\Gamma_i^- \cup \Gamma_j^+\right), \tag{11}$$

$$x_j \in \{0, 1\}, \qquad j \in V, \tag{12}$$

$$y_{ij} \in \{0, 1\}, \qquad i, j \in V, i \neq j, \tag{13}$$

$$f_a^{ij} \in \{0, 1\}, \qquad i, j \in V, i < j, a \in A \setminus \left(\Gamma_i^- \cup \Gamma_j^+\right). \tag{14}$$

The objective of the problem is to maximise the amount of covered demand. Constraint (1) fixes the number of located facilities. The family of constraints (2) guarantees that either node i is itself a service facility or is assigned to at most one node. The family of constraints (3) ensures that a node is assigned to an open facility. The families of constraints (4) and (5) force that the reduction on the length of the edges in the network is feasible. The families of constraints (6)–(10) ensure that if y_{ij} (y_{ji}) takes value one, there exists a path shorter than or equal to R from i to j (from j to i). Indeed, if $y_{ij} + y_{ji} = 1$, then the flow balance constraints (8)–(10) aim at building a path from i to j and consequently from j to i. Furthermore, constraints (6)-(7) ensure that this path is shorter than

or equal to R. Note that constraints (2) and (3) imply $y_{ij} + y_{ji} \leq 1$, for $i, j \in V, i \neq j$.

Next, we introduce a result that proves that the integrality condition of some families of variables of (Flow-Cov) can be relaxed.

Lemma 1. *An equivalent formulation of (Flow-Cov) is obtained substituting the set of constraints (12) by:*

$$0 \leq x_j \leq 1, \quad j \in V, \tag{15}$$

and the set of constraints (13) by:

$$0 \leq y_{ij} \leq 1, \quad i, j \in V, i \neq j. \tag{16}$$

Proof. Concerning the first part of the lemma, the x-variables inherit the integrality condition from y-variables due to constraints (2) and (3).

Let x^* and y^* be optimal values for the x- and y-variables, respectively, of formulation (Flow-Cov) when constraints (12) are substituted by (15). For any $i \in V$ such that $\sum_{j \in V \setminus \{i\}} y^*_{ij} = 1$, constraint (2) ensures that $x^*_i = 0$. On the other hand, for any $i \in V$, such that there exists $j_0 \in V$, $j_0 \neq i$, with $y^*_{j_0 i} = 1$, constraints (3) guarantee that $x^*_i = 1$. Finally, the model will choose to locate the remaining service facilities (up to a total of p) at the uncovered nodes with the largest demand.

Regarding the second part of the lemma, following a similar argument than before, we conclude that the y-variables inherit the integrality condition from the f-variables due to constraints (9) and (10) and the condition that $y_{ij} + y_{ji} \leq 1$ (derived by constraints (2) and (3)). □

Observe that even though the f-variables are the intuitive candidates for relaxation (since their number is much larger than the number of x- and y-variables), there are examples where the optimal value differs when the integrality condition of these variables is relaxed.

An alternative formulation for Up-MCLP can be derived from the formulation (Flow-Cov) by replacing constraints (6), (7), and (11) with the following ones:

$$\sum_{a \in A \setminus (\Gamma_i^- \cup \Gamma_j^+)} \left(f_a^{ij} \ell_{e_a} - \gamma_a^{ij} \right) \leq R, \quad i, j \in V, i < j, \tag{17}$$

$$\gamma_a^{ij} \leq u_{e_a} f_a^{ij}, \quad i, j \in V, i < j, a \in A \setminus \left(\Gamma_i^- \cup \Gamma_j^+ \right), \tag{18}$$

$$0 \leq \gamma_a^{ij} \leq \delta_{e_a}, \quad i, j \in V, i < j, a \in A \setminus \left(\Gamma_i^- \cup \Gamma_j^+ \right), \tag{19}$$

where γ_a^{ij} represents the reduction on the length of arc a if this arc belongs to a path of length $\leq R$ from node i to node j, for $i, j \in V, i < j$, $a \in A \setminus \left(\Gamma_i^- \cup \Gamma_j^+ \right)$. A preliminary computational analysis showed that the alternative formulation (Flow-Cov) for Up-MCLP where constraints (6), (7), and (11) are replaced with (17), (18), and (19) is better than the original (Flow-Cov). In this analysis, which was carried out using the data described in Section 6.1, we compared the formulations based on the number of instances solved to optimality within the time limit, the time employed to obtain the optimal solutions, the MIP relative gaps, the best solution gaps, and the linear relaxation gaps. More details about these performance values can be found in Section 6.

3.1. Preprocessing phase

Next, we present two results for preprocessing the model which reduce the number of constraints and variables of the above formulation and, subsequently, shorten the computational time required to solve them to optimality. The idea of the first is that if the distance between node i and node j is smaller than or equal to R even before modifying the edge lengths of the network, then node i can always be covered by a facility located at node j (and vice versa) regardless of the edge length reductions made in the network.

Proposition 1. *If $d(i, j) \leq R$, for $i, j \in V, i < j$ then it is not necessary to include either the f_a^{ij}-variables or the α_a^{ij}-variables (γ_a^{ij}-variables) in formulation (Flow-Cov). Moreover, we can remove the constraints associated with these variables from the family of constraints (6)–(11) and (14) in the original (Flow-Cov) formulation and (8)–(10), (14), (17)–(19) in the alternative (Flow-Cov) formulation.*

The second result analyses the opposite case, i.e., Proposition 2 considers the situation in which the distance between node i and node j is greater than R independently of the edge length reductions.

Proposition 2. *If one of the following four conditions is fulfilled for $i, j \in V$, $i < j$, the variables $y_{ij}, y_{ji}, f_a^{ij}, \alpha_a^{ij} (\gamma_a^{ij})$ for $a \in A$ can be removed from the (Flow-Cov) formulation. Moreover, the constraints associated with this pair of nodes can be deleted, in particular (2), (3), (6)-(11), (13), (14) in the original (Flow-Cov) formulation and (2), (3), (8)–(10), (13), (14), (17)–(19) in the alternative (Flow-Cov) formulation.*

i) $d(i, j) > R + \sum_{e \in E} u_e$, *for* $i, j \in V, i < j$.

ii) $d(i, j, u) > R$, *for* $i, j \in V, i < j$, *where* $d(i, j, u)$ *is the length of the shortest path from* i *to* j *in a graph with edge lengths* $\ell_e - u_e$, *for* $e \in E$.

iii) $$d(i, j) > R + \sum_{k=1}^{\bar{k}} u_{e_{\sigma(k)}} + \frac{B - \sum_{k=1}^{\bar{k}} u_{e_{\sigma(k)}}}{c_{e_{\sigma(\bar{k}+1)}}} \quad \text{for} \quad i, j \in V, i < j,$$
where $\bar{k}$ *is the largest index* k *that satisfies the following condition:*
$$\sum_{h=1}^{k} u_{e_{\sigma(h)}} c_{e_{\sigma(h)}} \leq B, \tag{20}$$
and $\sigma(\cdot)$ *is a permutation of* $\{1, \ldots, m\}$ *that sorts the unit upgrade costs in non-decreasing order.*

iv) *The optimal value of the following problem is greater than* R, *for* $i, j \in V, i < j$,

$$\begin{aligned}
\left(P_{d(i,j,\delta^{ij})}\right) \quad \min \quad & \sum_{a \in A} (f_a \ell_{e_a} - \gamma_a) \\
\text{s.t.} \quad & (4), (5), \\
& \sum_{a \in \Gamma_k^+} f_a - \sum_{a \in \Gamma_k^-} f_a = g_k, \quad k \in V, && (21) \\
& \gamma_a \leq u_{e_a} f_a, \quad a \in A, && (22) \\
& \gamma_a \leq \delta_{e_a}, \quad a \in A, && (23) \\
& f_a \in \{0, 1\}, \quad a \in A, && (24)
\end{aligned}$$

where the f-variables and γ-variables are defined as above (we dropped the indices i and j for the ease of exposition), and

$$g_k = \begin{cases} 1, & \text{if } k = i, \\ -1, & \text{if } k = j, \\ 0, & \text{otherwise.} \end{cases}$$

Proof. Each of the items of the proposition is proven below.

i) The first condition considers the case where the distance from i to j is greater than R even when reducing the length of every edge by the maximum amount allowed. Therefore, it is straightforward to conclude that the distance between the two nodes cannot be less than or equal to R.

ii) In the previous condition, the maximum amount of reduction in the whole network was considered without taking into account the edges for which this reduction is made. Now we compute the shortest path between two nodes in the network assuming an unlimited budget, i.e., the full discount is applied to all edges. For each edge e, let $\ell_{e_u} = \ell_e - u_e$. For $i, j \in V$, let $d(i, j, u)$ be the length of the shortest path connecting i with j where the length of the edges are ℓ_{e_u}, for $e \in E$. Hence, even if reducing the maximum amount allowed on all edges, the distance is greater than R, clearly, node i cannot be assigned to node j, and vice versa. Although condition *i)* is weaker than *ii)*, *i)* can be checked more efficiently.

iii) This condition is similar to the first one, but takes into account the budget constraint (4). In this case, we calculate the maximum reduction in the network allowed by the budget. For this purpose, we sort the upgrade costs c_e, for $e \in E$, in non-decreasing order. Let σ be a permutation of $\{1, \ldots, m\}$ such that $c_{e_{\sigma(1)}} \leq c_{e_{\sigma(2)}} \leq \ldots \leq c_{e_{\sigma(m)}}$. Then, we compute the maximum total length reduction over the network, i.e., we spend the budget on upgrading the cheapest edges. Let $\bar{k}$ be the largest index k that satisfies condition (20). Therefore, the right-hand side of *iii)* minus R is the maximum length reduction between any two nodes of the network. Taking into account the above arguments, we conclude that node i cannot be assigned to a facility located at node j, and vice versa.

iv) Condition *iii)* provides the maximal reduction without taking into account whether this reduction can be achieved in a path form i to j. For this reason, that bound can be tightened, but it requires to solve a separate problem for each pair of vertices. Formulation $\left(P_{d(i,j,\delta^{ij})}\right)$ computes the shortest path between node i and node j assuming that all the budget can be spent just for the path between those two nodes. Therefore, the optimal value of this problem, named $d(i, j, \delta^{ij})$, is the minimal distance between node i and node j after the most favourable edges length reductions. Hence, if $d(i, j, \delta^{ij})$ is greater than R, node i can never be assigned to a facility at node j, and vice versa. □

As stated in Demgensky et al. (2002), the shortest path problem where the length of the edges can be reduced, $\left(P_{d(i,j,\delta^{ij})}\right)$, is NP-hard. However, the optimal value of the LP relaxation of formulation $\left(P_{d(i,j,\delta^{ij})}\right)$ provides a valid bound that can still be used instead, albeit yielding a weaker condition. If this value is greater than R, then i can never cover j, and vice versa.

3.2. Valid inequalities

In the previous subsection, we have presented two results to preprocess the model, reducing the number of constraints and variables. In this one, we propose several families of valid inequalities to strengthen the (Flow-Cov) formulation which help us to further shorten the computational times.

Proposition 3. *Let $\hat{V}_i := \{j \in V \setminus \{i\} : d(i, j) \leq R\}$. The following families of constraints are valid inequalities for (Flow-Cov):*

$$f^{ij}_{(k,q)} + f^{ij}_{(q,k)} \leq 1, \qquad [k, q] \in E, i, j \in V, i < j, k, q \neq i, j, \tag{25}$$

$$y_{kj} \geq y_{ij} + f^{ij}_a - 1, \quad i, j, k \in V, i < j, k \neq i, k \neq j, a \in \Gamma^-_k, \tag{26}$$

$$y_{ki} \geq y_{ji} + f^{ij}_a - 1, \quad i, j, k \in V, i < j, k \neq i, k \neq j, a \in \Gamma^-_k, \tag{27}$$

$$x_j \leq \sum_{k:k\neq i, d(i,k)\leq d(i,j)} y_{ik} + x_i, \qquad i \in V, j \in \hat{V}_i, \tag{28}$$

$$x_j \leq \sum_{k:k\neq i, d(i,k,\delta^{ik})\leq d(i,j)} y_{ik} + x_i, \qquad i \in V, j \in \hat{V}_i, \tag{29}$$

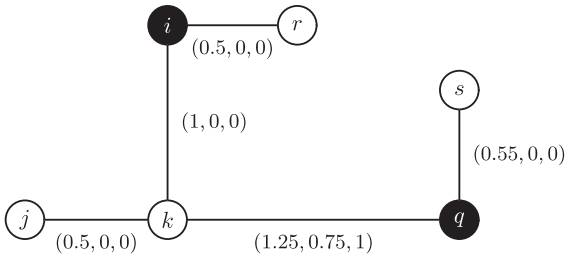


Fig. 2. Illustration of incompatibility.

Proof. The proof of valid inequalities is given below:

The first family of constraints (25) ensure that an edge is not traversed in both directions on a path from i to j, $i < j$.

The second and the third families of constraints, (26) and (27), are based on the fact that a path between two non-adjacent nodes i and j will traverse at least one other node. Therefore, if there exists a path whose length is less than or equal to R that connects a facility at node j (i) with demand point i (j) and traverses node k, then node k will also be assigned to facility j (i). More concretely, given $i, j \in V, i<j$, if $f^{ij}_a = 1$ for some $a \in A$, such that $a \in \Gamma^-_k$, $k \neq i, k \neq j$ and $y_{ij} = 1$ ($y_{ji} = 1$), then the constraints impose that $y_{kj} = 1$ ($y_{ki} = 1$).

Regarding (28), these constraints ensure that a node will be served by the closest service facility that is within the covering distance before upgrading the network (whenever at least one service facility is closer than the coverage radius before upgrading the network). Observe that these constraints eliminate symmetries and are valid also for formulation (Flow-Cov) because nodes might not be assigned to the closest service facility in the upgraded network. Nevertheless, the situation would be incompatible with constraints (26) and (27), as explained in the following remark (Remark 1).

Finally, whenever at least one service facility j is closer to a node i than the coverage radius before upgrading the network, constraints (29) ensure that this node will either host a facility itself or be assigned to this service facility or to a facility that can be closer after upgrading the network (it considers the distances in the range of $d(i, j)$ and the most favourable edge length reductions, i.e., $d(i, k, \delta^{ik})$ for any $k \neq i$.). □

Remark 1.

i) Constraints (26) and (27) might be incompatible with (28), i.e., constraints (26)–(28) cannot be included in the formulation simultaneously.

ii) The family of valid inequalities (28) is tighter than (29), but (29) are not incompatible with (26) and (27).

In the following, we present an example illustrating the first part of Remark 1.

Example 2. Consider the network depicted in Fig. 2. For each edge, its length, its upper bound of reduction, and its cost per unit of reduction, (ℓ_e, u_e, c_e), are printed next to the edge. Let $R = 1$, $p = 2$, $B = 0.75$, and the demand of nodes $w_i = 1$, $w_j = 1$, $w_k = 1$, $w_q = 1$, $w_r = 1000$, $w_s = 1000$. It is straightforward to conclude that the optimal location of the services are the dark nodes, i.e., $x^*_i = 1$ and $x^*_q = 1$, and that the optimal edge length reduction is $\delta^*_{[k,q]} = 0.75$.

In this case, from constraints (28) we obtain that $x_i \leq y_{ki} + y_{kj} + x_k$. Then, $y^*_{ki} = 1$. On the other hand, facility q is the only one that covers node j, then $y^*_{jq} = 1$. Moreover, as the path from node j to node q traverses node k, we obtain that $f^{jq*}_{(j,k)} = 1$. Therefore, from constraint (26), we obtain that $y^*_{kq} = 1$. Thus, we have found that these families of constraints are incompatible (y_{ki} and y_{kq} can not take value one simultaneously due to constraints (2)).

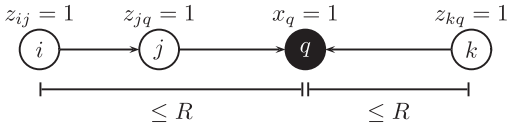


Fig. 3. Illustration of z-variables.

Note that the ideas behind constraints (28) and (29) are practically identical. The reason why constraints (29) are not incompatible with (26) and (27) is that the constraints (29) do not force that the node is assigned to the closest service facility before upgrading, instead for given i, j such that $i \neq j$ and $d(i, j) \leq R$, it enables the node to be assigned to another node whose distance after the most favourable edge length reductions is smaller than or equal to $d(i, j)$. In Espejo, Marín, & Rodríguez-Chía (2012) a detailed description of closest service assignment constraints is given.

Observe that the variables dropped from the formulation in the preprocessing phase (Propositions 1 and 2), can also be removed from the valid inequalities presented in this subsection. In the next section, an alternative formulation for this problem is developed.

4. Path formulation

In this section we present our second formulation for Up-MCLP. It contains fewer variables and constraints than (Flow-Cov). However, this comes at the expense of reducing the scope of preprocessing the model.

This formulation again models paths of length at most R from a customer node i to a service provider. However, in contrast to (Flow-Cov), the path from i is not modelled as a flow but through the immediate successor of i on a path of length $\leq R$ to a facility. For this purpose, we introduce two new binary variables z_{ij} (z_{ji}) for $[i, j] \in E$, such that z_{ij} (z_{ji}) is equal to one if node j (i) is the next node on a path of length at most R from i (j) to a service facility. In Fig. 3 we illustrate this family of variables where the dark node represents a facility. If i is covered by a facility at q, then also j must be covered. Note that a feasible solution resembles a forest rooted at the facilities.

For the sake of clarity, a description of the decision variables used in the formulation is given next.

Decision variables

x_j	1, if there is a facility at node j, and 0, otherwise, for $j \in V$.
z_{ij}	1, if node j is the next node on a path of length $\leq R$ from i to a facility, and 0, otherwise, for $[i, j] \in E$.
d_i	An upper bound of the length of the built path from node i to its assigned service facility, for $i \in V$.
$\delta_e = \delta_{[i,j]}$	The amount of reduction of the length of edge $e = [i, j]$, for $e \in E$.

The formulation for problem Up-MCLP using these variables, (Path), is as follows:

$$(Path) \quad \max \quad \sum_{i \in V} w_i \left(x_i + \sum_{j:[i,j] \in E} z_{ij} \right)$$

$$\text{s.t.} \quad (1), (4), (12),$$

$$\sum_{j:[i,j] \in E} z_{ij} + x_i \leq 1, \qquad i \in V, \tag{30}$$

$$\sum_{j:[i,j] \in E, j \neq k} z_{ij} + x_i \geq z_{ki}, \qquad [k, i] \in E, \tag{31}$$

$$0 \leq d_i \leq R \sum_{j:[i,j] \in E} z_{ij} \qquad i \in V, \tag{32}$$

$$d_i \geq d_j + \ell_{[i,j]} z_{ij} - \delta_{[i,j]} - R(1 - z_{ij}), \qquad [i, j] \in E, \tag{33}$$

$$0 \leq \delta_e \leq u_e (z_{ij} + z_{ji}), \qquad e = [i, j] \in E, \tag{34}$$

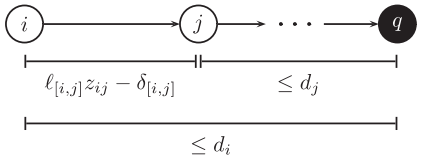


Fig. 4. Illustration of constraints (33).

$$z_{ij} \in \{0, 1\}, \qquad [i, j] \in E. \tag{35}$$

The family of constraints (30) states that each node is assigned to at most one facility or this node is itself a service facility. The family of constraints (31) ensures that a node k is not assigned to its service facility through a node i, unless node i is also covered or a facility itself. The family of constraints (32) and (33) set the value of d_i, a bound on the distance from node i to its facility, if there exists a path of length at most R. We note that (33) are equivalent to the well-known Miller-Tucker-Zemlin subtour elimination constraints, extended by our edge length reduction variables. In Fig. 4, an illustration of constraints (33) is depicted, in which the dark node represents a facility. Finally, the families of constraints (4) and (34) establish the bounds on the amount of length edge reductions.

Note that constraints (30) and (31) ensure that

$$z_{ij} + z_{ji} \leq 1, \quad [i, j] \in E. \tag{36}$$

The following result presents an improvement to the previous formulation, proving that the integrality condition on the x-variables can be relaxed, providing a new family of valid inequalities, and strengthening a family of constraints.

Proposition 4. *The formulation (Path) can be enhanced as follows:*

i) *The binary condition for the x-variables can be relaxed.*
ii) *The following are valid inequalities for (Path).*

$$d_i \geq \sum_{j:[i,j] \in E} \left(\ell_{[i,j]} - u_{[i,j]} \right) z_{ij}, \quad i \in V. \tag{37}$$

iii) *Constraints (33) can be reinforced as follows*

$$d_i \geq d_j + \ell_{[i,j]} z_{ij} - \delta_{[i,j]} - R(1 - z_{ij}) + z_{ji} (R - \ell_{[i,j]})^+, \quad [i, j] \in E, \tag{38}$$

where $a^+ := \max\{a, 0\}$.

Proof. The proof of *i)* is very similar to the proof of the first part of Lemma 1.

Regarding statement *ii)*, the idea behind these constraints is based on the fact that if a non-facility node is covered, the distance from that node to its assigned facility will be at least the length of the adjacent edge in the path to the service provider, minus the maximally allowed edge length reduction, i.e.,

$$d_i \geq \left(\ell_{[i,j]} - u_{[i,j]} \right) z_{ij}, \quad [i, j] \in E. \tag{39}$$

Moreover, each node is linked to at most one other node in the path to its service facility because of constraint (30).

In order to prove result *iii)*, we analyse the possible cases. Since the z-variables are binary and constraints (36) are satisfied, we get the following four possibilities in the optimal solution for $[i, j] \in E$: a) $z^*_{ij} = z^*_{ji} = 0$, b) $z^*_{ij} = 1, z^*_{ji} = 0$, c.1) $z^*_{ij} = 0, z^*_{ji} = 1$, with $R - \ell_{[i,j]} \leq 0$, and c.2) $z^*_{ij} = 0, z^*_{ji} = 1$, with $R - \ell_{[i,j]} > 0$. In cases a), b), and c.1) constraints (33) are fulfilled. Hence, (38) is valid. Therefore, we focus on case c.2). In this case, since the lower bounds of d_i is only given by (33), we can assume without loss of generality that (33) is satisfied with equality, i.e.:

$$d^*_j = d^*_i + \ell_{[i,j]} - \delta^*_{[i,j]}.$$

Therefore,

$$d_i^* = d_j^* - \ell_{[i,j]} + \delta_{[i,j]}^* \geq d_j^* - \ell_{[i,j]} - \delta_{[i,j]}^*.$$

Hence, since $z_{ij}^* = 0$, $z_{ji}^* = 1$, we have that:

$$d_i^* \geq d_j^* + \ell_{[i,j]} z_{ij}^* - \delta_{[i,j]}^* - R(1 - z_{ij}^*) + z_{ji}^*(R - \ell_{[i,j]}),$$

and consequently the family of inequalities (38) holds. Finally, this clearly strengthens the family of constraints (33). □

In what follows, we will refer to (Path) as the formulation where the above proposition has been applied. Next, we present some valid inequalities linking x- and z-variables. These inequalities are designed to strengthen the formulation. In (Path), it is not possible to represent which service is assigned to a given node. Therefore, the ideas of Proposition 2 cannot be used. Although it is possible to obtain valid inequalities for this formulation based on constraints (28).

Proposition 5. *The following families of constraints are valid inequalities for (Path):*

$$x_j \leq \sum_{k:[i,k]\in E, \ell_{[i,k]} - u_{[i,k]} \leq d(i,j)} z_{ik} + x_i, \qquad i \in V, j \in \hat{V}_i, \tag{40}$$

$$\sum_{i\in W} \sum_{j\in W:[i,j]\in E} z_{ij} \leq |W| - 1, \quad W \subset V, 3 \leq |W| \leq n - p, \tag{41}$$

$$d_i \geq \sum_{j:[i,j]\in S_1} \big(d_j - R(1 - z_{ij})\big) + \sum_{j:[i,j]\in S_2} \big(\ell_{[i,j]} z_{ij} - \delta_{[i,j]}\big), \; i \in V, S_1, S_2 \subseteq \Gamma_i. \tag{42}$$

Proof. If a facility is open at some node j whose distance to node i before upgrading the network was lower than or equal to the coverage radius (hypothesis of Proposition 1), we can be sure that node i will be covered by some facility. Therefore, in order to eliminate possible symmetries, we assume that either i is a facility itself or the immediate successor of node i on its path to a service facility is a node whose distance to i after upgrading can be smaller than or equal to $d(i, j)$. Using the above argument, the family of constraints (40) is obtained.

Secondly, we can include the valid inequalities (41) to avoid cycles. These inequalities are not required in (Path) because the family of constraints (33) or equivalently (38) avoid cycles in any feasible solution. However, they can improve the linear relaxation bounds.

Finally, we prove that constraints (42) are valid inequalities. Using constraint (30), we know that in any feasible solution, for each $i \in V$, there is at most one $j_0 \in V$, $[i, j_0] \in E$ such that $z_{ij_0} = 1$.

On the one hand, if $\sum_{j:[i,j]\in E} z_{ij} = 0$, we obtain that $d_i = 0$ by (32). Furthermore, this latter set of constraints ensures that $d_j \leq R$, for $j \in V$. Then, $d_j - R(1 - z_{ij}) \leq 0$, for $[i, j] \in E$. Moreover, since the δ-variables are non-negative and $z_{ij} = 0$, for $j \in V$, we obtain that $\ell_{[i,j]} z_{ij} - \delta_{[i,j]} \leq 0$, for $[i, j] \in E$. Hence, it holds that:

$$0 = d_i \geq \sum_{j:[i,j]\in S_1} \big(d_j - R(1 - z_{ij})\big) + \sum_{j:[i,j]\in S_2} \big(\ell_{[i,j]} z_{ij} - \delta_{[i,j]}\big), \quad S_1, S_2 \subseteq \Gamma_i.$$

On the other hand, if exists $j_0 \in V$, $[i, j_0] \in E$, such that $z_{ij_0} = 1$, using (33) we know that:

$$d_i \geq d_{j_0} - R(1 - z_{ij_0}) + \ell_{[i,j_0]} z_{ij_0} - \delta_{[i,j_0]}.$$

Furthermore, $d_j - R(1 - z_{ij}) \leq 0$, for $[i, j] \in E$, such that $j \neq j_0$, and $\ell_{[i,j]} z_{ij} - \delta_{[i,j]} \leq 0$, for $[i, j] \in E$, such that $j \neq j_0$. Therefore:

$$d_i \geq \sum_{j:[i,j]\in S_1} \big(d_j - R(1 - z_{ij})\big) + \sum_{j:[i,j]\in S_2} \big(\ell_{[i,j]} z_{ij} - \delta_{[i,j]}\big), \quad S_1, S_2 \subseteq \Gamma_i.$$

Thus, we conclude that constraints (42) are valid inequalities. □

Observe that in preliminary computational experiments the addition of the following constraints from family (41) as cuts in the branching tree was quite effective (more details are provided in Section 6):

$$z_{ij} + z_{ji} + z_{jk} + z_{kj} + z_{ik} + z_{ki} \leq 2, \quad [i, j], [j, k], [i, k] \in E. \tag{43}$$

Next, we solve the separation problem in the family of constraints (42), i.e., given a solution of the LP-relaxation of the formulation, find one or more constraints in family (42) that are not satisfied. Hence, sets S_1 and S_2 that maximises the right-hand-side of the inequality have to be identified. Let $\bar{d}$, $\bar{\delta}$, and $\bar{z}$ be the optimal vectors of values of the d-, δ-, and z-variables, respectively, in a node of the branching tree during the resolution of an instance of formulation (Path). Then, it is straightforward to conclude that one of the following constraints maximises the right-hand-side of (42):

$$d_i \geq \sum_{j:[i,j]\in E, \bar{d}_j > R(1-\bar{z}_{ij})} \big(d_j - R(1 - z_{ij})\big) + \sum_{j:[i,j]\in E, \ell_{[i,j]}\bar{z}_{ij} > \bar{\delta}_{[i,j]}} \big(\ell_{[i,j]} z_{ij} - \delta_{[i,j]}\big), \quad i \in V, \tag{44}$$

$$d_i \geq \sum_{j:[i,j]\in E, \bar{d}_j + \ell_{[i,j]}\bar{z}_{ij} > R(1-\bar{z}_{ij}) + \bar{\delta}_{[i,j]}} \big(d_j + \ell_{[i,j]} z_{ij} - R(1 - z_{ij}) - \delta_{[i,j]}\big), \quad i \in V. \tag{45}$$

In Section 6, the performance of this formulation and the effectiveness of the valid inequalities will be analysed.

5. Path-coverage formulation

In this section, we introduce a third formulation, which merges components from the first formulation with the second formulation. More precisely, we add the assignment variables y of (Flow-Cov) to (Path). For the sake of clarity, all variables of this formulation are explained below.

Decision variables

x_j	1, if there is a facility at node j, and 0, otherwise, for $j \in V$.
y_{ij}	1, if node i is assigned to a facility at node j, and 0, otherwise, for $i, j \in V, i \neq j$.
z_{ij}	1, if node j is the next node on a path of length $\leq R$ from i to its service facility, and 0, otherwise, for $[i, j] \in E$.
d_i	An upper bound of the length of the built path from node i to its service facility, for $i \in V$.
$\delta_e = \delta_{[i,j]}$	The amount of reduction of the length of edge $e = [i, j]$, for $e \in E$.

Next, we present the formulation of problem Up-MCLP using the variables described above:

$$(Path - Cov) \quad \max \sum_{i\in V} w_i \left(x_i + \sum_{j:[i,j]\in E} z_{ij} \right)$$

$$\text{s.t.} \quad (1), (3), (4), (12) - (13), (30) - (32), (34), (35), (38),$$

$$\sum_{k\in V\setminus\{i\}} y_{ik} = \sum_{j:[i,j]\in E} z_{ij}, \qquad i \in V, \tag{46}$$

$$y_{ik} \geq z_{ij} + z_{ji} + y_{jk} - 1, \qquad k \in V \setminus \{i, j\}, [i, j] \in E, \tag{47}$$

$$y_{ij} \geq z_{ij} + z_{ji} + x_j - 1, \qquad [i, j] \in E. \tag{48}$$

The family of constraints (46) establishes that if a node is assigned to a service facility, then there is a path from this node to its facility and vice versa. Constraints (47) ensure that if two nodes are on the same path, i.e., $z_{ij} = 1$, they must be assigned to the same facility k. Constraints (48) represent the particular case where node j hosts a service provider. Observe that the objective

function can also be expressed as follows:

$$\sum_{i\in V} w_i \left(x_i + \sum_{j\in V\setminus\{i\}} y_{ij} \right).$$

But in preliminary computational experiments, we have found that the objective function with the z-variables outperforms the one with the y-variables. This analysis was again carried out using the data described in Section 6.1 and the comparison was based on the number of instances solved to optimality within the time limit, the time employed to obtain the optimal solutions, the MIP relative gaps, the best solution gaps, and the linear relaxation gaps. More details about these performance values can be found in Section 6.

Below, we present a result that proves that the integrality condition of some families of variables can be relaxed.

Lemma 2. *The binary condition on the x-variables and the y-variables can be relaxed.*

Proof. The proof of the first part of this lemma is very similar to the proof of the first part of Lemma 1. Regarding the integrality condition on the y-variables, since their values are given by the values of the z-variables, it is straightforward to conclude that given an optimal solution it is possible to find another optimal solution in which the y-variables are integer. □

In contrast to (Path), this formulation controls the service facilities to which the nodes are assigned with the y-variables. This information allows us to use a more sophisticated preprocessing phase. For doing so, one of the results presented in Section 3.1 is used. Under the hypothesis of Proposition 2, i.e., a facility at node i will never be assigned to a facility located at node j, for $i, j \in V$, and vice versa, variables y_{ij} and y_{ji} are removed from all the constraints in which they are included (fixed to zero and not included in the formulation to save memory). Furthermore, using the information obtained in the preprocessing phase we can develop new valid inequalities, which are discussed in the next subsection.

5.1. Valid inequalities

This subsection is devoted to presenting valid inequalities for formulation (Path-Cov). We start by remarking that the valid inequalities (29) obtained for formulation (Flow-Cov) can also be implemented in (Path-Cov). Similarly, all the valid inequalities obtained for formulation (Path) are still valid for (Path-Cov), namely, the families of constraints (37) and (39)–(45). However, the additional information provided by the y-variables in formulation $(Path-Cov)$ can be used to strengthen some of them. The ones that can be enhanced using the covering variables are described below.

First, the lower bound for the d-variables can be improved, i.e., constraint (37) can be enhanced as:

$$d_i \geq \sum_{j\in V\setminus\{i\}} d(i,j,\delta^{ij}) y_{ij}, \quad i \in V. \tag{49}$$

Recall that $d(i,j,\delta^{ij})$ represents the distance between nodes i and j using the most favourable edge length reductions satisfying the budget constraint (4). In what follows, we will refer to (Path-Cov) as the formulation (Path-Cov) in which constraints (49) are included.

Finally, we present a new family of valid inequalities that reinforces constraints (47). The objective of this reinforcement is to improve the resolution of the formulation. It is based on the fact that if two nodes are linked (the sum of their z-variables is one), then both nodes will be assigned to the same service facility.

Lemma 3. *The following are valid inequalities for (Path-Cov):*

$$z_{ij} + z_{ji} \leq \sum_{k\in W, k\neq i} y_{ik} + x_i \mathcal{I}_W(i) + \sum_{k\notin W, k\neq j} y_{jk} + x_j(1 - \mathcal{I}_W(j)), \quad [i,j] \in E, W \subseteq V, \tag{50}$$

where $\mathcal{I}_W(i)$ is the indicator function, i.e., $\mathcal{I}_W(i) = 1$ if $i \in W$ and 0 otherwise.

Note that, for the case $W = \{k\}$, for $k \in V$, we obtain $z_{ij} + z_{ji} \leq y_{ik} + \sum_{t\in V, t\neq k, t\neq j} y_{jt} + x_j$, using constraints (30) and (36), it holds that $z_{ij} + z_{ji} \leq y_{ik} + \sum_{t\in V, t\neq k, t\neq j} y_{jt} + x_j \leq y_{ik} + 1 - y_{jk}$. Hence, some constraints of family (50) are tighter than (47).

As the cardinality of (50) is exponential, we solve the separation problem in this family of constraints. Therefore, the set W that minimises the right-hand-side of constraints (50) has to be identified. Let $\bar{y}$ ($\bar{x}$) be the optimal vector values of y-variables (x-variables) in a node of the branching tree during the resolution of an instance of formulation (Path-Cov). Then, it is straightforward to conclude that the following constraints minimise the right-hand-side of (50).

$$z_{ij} + z_{ji} \leq \sum_{k\in V: \bar{y}_{ik}\leq \bar{y}_{jk}, \bar{y}_{ik}\leq \bar{x}_j} y_{ik} + x_i \mathcal{I}_{\{k:\bar{x}_k\leq \bar{y}_{jk}\}}(i) + \sum_{k\in V: \bar{y}_{jk}<\bar{y}_{ik}, \bar{y}_{jk}<\bar{x}_i} y_{jk} + x_j \mathcal{I}_{\{k:\bar{x}_k<\bar{y}_{ik}\}}(j), \quad [i,j] \in E. \tag{51}$$

Note that if a pair of nodes satisfies at least one of the conditions of Proposition 2, their corresponding y-variables can be removed from all the constraints including the valid inequalities presented in this subsection.

In the following section, the performance of the three proposed formulations for Up-MCLP are compared.

6. Computational results

In this section, we present the results of several computational experiments which compare the performance of the three proposed formulations and show the improvements achieved thanks to the preprocessing phase and the inclusion of the valid inequalities developed throughout the paper. The experiments were conducted on an Intel(R) Xeon(R) W-2135 CPU 3.70 GHz 32 GB RAM, using CPLEX 20.1.0 in Concert Technology C++ with a time limit of 1800 s. We used the default parameter settings for CPLEX.

Regarding the preprocessing phase for formulations (Flow-Cov) and (Path-Cov), aiming to find a balance between preprocessing time and the quality of the $d(i,j,\delta^{ij})$ bounds for each pair $i, j \in V$, the following strategy has been implemented. First, we computed the matrix of pairwise shortest distances without upgrading and the matrix of pairwise shortest distances after upgrading all edges to their full maximum ($d(i,j,u)$) using the Floyd-Warshall algorithm. Then, we checked if the hypothesis of Proposition 1 or if the hypotheses *i)-iii)* of Proposition 2 are fulfilled. If either of these conditions is satisfied for a pair $i, j \in V$, we removed the corresponding variables and constraints and we used $d(i,j,u)$ as $d(i,j,\delta^{ij})$ in the valid inequalities that are required (as e.g. constraints (29)). This could be done because $d(i,j,u)$ is a lower bound of $d(i,j,\delta^{ij})$. If neither of these conditions were fulfilled for a given pair $i, j \in V$, we solved the linear relaxation of $\left(P_{d(i,j,\delta^{ij})}\right)$. The minimum between its optimal objective value and $d(i,j,u)$ is the value that we used as $d(i,j,\delta^{ij})$ in the corresponding valid inequalities. As before, this can be done because both values are lower bounds of $d(i,j,\delta^{ij})$. For sake of clarity, we summarise the process in Algorithm 1 .

In preliminary computational experiments, we checked the performance of the alternative formulation (Flow-Cov) for Up-MCLP

Algorithm 1: Preprocessing phase

Input: *Formulation.*
Output: *Preprocessed formulation.*
foreach $i, j \in V, i < j$ **do**
  Compute $d(i, j, u)$.
  if *the hypothesis of Proposition 1 or the hypotheses i)-iii) of Proposition 2 are fulfilled.* **then**
    1. *Apply the proposition removing the corresponding variables and constraints.*
    2. *Set* $d(i, j, \delta^{ij}) = d(i, j, u)$ *if it is required in any constraint.*
  else
    Solve the linear relaxation of $\left(P_{d(i,j,\delta^{ij})}\right)$ *with objective value* $LRP_{d(i,j,\delta^{ij})}$.*Set* $d(i, j, \delta^{ij}) = \max\{d(i, j, u), LRP_{d(i,j,\delta^{ij})}\}$ *if it is required in any constraint.*
  end
end
return *Preprocessed formulation.*

and the valid inequalities to identify which ones performed best in each formulation. After this preliminary choice, we conclude that the best formulations are:

a) Formulation (Flow-Cov) where constraints (6), (7), and (11) have been replaced with (17), (18), and (19) and the family of constraints (25) is included, named (Flow-Cov) for short.
b) Formulation (Path) where constraints (37) are included and constraints (33) have been reinforced by constraints (38). In what follows, we call it formulation (Path).
c) Formulation (Path) with constraints (40) as valid inequalities and (43) as particular case of (41) in a pool of user cuts, named (Path) + VI for short.
d) Formulation (Path-Cov) where constraints (49) are included, we call it formulation (Path-Cov).
e) Formulation (Path-Cov) including constraints (29) and (43) as particular case of (41) in a pool of user cuts, named (Path-Cov) + VI for short.

Regarding the families of valid inequalities that we have used, in preliminary investigation, we observe that the family of constraints (40) had a better performance than the family of constraints (43) for formulation (Path) in the majority of the tested cases. Similarly, in the case of (Path-Cov), the family of constraints (29) tended to provide a greater improvement in performance than the family of constraints (43). Concerning the families of valid inequalities that we have not included in the reported numerical experiments, we would like to remark that including the constraints (44), (45), and (51) in the branching tree was effective, as the number of nodes in which the instances were solved decreased. However, this procedure is time-consuming, so that even though the instances were solved on fewer nodes the overall computation time increased.

The rest of the section is structured as follows. First, the data used in the computational experiments are described. Second, the advantages of the preprocessing phase are shown. Then, the following subsections compare the different formulations with and without valid inequalities in complete graphs and in sparse graphs, respectively. These subsections illustrate the great value of the preprocessing phase and the addition of valid inequalities.

6.1. Data

The computational experiments were carried out on two different types of networks.

First, we generated instances adapting the procedure used in ReVelle, Scholssberg, & Williams (2008), Cordeau et al. (2019), among others. Nodes were given by points whose coordinates followed a uniform distribution over [0,30]. Then, we computed the complete graph where the length of the edges is the Euclidean distance between the nodes. We named these instances as "graph" followed by the number of vertices, e.g., "graph30" is a complete graph with 30 nodes and 435 edges.

Secondly, we used the uncapacitated p-median datasets from the OR-Library, called pmed, see Beasley (1990). As said in the documentation of these datasets, Floyd's algorithm was applied to obtain a symmetric allocation cost matrix that satisfied the triangle inequality. The main difference with respect to the previous datasets is that the p-median instances are sparser graphs (the number of edges is $n^2/50$).

The parameters have been chosen as described below. The number of facilities, p, was proportional to the number of vertices, i.e., $p \in \{1, n/10, n/20\}$. The node weights or demands, w_i for $i \in V$, were integers uniform randomly generated between 1 and 100. We tested three different coverage radii, R, such that we could cover approximately 50%, 60%, and 70% of the total demand when solving the maximal covering location problem without upgrading (DT_{MCLP}), i.e., $R \in \{R(50\%DT_{\text{MCLP}}), R(60\%DT_{\text{MCLP}}), R(70\%DT_{\text{MCLP}})\}$. Upgrading costs, c_e, for $e \in E$, were uniform randomly generated between 1 and 3. The upper bounds u_e, for $e \in E$, were uniform randomly generated from $(0, 30\%\ell_e)$, for $e \in E$. Then, the length of the edges was modified as $\ell_e + u_e$, for $e \in E$. This implies that the instances satisfy the triangle inequality when the full discount is applied in all edges. Finally, the budget B was computed as follows. First, we sorted the upgrade costs $c_e u_e$, for $e \in E$, in non-increasing order. Let ρ be a permutation of set E such that $c_{e_{\rho(1)}} u_{e_{\rho(1)}} \geq c_{e_{\rho(2)}} u_{e_{\rho(2)}} \geq \ldots \geq c_{e_{\rho(m)}} u_{e_{\rho(m)}}$. Then, since we are constructing a forest with p components (as seen in Section 4), we can assume that at most $n - p$ edges will be upgraded. Therefore, we computed the maximum required budget for upgrading the most expensive edges,

$$B_{max} = \sum_{t=1}^{n-p} u_{e_{\sigma(t)}} c_{e_{\sigma(t)}},$$

and selected $B \in \{0.5\%B_{\max}, 1\%B_{\max}, 5\%B_{\max}\}$.

6.2. Preprocessing phase

In this subsection, we show the enhancements provided by the preprocessing, i.e., Propositions 1 and 2. For doing so, we solve (Flow-Cov) and (Path-Cov) with and without preprocessing.

As an illustrative example, we include the results for graph40 in Table 3. The performance was similar in the rest of the datasets. The results are the average over five instances generated with the same procedure, varying only the random seed for the generator. The first column indicates the name of the dataset, the number of nodes and the number of edges. Next, the percentage of the maximal budget ($B\%$), and the number of located facilities are depicted (p), followed by the approximate percentage of covered demand in the MCLP without upgrading using this radius ($R\%$). The following four columns describe information about (Flow-Cov) without preprocessing. The first one shows the average time (in seconds) of solving the corresponding five instances. Observe that if any of these instances is not solved to optimality, 1800 s is considered as its solution time to compute this average. Then, the following column of this group depicts the MIP relative gap reported by CPLEX

Table 3
Performance of formulations (Flow-Cov) and (Path-Cov) with and without preprocessing.

Data	$B\%$	p	$R\%$	(Flow-Cov)												(Path-Cov)											
				Without preprocessing				With preprocessing								Without preprocessing				With preprocessing							
				t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	$R_c\%$	$R_v\%$	$R_{bv}\%$	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	$R_c\%$	$R_v\%$	$R_{bv}\%$
graph40 $\|V\| = 40, \|E\| = 780$	0.5	1	50	1042.5	0.0(5)	0.0	93.2	0.1	0.7	0.0(5)	0.0	2.3	97.0	97.0	96.9	1807.9	99.9(0)	5.7	93.2	0.1	2.3	0.0(5)	0.0	2.4	69.9	46.2	52.5
			60	1490.0	15.4(3)	0.0	52.3	0.1	3.8	0.0(5)	0.0	0.7	95.8	95.7	95.7	1804.2	74.8(0)	12.4	52.3	0.2	8.8	0.0(5)	0.0	0.7	63.1	37.6	43.5
			70	1556.8	7.3(4)	0.0	35.4	0.2	2.0	0.0(5)	0.0	1.0	96.0	96.0	95.9	1807.5	59.6(0)	13.4	35.4	0.2	724.2	1.0(3)	0.0	1.0	58.3	32.0	37.5
		2	50	1578.7	74.4(1)	3.9	87.0	0.1	0.4	0.0(5)	0.0	1.2	98.4	98.4	98.4	61.3	0.0(5)	0.0	88.7	0.1	0.4	0.0(5)	0.0	3.3	80.8	68.7	74.3
			60	1801.2	64.1(0)	4.2	57.0	0.1	0.4	0.0(5)	0.0	0.9	97.8	97.8	97.7	1093.2	4.3(3)	0.0	57.0	0.1	0.4	0.0(5)	0.0	0.9	77.5	60.0	66.2
			70	1801.7	40.5(0)	4.4	34.1	0.1	0.7	0.0(5)	0.0	1.8	97.9	97.8	97.8	1769.3	17.5(1)	1.2	34.1	0.1	2.4	0.0(5)	0.0	1.8	74.3	53.1	59.5
		4	50	2.9	0.0(5)	0.0	39.4	0.1	0.1	0.0(5)	0.0	1.0	99.1	99.1	99.0	0.3	0.0(5)	0.0	67.2	0.1	0.1	0.0(5)	0.0	2.8	86.7	83.4	87.3
			60	27.5	0.0(5)	0.0	48.6	0.1	0.2	0.0(5)	0.0	0.8	99.0	98.9	98.9	8.4	0.0(5)	0.0	56.8	0.1	0.2	0.0(5)	0.0	1.7	84.9	78.5	83.1
			70	1546.9	25.6(1)	0.6	34.3	0.1	1.0	0.0(5)	0.0	3.9	98.3	98.2	98.2	305.4	0.0(5)	0.0	34.3	0.1	0.6	0.0(5)	0.0	4.4	81.9	70.8	76.3
	1	1	50	1384.9	18.2(4)	0.1	86.0	0.1	1.1	0.0(5)	0.0	0.4	96.0	96.0	96.0	1807.3	98.4(0)	6.6	86.0	0.1	3.6	0.0(5)	0.0	0.5	68.8	45.5	51.7
			60	1611.6	17.1(3)	0.0	51.8	0.1	1.5	0.0(5)	0.0	0.4	95.2	95.2	95.1	1809.0	71.3(0)	10.7	51.8	0.1	5.4	0.0(5)	0.0	0.4	62.5	37.2	43.1
			70	1801.9	27.9(0)	0.0	34.0	0.2	2.6	0.0(5)	0.0	1.5	94.7	94.7	94.7	1816.8	58.6(0)	13.8	34.0	0.2	135.3	0.0(5)	0.0	1.5	56.9	31.3	36.7
		2	50	1579.3	69.8(1)	1.9	84.9	0.1	0.4	0.0(5)	0.0	0.9	98.1	98.1	98.0	95.8	0.0(5)	0.0	86.5	0.1	0.3	0.0(5)	0.0	1.4	80.4	68.4	74.1
			60	1801.4	63.8(0)	6.1	53.2	0.1	0.8	0.0(5)	0.0	2.8	96.8	96.8	96.8	1609.6	12.4(1)	0.0	53.2	0.1	1.7	0.0(5)	0.0	2.8	76.5	59.3	65.4
			70	1801.6	43.3(0)	7.6	31.6	0.1	1.9	0.0(5)	0.0	3.0	96.7	96.7	96.6	1804.5	23.9(0)	1.2	31.6	0.1	41.9	0.0(5)	0.0	3.1	73.1	52.3	58.6
		4	50	2.9	0.0(5)	0.0	39.4	0.1	0.1	0.0(5)	0.0	1.0	99.1	99.1	99.0	0.3	0.0(5)	0.0	67.2	0.1	0.1	0.0(5)	0.0	2.8	86.7	83.4	87.3
			60	27.6	0.0(5)	0.0	48.6	0.1	0.2	0.0(5)	0.0	0.8	99.0	98.9	98.9	8.4	0.0(5)	0.0	56.8	0.1	0.2	0.0(5)	0.0	1.7	84.9	78.5	83.1
			70	1547.3	25.7(1)	0.6	34.3	0.1	1.0	0.0(5)	0.0	3.9	98.3	98.2	98.2	301.5	0.0(5)	0.0	34.3	0.1	0.6	0.0(5)	0.0	4.4	81.9	70.8	76.3
	5	1	50	1805.7	88.6(0)	3.3	82.1	0.1	1.7	0.0(5)	0.0	0.3	95.0	95.0	95.0	1807.6	87.3(0)	3.3	82.1	0.1	0.6	0.0(5)	0.0	0.3	67.7	44.9	51.0
			60	1805.2	46.(0)	0.4	45.4	0.2	2.1	0.0(5)	0.0	0.5	94.3	94.3	94.3	1807.6	68.3(0)	12.8	45.4	0.1	16.3	0.0(5)	0.0	0.5	61.6	36.7	42.5
			70	1802.5	43.1(0)	7.4	31.0	0.2	1.9	0.0(5)	0.0	0.0	94.2	94.2	94.1	1811.1	44.1(0)	7.9	31.0	0.2	3.6	0.0(5)	0.0	0.0	56.3	31.0	36.3
		2	50	1800.6	106.7(0)	12.8	79.8	0.1	0.4	0.0(5)	0.0	1.1	97.4	97.4	97.3	211.0	0.0(5)	0.0	81.6	0.1	0.4	0.0(5)	0.0	1.1	79.7	67.9	73.5
			60	1801.3	61.7(0)	8.6	47.6	0.1	1.1	0.0(5)	0.0	0.3	96.1	96.0	96.0	1804.8	14.4(0)	0.0	47.6	0.1	0.7	0.0(5)	0.0	0.3	75.7	58.7	64.8
			70	1802.0	41.7(0)	9.9	27.6	0.1	1.5	0.0(5)	0.0	0.0	95.5	95.4	95.4	1804.3	21.3(0)	1.5	27.6	0.1	2.0	0.0(5)	0.0	0.0	71.8	51.4	57.6
		4	50	3.1	0.0(5)	0.0	36.1	0.1	0.1	0.0(5)	0.0	1.2	98.9	98.8	98.8	0.3	0.0(5)	0.0	63.4	0.1	0.1	0.0(5)	0.0	1.3	86.4	83.1	87.1
			60	512.0	0.0(5)	0.0	45.9	0.1	0.2	0.0(5)	0.0	0.0	98.5	98.4	98.4	6.4	0.0(5)	0.0	54.1	0.1	0.2	0.0(5)	0.0	0.0	84.4	78.1	82.7
			70	1801.1	35.4(0)	6.0	27.1	0.1	1.0	0.0(5)	0.0	1.5	97.4	97.3	97.3	463.7	1.4(4)	0.0	27.1	0.1	0.8	0.0(5)	0.0	1.5	80.8	70.1	75.5

(G%) and in brackets the number of instances solved to optimality within the time limit. Next, it is provided the best solution gap, (G^t_{BS}%), computed as follows:

$$G^t_{BS}\% = \frac{BS^t - BS}{BS^t} \cdot 100,$$

where BS is the best MIP objective value found within the time limit by the formulation and BS^t is the best MIP solution value found within the time limit across all formulations. Finally, it is shown the linear relaxation gap, (G^t_{LP}%), computed as follows:

$$G^t_{LP}\% = \frac{LP - BS^t}{BS^t} \cdot 100,$$

where LP is the optimal solution value of the linear relaxation of the formulation. Note that G^t_{LP}% enables us to compare the linear relaxation of the formulations with each other (it could be possible that G% is greater than G^t_{LP}%). The following blocks of columns depict information about the rest of formulations, (Flow-Cov) with preprocessing and (Path-Cov) with and without preprocessing. Observe that the blocks corresponding to formulations with preprocessing include eight columns. The first column of these blocks reports the average time of the preprocessing phase in seconds, the next one shows the average total time (in seconds) of solving the corresponding five instances including the preprocessing time. The third, the fourth, and the fifth columns report the average G%, G^t_{BS}, and G^t_{LP} respectively. Then, the average percentage of reduction in the number of constraints (R_c%), variables (R_v%), and binary variables (R_{bv}%) are depicted. The percentage of reduction in the number of constrains is computed for formulation (Flow-Cov) as follows:

$$R_c\% = \frac{\#\text{constraint of (Flow-Cov) without prep.} - \#\text{constraint of (Flow-Cov) with prep.}}{\#\text{constraint of (Flow-Cov) without prep.}} \cdot 100.$$

The others are calculated analogously. For interested readers, a similar table that compares the performance of the different phases of the preprocessing can be found in supplementary material.

As can be appreciated in Table 3, the preprocessing phase yields a huge reduction in computation time. For example, using formulation (Flow-Cov) without preprocessing, only 53 instances are solved to optimality within the time limit, while using formulation (Flow-Cov) with preprocessing, all instances are solved (135) and the average time of solving each instance (including the time of preprocessing) is 1.1 s. Furthermore, the reduction in the number of constraints, variables and binary variables is also very large, approximately 97% on average. In formulation (Path-Cov), the reduction of time in the resolution process and the reduction of the size of the problem are also very substantial.

Based on the above results, we conclude that the preprocessing phase presented in the paper is extremely useful and effective. Therefore, in the subsequent computational experiments, the preprocessing phase is included.

6.3. Results for complete graphs

In this subsection, we compare the proposed formulations for complete graphs highlighting the effectiveness of the valid inequalities developed. For interested readers, non-parametric tests (Friedman test and Post-Hoc Holland adjust) to assess the statistical significance of the comparison among the different formulations can be found in supplementary material. We used the shiny application shinytest[1], see Carrasco, García, Rueda, Das, & Herrera (2020) for further details. The non-parametric tests show that there are significant differences between the formulations presented.

[1] https://github.com/JacintoCC/shinytests

The results of the smaller datasets (graph30 and graph40) are depicted in Table 4. As before, the provided results are the average over five instances generated with the same procedure, varying only the random seed for the generator. The table describes information about (Flow-Cov), (Path), (Path) + VI and (Path-Cov), and its structure is similar to that of Table 3. Observe that the blocks corresponding to the (Path) and (Path) + VI formulations have no preprocessing time, since we did not provide a preprocessing for these formulations. Note also that the results of (Path-Cov)+ VI are not included because they are really similar to (Path-Cov). The differences between these formulations will be shown in instances with a larger number of nodes and edges. Moreover, since several of the valid inequalities are included as cuts in the branching tree, the linear relaxation gap G^t_{LP}% of the formulation with valid inequalities is practically the same as the one for the formulation without valid inequalities. Therefore, they do not appear again in the table.

Finally, the formulation that provided the smallest average total time is highlighted. If any of the five instances were not solved to optimality, the formulation that solved more instances is shown in bold.

The results in Table 4 show that formulations (Flow-Cov) and (Path-Cov) outperform (Path) and (Path) + VI (the resolution times, the number of instances solved, the MIP relative gap, the best solution gap, and the linear relaxation gap of these formulations are worse). Moreover, it is clear from these results that the valid inequalities improve the performance of formulation (Path) as shown in the number of instances solved to optimality (208 instances with respect to 217 instances) and the average total time (489 s with respect to 416 s). However, this improvement is not large enough to make this formulation competitive with respect to formulations (Flow-Cov) and (Path-Cov) on complete graphs. Nevertheless, it can be seen that formulations (Path) and (Path) + VI in graph40 solve many more instances than formulations (Flow-Cov) and (Path-Cov) without preprocessing (85 and 91 instances with respect to 53 and 64 instances), as can be seen in Table 3.

In Table 5, the average of the number of constraints (c), the number of variables (v), the number of binary variables (bv), and the number of nodes visited in the branching tree (nodes) for each dataset (graph30 and graph40) and each formulation is reported. A detailed table can be found in the supplementary material. It can be seen that the dimension of (Flow-Cov) is much larger than the others. Note also that the inclusion of valid inequalities in formulation (Path) decreases considerably the number of nodes used. Observe that (Path-Cov) is the one with intermediate size and number of nodes.

In Table 6, a second set of computational experiments is reported. Here, datasets of larger size (graph100 and graph120) are solved so that formulations (Flow-Cov), (Path-Cov) and (Path-Cov) + VI can be compared. Table 6 has the same structure as Table 4, but now (Path-Cov) + VI is included whereas (Path) and (Path) + VI are not. Moreover, a similar analysis to the one in Table 5 is reported in Table 7 for graph100 and graph120. A detailed table can be found in the supplementary material. For the purpose of a clearer comparison of these formulations, the performance profile graph of the number of solved instances is depicted in Fig. 5.

Analysing the results shown in Table 6, we can conclude that the difficulty of solving the instances is highly dependent on the parameters (B, R and p). It can be observed that the instances become more difficult as B and p decrease and R increases. As shown in Table 6 and Fig. 5, the performance of these formulations is quite similar. An interesting observation is that they complement each other. In other words, there are instances in which formulation (Flow-Cov) did not find the optimal solution within the time

Table 4
Performance of formulations (Flow-Cov), (Path), (Path) + VI, (Path-Cov) on graph30 and graph40.

Data	$B\%$	p	$R\%$	(Flow-Cov)					(Path)				(Path) + VI			(Path-Cov)				
				t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{total}	$G\%$	$G^t_{BS}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$
graph30, $\|V\| = 30, \|E\| = 435$	0.5	1	50	**0.1**	**0.5**	**0.0(5)**	**0.0**	**2.2**	54.6	0.0(5)	0.0	84.8	13.2	0.0(5)	0.0	0.1	0.7	0.0(5)	0.0	2.2
			60	**0.1**	**0.4**	**0.0(5)**	**0.0**	**0.9**	533.2	6.1(4)	0.0	50.9	144.1	0.0(5)	0.0	0.1	1.4	0.0(5)	0.0	0.9
			70	**0.1**	**0.5**	**0.0(5)**	**0.0**	**0.7**	1162.9	14.4(2)	0.2	34.7	865.0	8.2(3)	0.0	0.1	0.8	0.0(5)	0.0	0.7
		2	50	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.6**	0.3	0.0(5)	0.0	83.4	0.3	0.0(5)	0.0	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.6**
			60	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.0**	0.7	0.0(5)	0.0	54.3	0.5	0.0(5)	0.0	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.0**
			70	**0.1**	**0.3**	**0.0(5)**	**0.0**	**1.0**	5.0	0.0(5)	0.0	36.0	2.9	0.0(5)	0.0	**0.0**	**0.3**	**0.0(5)**	**0.0**	**1.1**
		3	50	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.4**	**0.1**	**0.0(5)**	**0.0**	**79.2**	**0.1**	**0.0(5)**	**0.0**	**0.0**	**0.1**	**0.0(5)**	**0.0**	**3.0**
			60	**0.0**	**0.1**	**0.0(5)**	**0.0**	**2.6**	0.3	0.0(5)	0.0	53.8	0.3	0.0(5)	0.0	**0.0**	**0.1**	**0.0(5)**	**0.0**	**2.7**
			70	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.7**	0.8	0.0(5)	0.0	32.0	0.7	0.0(5)	0.0	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.7**
	1	1	50	**0.1**	**0.8**	**0.0(5)**	**0.0**	**4.1**	102.7	0.0(5)	0.0	77.2	49.7	0.0(5)	0.0	0.1	1.9	0.0(5)	0.0	4.2
			60	**0.1**	**0.6**	**0.0(5)**	**0.0**	**0.7**	620.2	2.9(4)	0.0	45.3	479.4	1.3(4)	0.0	0.1	2.2	0.0(5)	0.0	0.7
			70	**0.1**	**0.8**	**0.0(5)**	**0.0**	**2.0**	1370.0	10.0(3)	0.0	32.4	967.0	11.9(3)	0.0	0.1	4.1	0.0(5)	0.0	2.1
		2	50	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.6**	0.3	0.0(5)	0.0	83.4	0.3	0.0(5)	0.0	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.6**
			60	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.3**	0.7	0.0(5)	0.0	52.8	0.5	0.0(5)	0.0	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.3**
			70	**0.1**	**0.6**	**0.0(5)**	**0.0**	**1.5**	11.7	0.0(5)	0.0	32.5	6.1	0.0(5)	0.0	0.0	0.9	0.0(5)	0.0	1.5
		3	50	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.4**	**0.1**	**0.0(5)**	**0.0**	**79.2**	**0.1**	**0.0(5)**	**0.0**	**0.0**	**0.1**	**0.0(5)**	**0.0**	**3.0**
			60	**0.0**	**0.1**	**0.0(5)**	**0.0**	**2.6**	0.3	0.0(5)	0.0	53.8	0.3	0.0(5)	0.0	**0.0**	**0.1**	**0.0(5)**	**0.0**	**2.7**
			70	**0.1**	**0.1**	**0.0(5)**	**0.0**	**0.7**	0.8	0.0(5)	0.0	32.0	0.7	0.0(5)	0.0	**0.0**	**0.1**	**0.0(5)**	**0.0**	**0.7**
	5	1	50	0.1	0.6	0.0(5)	0.0	0.0	254.6	0.0(5)	0.0	66.2	98.3	0.0(5)	0.0	**0.1**	**0.2**	**0.0(5)**	**0.0**	**0.0**
			60	0.1	0.7	0.0(5)	0.0	0.0	1097.0	5.5(3)	0.0	42.9	653.4	3.7(4)	0.0	**0.1**	**0.2**	**0.0(5)**	**0.0**	**0.0**
			70	**0.1**	**0.8**	**0.0(5)**	**0.0**	**0.0**	1384.6	13.3(2)	0.0	28.8	1329.7	14.(2)	0.0	0.1	1.5	0.0(5)	0.0	0.0
		2	50	0.0	0.2	0.0(5)	0.0	2.2	0.4	0.0(5)	0.0	70.6	0.4	0.0(5)	0.0	**0.0**	**0.1**	**0.0(5)**	**0.0**	**3.6**
			60	**0.0**	**0.2**	**0.0(5)**	**0.0**	**1.4**	1.0	0.0(5)	0.0	49.4	0.8	0.0(5)	0.0	**0.0**	**0.2**	**0.0(5)**	**0.0**	**1.4**
			70	0.1	0.5	0.0(5)	0.0	1.2	20.4	0.0(5)	0.0	30.7	11.1	0.0(5)	0.0	**0.1**	**0.4**	**0.0(5)**	**0.0**	**1.2**
		3	50	**0.0**	**0.1**	**0.0(5)**	**0.0**	**1.1**	**0.1**	**0.0(5)**	**0.0**	**72.1**	**0.1**	**0.0(5)**	**0.0**	**0.0**	**0.1**	**0.0(5)**	**0.0**	**1.8**
			60	**0.1**	**0.2**	**0.0(5)**	**0.0**	**3.2**	0.5	0.0(5)	0.0	47.4	0.5	0.0(5)	0.0	**0.0**	**0.2**	**0.0(5)**	**0.0**	**4.6**
			70	0.1	0.3	0.0(5)	0.0	4.0	1.3	0.0(5)	0.0	26.2	0.9	0.0(5)	0.0	**0.0**	**0.2**	**0.0(5)**	**0.0**	**4.2**

(*continued on next page*)

Table 4 (*continued*)

Data	$B\%$	p	$R\%$	(Flow-Cov)					(Path)				(Path) + VI			(Path-Cov)				
				t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{total}	$G\%$	$G^t_{BS}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$
graph40, $\|V\| = 40$, $\|E\| = 780$	0.5	1	50	**0.1**	**0.7**	**0.0(5)**	**0.0**	**2.3**	1395.1	31.1(2)	0.0	93.2	1089.2	39.(3)	6.1	0.1	2.3	0.0(5)	0.0	2.4
			60	**0.1**	**3.8**	**0.0(5)**	**0.0**	**0.7**	1801.0	48.3(0)	5.1	52.3	1802.6	45.9(0)	5.8	0.2	8.8	0.0(5)	0.0	0.7
			70	**0.2**	**2.0**	**0.0(5)**	**0.0**	**1.0**	1801.6	43.6(0)	8.2	35.4	1803.7	62.1(0)	16.9	0.2	724.2	1.0(3)	0.0	1.0
		2	50	**0.1**	**0.4**	**0.0(5)**	**0.0**	**1.2**	6.2	0.0(5)	0.0	88.6	4.5	0.0(5)	0.0	**0.1**	**0.4**	**0.0(5)**	**0.0**	**3.3**
			60	**0.1**	**0.4**	**0.0(5)**	**0.0**	**0.9**	143.6	0.0(5)	0.0	57.0	53.9	0.0(5)	0.0	**0.1**	**0.4**	**0.0(5)**	**0.0**	**0.9**
			70	**0.1**	**0.7**	**0.0(5)**	**0.0**	**1.8**	878.8	2.3(3)	0.0	34.1	549.8	1.1(4)	0.0	0.1	2.4	0.0(5)	0.0	1.8
		4	50	**0.1**	**0.1**	**0.0(5)**	**0.0**	**1.0**	0.2	0.0(5)	0.0	67.2	0.2	0.0(5)	0.0	**0.1**	**0.1**	**0.0(5)**	**0.0**	**2.8**
			60	**0.1**	**0.2**	**0.0(5)**	**0.0**	**0.8**	0.8	0.0(5)	0.0	56.8	0.6	0.0(5)	0.0	**0.1**	**0.2**	**0.0(5)**	**0.0**	**1.7**
			70	0.1	1.0	0.0(5)	0.0	3.9	6.9	0.0(5)	0.0	34.3	4.4	0.0(5)	0.0	**0.1**	**0.6**	**0.0(5)**	**0.0**	**4.4**
	1	1	50	**0.1**	**1.1**	**0.0(5)**	**0.0**	**0.4**	1636.2	36.3(1)	3.0	86.0	1367.6	32.2(3)	3.2	0.1	3.6	0.0(5)	0.0	0.5
			60	**0.1**	**1.5**	**0.0(5)**	**0.0**	**0.4**	1800.5	48.2(0)	7.1	51.8	1801.1	55.8(0)	9.7	0.1	5.4	0.0(5)	0.0	0.4
			70	**0.2**	**2.6**	**0.0(5)**	**0.0**	**1.5**	1801.0	42.2(0)	8.1	34.0	1802.9	48.6(0)	11.9	0.2	135.3	0.0(5)	0.0	1.5
		2	50	0.1	0.4	0.0(5)	0.0	0.9	10.1	0.0(5)	0.0	86.5	5.9	0.0(5)	0.0	**0.1**	**0.3**	**0.0(5)**	**0.0**	**1.4**
			60	**0.1**	**0.8**	**0.0(5)**	**0.0**	**2.8**	255.7	0.0(5)	0.0	53.2	168.2	0.0(5)	0.0	0.1	1.7	0.0(5)	0.0	2.8
			70	**0.1**	**1.9**	**0.0(5)**	**0.0**	**3.0**	1058.5	3.1(3)	0.0	31.6	817.9	1.8(3)	0.0	0.1	41.9	0.0(5)	0.0	3.1
		4	50	**0.1**	**0.1**	**0.0(5)**	**0.0**	**1.0**	0.2	0.0(5)	0.0	67.2	0.2	0.0(5)	0.0	**0.1**	**0.1**	**0.0(5)**	**0.0**	**2.8**
			60	**0.1**	**0.2**	**0.0(5)**	**0.0**	**0.8**	0.8	0.0(5)	0.0	56.8	0.6	0.0(5)	0.0	**0.1**	**0.2**	**0.0(5)**	**0.0**	**1.7**
			70	0.1	1.0	0.0(5)	0.0	3.9	6.9	0.0(5)	0.0	34.3	4.4	0.0(5)	0.0	**0.1**	**0.6**	**0.0(5)**	**0.0**	**4.4**
	5	1	50	0.1	1.7	0.0(5)	0.0	0.3	1549.1	37.7(1)	0.0	82.1	1488.2	36.2(1)	0.0	**0.1**	**0.6**	**0.0(5)**	**0.0**	**0.3**
			60	**0.2**	**2.1**	**0.0(5)**	**0.0**	**0.5**	1800.6	50.4(0)	8.8	45.4	1806.5	44.6(0)	7.3	0.1	16.3	0.0(5)	0.0	0.5
			70	**0.2**	**1.9**	**0.0(5)**	**0.0**	**0.0**	1802.9	38.6(0)	7.2	31.0	1803.4	41.2(0)	8.6	0.2	3.6	0.0(5)	0.0	0.0
		2	50	**0.1**	**0.4**	**0.0(5)**	**0.0**	**1.1**	15.7	0.0(5)	0.0	81.5	10.5	0.0(5)	0.0	**0.1**	**0.4**	**0.0(5)**	**0.0**	**1.1**
			60	0.1	1.1	0.0(5)	0.0	0.3	499.0	0.2(4)	0.0	47.6	234.3	0.0(5)	0.0	**0.1**	**0.7**	**0.0(5)**	**0.0**	**0.3**
			70	**0.1**	**1.5**	**0.0(5)**	**0.0**	**0.0**	1468.1	4.6(1)	0.1	27.6	1202.5	2.9(2)	0.1	0.1	2.0	0.0(5)	0.0	0.0
		4	50	**0.1**	**0.1**	**0.0(5)**	**0.0**	**1.2**	0.2	0.0(5)	0.0	63.4	0.2	0.0(5)	0.0	**0.1**	**0.1**	**0.0(5)**	**0.0**	**1.3**
			60	**0.1**	**0.2**	**0.0(5)**	**0.0**	**0.0**	0.8	0.0(5)	0.0	54.1	0.7	0.0(5)	0.0	**0.1**	**0.2**	**0.0(5)**	**0.0**	**0.0**
			70	0.1	1.0	0.0(5)	0.0	1.5	39.4	0.0(5)	0.0	27.1	13.5	0.0(5)	0.0	**0.1**	**0.8**	**0.0(5)**	**0.0**	**1.5**

Table 5
Number of constraints, variables, and nodes visited in the branching tree of formulations (Flow-Cov), (Path), (Path) + VI, (Path-Cov) on graph30 and graph40.

	(Flow-Cov)				(Path)				(Path) + VI				(Path-Cov)			
	c	v	bv	nodes	c	v	bv	nodes	c	v	bv	nodes	c	v	bv	nodes
graph30	8551.5	6754.0	3441.2	0.6	716.1	382.3	245.2	1124757.3	918.3	382.3	245.2	607817.9	2632.1	588.2	451.1	2357.8
graph40	25549.9	20155.2	10180.8	9.5	1152.4	611.7	395.1	2661373.2	1497.2	611.7	395.1	1724565.8	5163.7	955.7	739.2	107560.3

Table 6
Performance of formulations (Flow-Cov), (Path-Cov), and (Path-Cov) + VI on graph100 and graph120.

Data	$B\%$	p	$R\%$	(Flow-Cov)					(Path-Cov)					(Path-Cov) + VI			
				t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$
graph100, $\|V\| = 100, \|E\| = 4950$	0.5	1	50	**6.8**	**1512.9**	**5.7(1)**	**0.0**	**6.9**	6.8	1807.1	10.6(0)	3.2	7.0	6.8	1564.4	7.9(1)	0.8
			60	**7.1**	**1716.5**	**1.9(1)**	**0.0**	**2.4**	7.3	1807.5	3.3(0)	0.8	2.4	7.2	1807.5	2.6(0)	0.2
			70	**8.1**	**1770.6**	**1649.1(1)**	**14.7**	**3.2**	7.9	1808.3	4.4(0)	1.1	3.3	7.9	1808.4	4.8(0)	1.4
		5	50	1.2	12.0	0.0(5)	0.0	1.6	1.2	7.5	0.0(5)	0.0	1.7	**1.2**	**5.7**	**0.0(5)**	**0.0**
			60	**1.3**	**455.3**	**0.3(4)**	**0.0**	**4.8**	1.3	710.2	0.0(4)	0.0	5.1	1.3	654.6	0.0(4)	0.0
			70	**1.6**	**624.0**	**0.0(4)**	**0.0**	**2.5**	1.6	1101.1	1.5(2)	0.3	2.5	1.6	1096.8	1.5(2)	0.5
		10	50	1.1	2.0	0.0(5)	0.0	2.8	**1.1**	**1.4**	**0.0(5)**	**0.0**	**3.8**	**1.1**	**1.4**	**0.0(5)**	**0.0**
			60	1.2	8.4	0.0(5)	0.0	3.5	1.2	3.0	0.0(5)	0.0	4.0	**1.2**	**2.1**	**0.0(5)**	**0.0**
			70	1.2	19.7	0.0(5)	0.0	3.3	1.2	6.5	0.0(5)	0.0	3.6	**1.2**	**6.2**	**0.0(5)**	**0.0**
	1	1	50	7.2	1285.7	3.3(2)	0.0	3.6	**7.0**	**1262.4**	**4.1(2)**	**0.4**	**3.7**	6.9	1265.4	3.7(2)	0.0
			60	**7.1**	**1383.9**	**0.6(3)**	**0.0**	**0.6**	7.4	1557.2	1.5(2)	0.8	0.6	7.3	1378.9	0.9(2)	0.3
			70	**8.1**	**1643.4**	**16.3(2)**	**11.1**	**0.9**	8.1	1692.6	1.6(1)	0.7	0.9	8.1	1710.9	2.0(1)	1.1
		5	50	1.2	16.0	0.0(5)	0.0	1.3	**1.2**	**6.7**	**0.0(5)**	**0.0**	**1.3**	1.2	6.9	0.0(5)	0.0
			60	**1.3**	**413.9**	**0.3(4)**	**0.0**	**3.9**	1.3	473.8	0.3(4)	0.0	4.1	1.3	448.4	0.3(4)	0.0
			70	**1.6**	**244.2**	**0.0(5)**	**0.0**	**1.8**	1.7	1091.4	0.8(2)	0.1	1.9	1.6	1087.7	1.5(2)	0.4
		10	50	1.1	1.9	0.0(5)	0.0	2.8	1.1	**1.4**	**0.0(5)**	**0.0**	**3.8**	**1.1**	**1.4**	**0.0(5)**	**0.0**
			60	1.2	8.7	0.0(5)	0.0	3.5	1.2	2.8	0.0(5)	0.0	3.7	**1.2**	**2.2**	**0.0(5)**	**0.0**
			70	1.2	22.1	0.0(5)	0.0	2.7	1.2	14.8	0.0(5)	0.0	2.8	**1.2**	**14.1**	**0.0(5)**	**0.0**
	5	1	50	6.9	747.2	0.0(5)	0.0	0.0	7.1	284.7	0.0(5)	0.0	0.0	**7.0**	**278.3**	**0.0(5)**	**0.0**
			60	7.3	1064.1	0.0(5)	0.0	0.0	7.3	432.2	0.0(5)	0.0	0.0	**7.2**	**368.9**	**0.0(5)**	**0.0**
			70	7.9	1148.0	0.0(5)	0.0	0.0	**8.1**	**513.4**	**0.0(5)**	**0.0**	**0.0**	8.1	632.4	0.0(5)	0.0
		5	50	1.2	4.0	0.0(5)	0.0	0.0	**1.2**	**1.7**	**0.0(5)**	**0.0**	**0.0**	**1.2**	**1.7**	**0.0(5)**	**0.0**
			60	1.3	8.6	0.0(5)	0.0	0.0	**1.3**	**4.9**	**0.0(5)**	**0.0**	**0.0**	1.2	5.3	0.0(5)	0.0
			70	**1.7**	**21.1**	**0.0(5)**	**0.0**	**0.1**	1.7	403.0	0.0(4)	0.0	0.1	1.6	22.9	0.0(5)	0.0
		10	50	1.1	1.7	0.0(5)	0.0	0.9	1.2	1.4	0.0(5)	0.0	1.0	**1.1**	**1.3**	**0.0(5)**	**0.0**
			60	1.2	3.3	0.0(5)	0.0	0.3	**1.2**	**1.5**	**0.0(5)**	**0.0**	**0.3**	**1.1**	**1.5**	**0.0(5)**	**0.0**
			70	1.2	7.6	0.0(5)	0.0	0.5	1.2	4.7	0.0(5)	0.0	0.5	**1.2**	**5.2**	**0.0(5)**	**0.0**

(*continued on next page*)

Table 6 (*continued*)

Data	$B\%$	p	$R\%$	(Flow-Cov)					(Path-Cov)					(Path-Cov) + VI			
				t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$
graph120, $\|V\| = 120$, $\|E\| = 7140$	0.5	1	50	**10.6**	**1718.2**	**6503.6(1)**	**16.9**	**3.3**	10.8	1820.2	3.6(0)	0.3	3.3	10.9	1811.3	4.2(0)	0.8
			60	12.2	1814.7	9004.2(0)	24.8	3.3	12.1	1812.6	4.0(0)	0.6	3.3	12.2	1812.5	7.5(0)	3.3
			70	13.3	1815.4	10993.9(0)	28.8	3.4	13.4	1814.1	3.6(0)	0.1	3.4	13.1	1814.0	4.7(0)	1.2
		6	50	2.1	462.0	0.7(4)	0.0	4.5	**2.1**	**396.3**	**0.5(4)**	**0.0**	**4.7**	2.1	405.6	1.1(4)	0.1
			60	**2.2**	**1353.5**	**0.1(3)**	**0.0**	**3.5**	2.2	1247.1	0.8(2)	0.1	3.6	2.2	1309.7	0.8(2)	0.1
			70	**3.9**	**1248.9**	**0.9(2)**	**0.0**	**3.7**	3.9	1592.4	5.(1)	1.7	3.8	3.9	1744.7	4.9(1)	1.7
		12	50	2.0	4.2	0.0(5)	0.0	1.7	**2.0**	**2.3**	**0.0(5)**	**0.0**	**2.2**	**2.0**	**2.3**	**0.0(5)**	**0.0**
			60	2.0	76.1	0.0(5)	0.0	5.1	2.0	6.1	0.0(5)	0.0	5.7	**2.0**	**6.0**	**0.0(5)**	**0.0**
			70	2.0	355.5	0.0(5)	0.0	3.4	2.1	215.1	0.0(5)	0.0	3.8	**2.0**	**65.4**	**0.0(5)**	**0.0**
	1	1	50	10.7	1798.6	2019.5(1)	18.4	0.5	11.0	1796.0	1.0(1)	0.4	0.5	**10.9**	**1768.8**	**0.8(1)**	**0.3**
			60	12.3	1815.1	9023.2(0)	30.3	1.7	12.3	1719.2	2.2(1)	0.5	1.7	**12.3**	**1728.1**	**2.2(2)**	**0.4**
			70	13.2	1815.4	11005.2(0)	30.0	1.8	13.7	1803.5	1.9(1)	0.1	1.8	**13.7**	**1734.1**	**2.3(2)**	**0.6**
		6	50	2.1	456.6	0.5(4)	0.0	3.9	2.1	529.4	0.5(4)	0.0	4.1	**2.1**	**412.0**	**0.8(4)**	**0.2**
			60	**2.2**	**1076.9**	**0.1(3)**	**0.0**	**3.0**	2.2	1274.0	0.9(2)	0.1	3.1	2.2	1323.6	0.8(2)	0.1
			70	**4.1**	**1480.6**	**1.4(2)**	**0.0**	**3.8**	4.1	1806.9	5.3(0)	1.9	3.9	4.1	1655.6	5.5(1)	1.9
		12	50	2.0	4.2	0.0(5)	0.0	1.7	2.0	2.3	0.0(5)	0.0	2.2	**1.9**	**2.2**	**0.0(5)**	**0.0**
			60	2.0	263.1	0.0(5)	0.0	5.3	2.0	7.0	0.0(5)	0.0	5.6	**2.0**	**6.2**	**0.0(5)**	**0.0**
			70	2.1	780.9	0.2(4)	0.0	3.2	2.0	380.8	0.2(4)	0.0	3.6	**2.0**	**335.9**	**0.0(5)**	**0.0**
	5	1	50	10.8	1530.5	2019.3(2)	18.5	0.0	**10.9**	**367.1**	**0.0(5)**	**0.0**	**0.0**	11.1	506.9	0.0(5)	0.0
			60	12.4	1816.0	7167.1(0)	31.3	0.0	12.5	971.9	5.8(4)	4.5	0.0	**12.2**	**776.0**	**0.0(5)**	**0.0**
			70	13.2	1817.0	6993.(0)	31.4	0.0	**13.6**	**985.4**	**0.0(5)**	**0.0**	**0.0**	13.6	1057.8	0.0(5)	0.0
		6	50	2.1	30.9	0.0(5)	0.0	0.2	**2.1**	**5.3**	**0.0(5)**	**0.0**	**0.2**	2.1	7.7	0.0(5)	0.0
			60	2.3	102.7	0.0(5)	0.0	0.4	**2.2**	**44.7**	**0.0(5)**	**0.0**	**0.4**	2.2	48.6	0.0(5)	0.0
			70	**4.0**	**281.8**	**0.0(5)**	**0.0**	**0.4**	4.1	1484.1	0.4(1)	0.1	0.4	4.1	1451.1	0.2(1)	0.0
		12	50	2.0	3.6	0.0(5)	0.0	0.6	**2.0**	**2.2**	**0.0(5)**	**0.0**	**0.7**	**2.0**	**2.2**	**0.0(5)**	**0.0**
			60	2.0	24.3	0.0(5)	0.0	1.5	2.0	4.4	0.0(5)	0.0	1.6	**2.0**	**4.3**	**0.0(5)**	**0.0**
			70	2.0	52.5	0.0(5)	0.0	0.7	2.0	10.3	0.0(5)	0.0	0.7	**2.0**	**10.0**	**0.0(5)**	**0.0**

Table 7
Number of constraints, variables, and nodes visited in the branching tree of formulations (Flow-Cov), (Path-Cov), and (Path-Cov) + VI on graph100 and graph120.

	(Flow-Cov)				(Path-Cov)				(Path-Cov) + VI			
	c	v	bv	nodes	c	v	bv	nodes	c	v	bv	nodes
graph100	796249.9	632308.2	316639.7	575.3	55293.7	4581.3	3605.5	95947.3	55310.7	4581.3	3605.5	71610.8
graph120	1556268.4	1236714.5	619029.0	1776.1	88816.1	6384.0	5035.1	166633.4	88847.8	6384.0	5035.1	154253.2

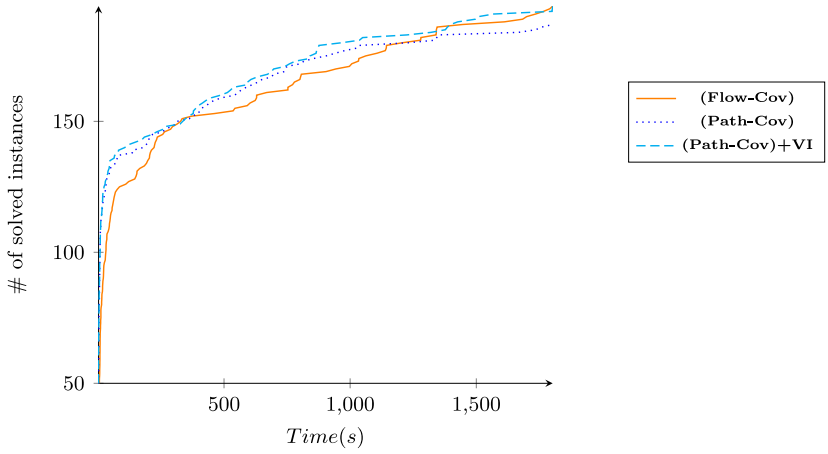


Fig. 5. Performance profile graph of #solved instances using (Flow-Cov), (Path-Cov), and (Path-Cov) + VI formulations on graph100 and graph120.

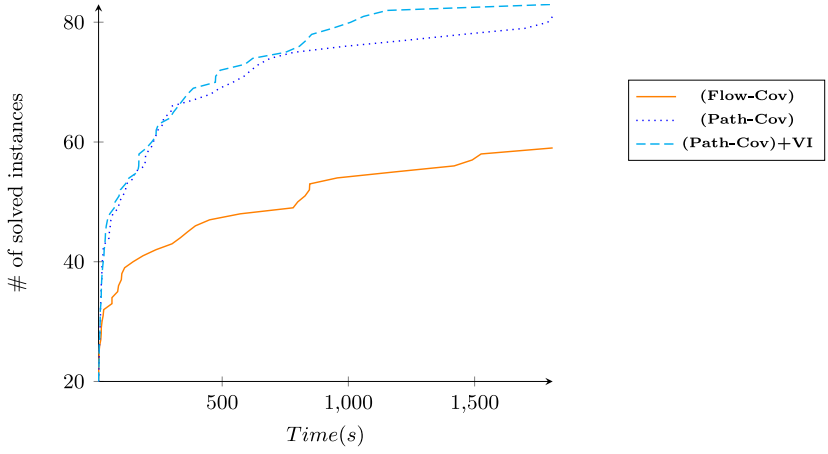


Fig. 6. Performance profile graph of #solved instances using (Flow-Cov), (Path-Cov), and (Path-Cov) + VI formulations on pmedb dataset.

limit but (Path-Cov) + VI did and vice versa. Nevertheless, (Path-Cov) + VI found the optimal solution in more instances than (Path-Cov) and all the instances solved to optimality by (Path-Cov) were also solved to optimality by (Path-Cov) + VI. It can be appreciated that the average preprocessing time is lower than fourteen seconds in all cases. Furthermore, the linear relaxation of the formulations (G^t_{LP}%) is quite good, being on average 2.1% for (Flow-Cov) and 2.3% for (Path-Cov) and (Path-Cov) + VI.

In view of the results reported in this subsection, we conclude that the best formulations for solving Up-MCLP on complete graphs are (Flow-Cov) and (Path-Cov) + VI. In the next subsection, sparser graphs will be analysed.

6.4. Results on sparse graphs

The aim of this subsection is to compare the proposed formulations on sparse graphs. In particular, we used the uncapacited p-median datasets from the OR-Library. For interested readers, non-parametric tests (Friedman test and Post-Hoc Holland adjust) to assess the statistical significance of the comparison among the different formulations can be found in supplementary material. They show that there are significant differences between the formulations presented.

Table 8 has the same structure as Table 4. In this table, we reported the results of formulations (Flow-Cov), (Path), (Path) + VI, (Path-Cov), and (Path-Cov) + VI on the smallest pmed datasets (pmed1-pmed5), named pmeds. These networks contain 100 nodes and 195.2 edges on average (the smallest have 190 and the largest 198). The provided results are the average over the five datasets where the other parameters were randomly generated following the procedure described in Section 6.1. Moreover, a similar analysis to Table 7 is reported in Table 9 for pmeds instances. A detailed table can be found in the supplementary material.

Similarly to complete graphs, it can be seen in Table 8 that the difficulty of the instances is highly parameter-dependent. In addition, it is shown that the preprocessing time is small (less than two seconds in all instances). Furthermore, the number of instances solved to optimality by (Flow-Cov), (Path-Cov), and (Path-Cov) + VI is considerably higher than the ones solved by (Path) and (Path) + VI. Observe that the average of the linear relaxation gaps

Table 8
Performance of formulations (Flow-Cov), (Path), (Path) + VI, (Path-Cov), and (Path-Cov) + VI on pmeds.

Data	B%	p	R%	(Flow-Cov)					(Path)				(Path) + VI			(Path-Cov)					(Path-Cov) + VI			
				t_{st}	t_{total}	G%	G^t_{BS}%	G^t_{LP}%	t_{total}	G%	G^t_{BS}%	G^t_{LP}%	t_{total}	G%	G^t_{BS}%	t_{st}	t_{total}	G%	G^t_{BS}%	G^t_{LP}%	t_{st}	t_{total}	G%	G^t_{BS}%
pmeds, $\|V\| =$	0.5	1	50	1.4	26.1	0.0(5)	0.0	4.3	918.8	17.5(4)	2.3	87.5	474.8	23.1(4)	4.5	**1.4**	**20.5**	**0.0(5)**	**0.0**	**4.3**	1.4	24.2	0.0(5)	0.0
100, $\|E\| = 195.2$			60	**1.5**	**15.5**	**0.0(5)**	**0.0**	**2.1**	1764.7	39.5(1)	8.7	55.3	1168.2	31.9(3)	7.8	1.5	38.7	0.0(5)	0.0	2.1	1.5	25.7	0.0(5)	0.0
			70	**1.5**	**32.6**	**0.0(5)**	**0.0**	**4.0**	1801.9	44.0(0)	8.9	35.8	1580.2	75.0(1)	16.9	1.5	991.7	1.5(3)	0.0	4.1	1.5	572.9	0.3(4)	0.0
		5	50	**1.2**	**1.4**	**0.0(5)**	**0.0**	**1.1**	14.5	0.0(5)	0.0	75.8	13.5	0.0(5)	0.0	**1.2**	**1.4**	**0.0(5)**	**0.0**	**2.7**	**1.2**	**1.4**	**0.0(5)**	**0.0**
			60	**1.2**	**1.8**	**0.0(5)**	**0.0**	**1.3**	575.0	0.0(5)	0.0	57.9	589.1	0.0(5)	0.0	1.2	2.1	0.0(5)	0.0	3.4	1.2	2.3	0.0(5)	0.0
			70	1.2	3.9	0.0(5)	0.0	1.7	803.4	3.2(3)	0.0	38.8	979.5	2.5(3)	0.0	1.2	3.9	0.0(5)	0.0	2.6	**1.2**	**3.7**	**0.0(5)**	**0.0**
		10	50	1.2	1.2	0.0(5)	0.0	0.1	0.8	0.0(5)	0.0	42.8	**0.7**	**0.0(5)**	**0.0**	1.1	1.2	0.0(5)	0.0	3.7	1.1	1.2	0.0(5)	0.0
			60	**1.2**	**1.2**	**0.0(5)**	**0.0**	**0.7**	1.6	0.0(5)	0.0	46.9	1.8	0.0(5)	0.0	1.1	1.3	0.0(5)	0.0	2.9	1.1	1.3	0.0(5)	0.0
			70	**1.2**	**1.4**	**0.0(5)**	**0.0**	**1.6**	9.2	0.0(5)	0.0	35.4	28.2	0.0(5)	0.0	1.2	1.5	0.0(5)	0.0	3.0	1.2	1.6	0.0(5)	0.0
	1	1	50	1.7	570.6	0.0(5)	0.0	7.2	1496.5	31.6(1)	2.7	77.1	1137.1	83.1(3)	13.1	1.7	202.6	0.0(5)	0.0	7.3	**1.6**	**68.7**	**0.0(5)**	**0.0**
			60	1.7	1460.1	2.4(2)	0.0	7.7	1800.6	40.7(0)	4.9	49.6	1488.5	38.2(1)	6.7	1.7	960.6	3.1(4)	1.3	7.8	**1.8**	**850.1**	**3.1(4)**	**1.3**
			70	**1.8**	**425.1**	**0.0(5)**	**0.0**	**4.8**	1801.2	29.3(0)	5.7	30.1	1800.7	349.5(0)	51.1	1.8	1425.3	2.7(2)	0.0	4.9	1.8	1513.8	3.8(1)	0.0
		5	50	1.2	1.8	0.0(5)	0.0	2.8	84.1	0.0(5)	0.0	75.7	77.7	0.0(5)	0.0	1.2	1.8	0.0(5)	0.0	4.9	**1.2**	**1.7**	**0.0(5)**	**0.0**
			60	**1.2**	**3.4**	**0.0(5)**	**0.0**	**5.0**	918.9	0.7(4)	0.1	55.8	1116.2	3.9(2)	0.0	1.2	22.7	0.0(5)	0.0	6.6	1.3	18.2	0.0(5)	0.0
			70	1.3	62.1	0.0(5)	0.0	5.2	1718.3	8.3(1)	0.2	36.3	1602.7	9.6(1)	0.3	1.3	76.6	0.0(5)	0.0	5.3	**1.3**	**39.9**	**0.0(5)**	**0.0**
		10	50	1.1	1.2	0.0(5)	0.0	1.5	0.8	0.0(5)	0.0	43.1	**0.7**	**0.0(5)**	**0.0**	1.1	1.2	0.0(5)	0.0	4.4	1.1	1.2	0.0(5)	0.0
			60	**1.2**	**1.3**	**0.0(5)**	**0.0**	**1.7**	2.6	0.0(5)	0.0	46.3	2.4	0.0(5)	0.0	1.2	1.9	0.0(5)	0.0	4.2	1.2	1.7	0.0(5)	0.0
			70	**1.2**	**1.7**	**0.0(5)**	**0.0**	**3.2**	18.0	0.0(5)	0.0	34.6	25.3	0.0(5)	0.0	1.2	2.6	0.0(5)	0.0	5.6	1.2	2.5	0.0(5)	0.0
	5	1	50	1.5	1015.6	0.0(5)	0.0	3.0	1288.4	24.(2)	2.7	55.1	1155.6	25.7(2)	3.2	**1.5**	**64.8**	**0.0(5)**	**0.0**	**3.0**	1.5	85.5	0.0(5)	0.0
			60	1.6	1407.8	3.4(2)	0.5	4.5	1800.6	25.1(0)	4.8	33.7	1690.5	27.4(1)	6.2	**1.6**	**869.4**	**2.5(3)**	**0.1**	**4.6**	1.6	1140.9	2.5(2)	0.1
			70	1.6	1205.0	1.8(2)	0.0	2.7	1801.7	22.6(0)	5.7	19.3	1758.5	268.6(1)	35.2	1.6	1077.4	1.8(3)	0.0	2.8	**1.7**	**946.9**	**1.9(3)**	**0.0**
		5	50	**1.2**	**2.3**	**0.0(5)**	**0.0**	**4.1**	102.8	0.0(5)	0.0	66.8	132.1	0.0(5)	0.0	1.2	3.0	0.0(5)	0.0	4.2	1.2	2.9	0.0(5)	0.0
			60	**1.2**	**5.2**	**0.0(5)**	**0.0**	**2.7**	686.2	0.3(4)	0.0	46.2	738.7	0.0(5)	0.0	1.2	21.0	0.0(5)	0.0	2.7	1.2	17.6	0.0(5)	0.0
			70	1.3	207.3	0.0(5)	0.0	3.0	1418.4	2.8(3)	0.1	28.8	1458.9	4.2(2)	0.0	1.3	48.5	0.0(5)	0.0	3.0	**1.3**	**48.1**	**0.0(5)**	**0.0**
		10	50	1.2	1.2	0.0(5)	0.0	0.8	1.0	0.0(5)	0.0	38.4	**0.8**	**0.0(5)**	**0.0**	1.1	1.3	0.0(5)	0.0	1.1	1.2	1.3	0.0(5)	0.0
			60	**1.2**	**1.5**	**0.0(5)**	**0.0**	**3.1**	4.3	0.0(5)	0.0	40.7	4.6	0.0(5)	0.0	1.2	1.9	0.0(5)	0.0	4.0	1.2	1.8	0.0(5)	0.0
			70	**1.2**	**2.2**	**0.0(5)**	**0.0**	**2.9**	61.8	0.0(5)	0.0	29.4	38.3	0.0(5)	0.0	1.2	4.1	0.0(5)	0.0	3.0	1.2	3.9	0.0(5)	0.0

Table 9
Number of constraints, variables, and nodes visited in the branching tree of formulations (Flow-Cov), (Path), (Path) + VI, (Path-Cov), and (Path-Cov) + VI on pmeds.

	(Flow-Cov)				(Path)				(Path) + VI				(Path-Cov)				(Path-Cov) + VI			
Data	c	v	bv	nodes	c	v	bv	nodes	c	v	bv	nodes	c	v	bv	nodes	c	v	bv	nodes
pmeds	166204.5	120362.9	60983.3	412.4	1023.2	606.4	376.0	1296509.9	2333.6	606.4	376.0	786929.8	7368.9	2248.1	2017.7	29238.2	7396.4	2248.1	2017.7	25113.6

Table 10
Performance of formulations (Flow-Cov), (Path-Cov), and (Path-Cov) + VI on pmedb.

Data	$B\%$	p	$R\%$	(Flow-Cov)					(Path-Cov)					(Path-Cov) + VI			
				t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$	$G^t_{LP}\%$	t_{st}	t_{total}	$G\%$	$G^t_{BS}\%$
pmedb, $\|V\| = 200$, $\|E\| = 777.8$	0.5	1	50	12.5	1815.3	5881.2(0)	18.7	9.1	12.4	1812.9	9.3(0)	0.1	9.9	12.3	1812.7	8.9(0)	0.0
			60	18.3	1821.2	6383.5(0)	33.1	7.4	18.5	1819.1	7.5(0)	0.0	8.0	18.4	1818.7	7.7(0)	0.1
			70	36.3	1839.9	13701.3(0)	24.8	5.8	37.6	1848.8	5.8(0)	0.0	6.4	37.1	1837.5	6.1(0)	0.3
		10	50	9.3	22.4	0.0(5)	0.0	2.1	**9.2**	**14.8**	**0.0(5)**	**0.0**	**4.3**	9.1	15.3	0.0(5)	0.0
			60	9.5	1451.5	1.2(2)	0.0	4.4	9.3	1150.8	1.1(2)	0.2	6.5	**9.3**	**975.3**	**1.3(3)**	**0.3**
			70	10.0	1811.3	2.8(0)	0.6	3.9	9.8	1815.2	2.7(0)	1.0	4.4	9.8	1811.3	2.6(0)	1.0
		20	50	**8.9**	**9.1**	**0.0(5)**	**0.0**	**1.8**	8.9	9.2	0.0(5)	0.0	3.6	**8.8**	**9.1**	**0.0(5)**	**0.0**
			60	9.0	45.4	0.0(5)	0.0	2.5	**9.0**	**13.3**	**0.0(5)**	**0.0**	**3.8**	8.9	14.4	0.0(5)	0.0
			70	9.4	785.8	0.1(4)	0.1	2.7	**9.2**	**617.7**	**0.4(4)**	**0.0**	**5.2**	9.2	771.0	0.3(3)	0.0
	1	1	50	11.2	1814.1	6101.7(0)	23.3	8.2	11.2	1811.5	8.4(0)	0.0	8.4	11.1	1811.5	8.4(0)	0.0
			60	16.2	1819.9	13905.8(0)	25.2	7.4	16.3	1819.4	7.5(0)	0.0	7.6	16.1	1816.8	7.5(0)	0.0
			70	33.7	1836.7	17155.8(0)	26.3	5.1	33.7	1834.1	5.4(0)	0.2	5.3	33.5	1834.1	5.5(0)	0.4
		10	50	9.3	299.3	0.0(5)	0.0	3.7	9.0	57.5	0.0(5)	0.0	4.7	**9.0**	**53.7**	**0.0(5)**	**0.0**
			60	9.4	1810.0	3.2(0)	0.1	5.5	9.5	1566.1	2.2(1)	0.5	5.8	**9.3**	**1526.1**	**1.7(1)**	**0.2**
			70	9.5	1811.1	3.9(0)	0.7	3.8	9.6	1811.5	2.7(0)	0.4	3.8	9.5	1810.6	4.1(0)	1.9
		20	50	9.2	9.5	0.0(5)	0.0	3.2	9.1	9.4	0.0(5)	0.0	4.2	**8.9**	**9.3**	**0.0(5)**	**0.0**
			60	9.0	141.0	0.0(5)	0.0	3.6	9.3	23.2	0.0(5)	0.0	4.5	**9.0**	**20.7**	**0.0(5)**	**0.0**
			70	9.3	1613.6	1.0(1)	0.1	3.2	9.2	1158.1	0.5(3)	0.0	4.1	**9.1**	**959.8**	**0.3(4)**	**0.0**
	5	1	50	11.2	1546.2	6102.5(2)	25.3	0.0	**11.2**	**181.1**	**0.0(5)**	**0.0**	**0.0**	11.0	185.9	0.0(5)	0.0
			60	16.6	1742.8	6551.2(1)	24.6	0.1	16.0	917.2	0.1(3)	0.0	0.1	**16.0**	**803.1**	**0.1(4)**	**0.0**
			70	33.2	1837.5	11417.1(0)	29.4	0.2	32.6	853.6	0.2(4)	0.0	0.2	**32.6**	**769.9**	**0.2(4)**	**0.0**
		10	50	9.1	342.8	0.0(5)	0.0	1.7	9.2	34.8	0.0(5)	0.0	1.7	**9.0**	**30.0**	**0.0(5)**	**0.0**
			60	9.6	1527.2	1.1(1)	0.2	1.7	9.2	696.4	0.0(5)	0.0	1.7	**9.2**	**434.5**	**0.0(5)**	**0.0**
			70	9.5	1811.4	1.2(0)	0.8	0.5	9.5	733.6	0.2(4)	0.0	0.5	**9.4**	**687.4**	**0.5(4)**	**0.3**
		20	50	**9.0**	**9.4**	**0.0(5)**	**0.0**	**1.2**	8.9	9.6	0.0(5)	0.0	1.3	8.8	9.5	0.0(5)	0.0
			60	9.2	44.5	0.0(5)	0.0	1.3	9.0	17.1	0.0(5)	0.0	1.4	**8.9**	**15.5**	**0.0(5)**	**0.0**
			70	9.3	1324.6	0.4(2)	0.0	0.8	9.1	730.6	0.0(4)	0.0	0.8	**9.3**	**493.2**	**0.1(4)**	**0.0**

Table 11
Number of constraints, variables, and nodes visited in the branching tree of formulations (Flow-Cov), (Path-Cov), and (Path-Cov) + VI on pmedb.

	(Flow-Cov)				(Path-Cov)				(Path-Cov) + VI			
Data	c	v	bv	nodes	c	v	bv	nodes	c	v	bv	nodes
pmedb	2147283.6	1599445.8	802394.7	1809.3	35199.8	6849.2	6350.2	97218.6	35336.9	6849.2	6350.2	80667.2

of (Flow-Cov) (3.1%), (Path-Cov) and (Path-Cov) + VI (4.0%) are better than the linear relaxation gap of (Path) and (Path) + VI (47.5%). In Table 9 the different sizes of the problem for each formulation can be appreciated.

In Table 10, we report the result obtained on datasets of larger size, named pmedb. More concretely, the table shows the results of formulations (Flow-Cov), (Path-Cov), and (Path-Cov) + VI on larger pmed datasets (pmed6-pmed10). These networks contain 200 nodes and 777.8 edges on average (the smallest have 774 and the largest 785). Note that the results of (Path), (Path) + VI are not reported because very few instances were solved to optimality. In Table 11, a similar analysis to Table 9 is depicted. A detailed table can be found in the supplementary material. In Fig. 6, the performance profile of the number of solved instances using these formulations is depicted.

As can be appreciated in Table 10 and Fig. 6, (Path-Cov) + VI outperforms (Flow-Cov) and (Path-Cov). Observe that although the gap of the linear relaxation of (Flow-Cov) (3.4%) is smaller than the gap of (Path-Cov) and (Path-Cov) + VI (4%), the latter is the one that solves to optimality within the time limit the largest number of instances.

Based on the results presented in this subsection, we conclude that the best formulation for solving Up-MCLP on pmedian graphs was (Path-Cov) + VI. Furthermore, the results included in this subsection show the usefulness of including the valid inequalities discussed in the paper.

7. Conclusions and outlook

In this paper, we have tackled an interesting problem: the upgrading maximal covering location problem with edge length modifications, Up-MCLP. As far as we know, it is the first time that this problem is discussed in the literature.

Since we were dealing with a new problem, we proposed three different mixed-integer formulations to model the situation from various perspectives. Moreover, we developed an effective preprocessing phase, which fixed many variables and reduced the size of the problem considerably. Then, for each formulation, we provided several sets of valid inequalities. These constraints allowed us to strengthen the formulations and to reduce the symmetries contained in the problem, shortening the time to solve the formulations. The performance of the three formulations and the improvement provided by the preprocessing phase and the valid inequalities can be appreciated in the computational results included in the paper. In these experiments, it can be seen that the most efficient formulations for solving Up-MCLP are (Flow-Cov) and (Path-Cov). In complete graphs, there is little difference in performance between these two formulations, while in sparse graphs (Path-Cov) performs better than (Flow-Cov). In both types of graphs, the addition of valid inequalities allows us to optimally solve a larger number of instances within the time limit.

We believe that this paper is an encouraging starting point that opens up many opportunities for further research. While our preprocessing techniques and valid inequalities allow us to solve significantly larger instances, and in shorter time, the size of the solvable instances still falls short of what you would encounter in practice. This leads to the necessity of developing heuristics that can obtain good solutions (even if they are not optimal) in shorter time than the exact algorithms. An additional line of future work is to further study and develop formulations for other upgrading versions of location problems. For example, interesting and similar problems could be obtained by modifying the covering criterion (e.g. gradual coverage, cooperative coverage, etc.), the location criterion (e.g. center problems, set covering problems, etc.), and the upgrading assumptions (e.g. non-linear upgrading cost or additional requirements on the sets of edges to be reduced, e.g., they have to form a connected set).

Acknowledgments

Marta Baldomero-Naranjo was partially supported by research project PID2020-114594GB-C22 (Agencia Estatal de Investigación, Spain and the European Regional Development's funds (ERDF)), research projects FEDER-UCA18-106895 and P18-FR-1422 (Regional Government of Andalucía, Spain and ERDF), the BritishSpanish Society Scholarship Programme 2018 (Telefónica and the BritishSpanish Society), and PhD grant UCA/REC01VI/2017 (Universidad de Cádiz).

Jörg Kalcsics was partially supported by research project PID2020-114594GB-C22 (Agencia Estatal de Investigación, Spain and ERDF) and Grant number EST2019-047 for research stays in Universidad de Cádiz (Plan Propio de Investigación y Transferencia de la UCA).

Alfredo Marín was partially supported by research project PID2019-110886RB-100 (Ministerio de Ciencia e Innovación, Spain). Part of this research were conducted while the author was on sabbatical at the University of Cádiz.

Antonio M. Rodríguez-Chía was partially supported by research project PID2020-114594GB-C22 (Agencia Estatal de Investigación, Spain and ERDF), research projects FEDER-UCA18-106895 and P18-FR-1422 (Regional Government of Andalucía, Spain and ERDF), and research project NetmeetData (Fundación BBVA).

The authors would like to thank the support of the Spanish Ministry of Science and Innovation through project RED2018-102363-T and the anonymous reviewers for their comments and suggestions.

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.ejor.2022.02.001.

References

Afrashteh, E., Alizadeh, B., & Baroughi, F. (2020). Optimal approaches for upgrading selective obnoxious p-median location problems on tree networks. *Annals of Operations Research, 289*(2), 153–172.

Alizadeh, B., & Etemad, R. (2016). Linear time optimal approaches for reverse obnoxious center location problems on networks. *Optimization, 65*(11), 2025–2036.

Álvarez-Miranda, E., & Sinnl, M. (2017). Lagrangian and branch-and-cut approaches for upgrading spanning tree problems. *Computers & Operations Research, 83*, 13–27.

Arana-Jiménez, M., Blanco, V., & Fernández, E. (2020). On the fuzzy maximal covering location problem. *European Journal of Operational Research, 283*(2), 692–705.

Avella, P., Boccia, M., & Vasilyev, I. (2009). Computational experience with general cutting planes for the set covering problem. *Operations Research Letters, 37*(1), 16–20.

Averbakh, I., Berman, O., Krass, D., Kalcsics, J., & Nickel, S. (2014). Cooperative covering problems on networks. *Networks, 63*(4), 334–349.

Baldomero-Naranjo, M., Kalcsics, J., & Rodríguez-Chía, A. M. (2021). Minmax regret maximal covering location problems with edge demands. *Computers & Operations Research, 130*, 105181.
Bansal, M., & Kianfar, K. (2017). Planar maximum coverage location problem with partial coverage and rectangular demand and service zones. *INFORMS Journal on Computing, 29*(1), 152–169.
Beasley, J. E. (1990). OR-Library: Distributing test problems by electronic mail. *Journal of the Operational Research Society, 41*(11), 1069–1072.
Berman, O., Kalcsics, J., & Krass, D. (2016). On covering location problems on networks with edge demand. *Computers & Operations Research, 74*(C), 214–227.
Berman, O., & Krass, D. (2002). The generalized maximal covering location problem. *Computers & Operations Research, 29*(6), 563–581.
Berman, O., & Wang, J. (2011). The minmax regret gradual covering location problem on a network with incomplete information of demand weights. *European Journal of Operational Research, 208*(3), 233–238.
Blanco, V., & Marín, A. (2019). Upgrading nodes in tree-shaped hub location. *Computers & Operations Research, 102*, 75–90.
Bonab, F. B., Burkard, R. E., & Gassner, E. (2011). Inverse p-median problems with variable edge lengths. *Mathematical Methods of Operations Research, 73*(2), 263–280.
Burkard, R. E., Gassner, E., & Hatzl, J. (2006). A linear time algorithm for the reverse 1-median problem on a cycle. *Networks, 48*(1), 16–23.
Burkard, R. E., Gassner, E., & Hatzl, J. (2008). Reverse 2-median problem on trees. *Discrete Applied Mathematics, 156*(11), 1963–1976.
Burkard, R. E., Lin, Y., & Zhang, J. (2004a). Weight reduction problems with certain bottleneck objectives. *European Journal of Operational Research, 153*(1), 191–199.
Burkard, R. E., Pleschiutschnig, C., & Zhang, J. (2004b). Inverse median problems. *Discrete Optimization, 1*(1), 23–39.
Carrasco, J., García, S., Rueda, M., Das, S., & Herrera, F. (2020). Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm and Evolutionary Computation, 54*, 100665.
Church, R. (1984). The planar maximal covering location problem. *Journal of Regional Science, 24*(2), 185–201.
Church, R., & Meadows, M. (1979). Location modeling utilizing maximum service distance criteria. *Geographical Analysis, 11*, 358–373.
Church, R., & ReVelle, C. (1974). The maximal covering location problem. *Papers of the Regional Science Association, 3*, 101–118.
Cordeau, J.-F., Furini, F., & Ljubić, I. (2019). Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research, 275*(3), 882–896.
Daskin, M. S. (1983). A maximum expected covering location model: Formulation, properties and heuristic solution. *Transportation Science, 17*(1), 48–70.
Demgensky, I., Noltemeier, H., & Wirth, H.-C. (2002). On the flow cost lowering problem. *European Journal of Operational Research, 137*(2), 265–271.
Dilkina, B., Lai, K. J., & Gomes, C. P. (2011). Upgrading shortest paths in networks. In T. Achterberg, & J. C. Beck (Eds.), *Integration of ai and or techniques in constraint programming for combinatorial optimization problems* (pp. 76–91). Berlin, Heidelberg: Springer Berlin Heidelberg.
Ensor, T., & Cooper, S. (2004). Overcoming barriers to health service access: Influencing the demand side. *Health Policy Plan, 19*(2), 69–79.
Espejo, I., Marín, A., & Rodríguez-Chía, A. M. (2012). Closest assignment constraints in discrete location problems. *European Journal of Operational Research, 219*(1), 49–58.
Fröhlich, N., Maier, A., & Hamacher, H. W. (2020). Covering edges in networks. *Networks, 75*(3), 278–290.
García, S., & Marín, A. (2019). Covering location problems. In G. Laporte, S. Nickel, & F. Saldanha da Gama (Eds.), *Location science* (pp. 99–119). Springer International Publishing.
Gassner, E. (2007). Up-and downgrading the 1-median in a network. *Technical Report 2007-6*. Graz University of Technology.
Gassner, E. (2009). Up-and downgrading the 1-center in a network. *European Journal of Operational Research, 198*(2), 370–377.
Gassner, E. (2012). An inverse approach to convex ordered median problems in trees. *Journal of Combinatorial Optimization, 23*(2), 261–273.
Guzmán, V. C., Masegosa, A. D., Pelta, D. A., & Verdegay, J. L. (2016). Fuzzy models and resolution methods for covering location problems: An annotated bibliography. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, 24*(04), 561–591.
Heuberger, C. (2004). Inverse combinatorial optimization: A survey on problems, methods, and results. *Journal of Combinatorial Optimization, 8*(3), 329–361.
(1997). In D. S. Hochbaum (Ed.), *Approximation algorithms for NP-hard problems*. Boston, MA, USA: PWS Publishing Co..
Karatas, M., & Eriskin, L. (2021). The minimal covering location and sizing problem in the presence of gradual cooperative coverage. *European Journal of Operational Research, 295*(3), 838–856.
Marín, A., Martínez-Merino, L. I., Rodríguez-Chía, A. M., & da Gama, F. S. (2018). Multi-period stochastic covering location problems: Modeling framework and solution approach. *European Journal of Operational Research, 268*(2), 432–449.
Melkote, S., & Daskin, M. S. (2001a). Capacitated facility location/network design problems. *European Journal of Operational Research, 129*(3), 481–495.
Melkote, S., & Daskin, M. S. (2001b). An integrated model of facility location and transportation network design. *Transportation Research Part A: Policy and Practice, 35*(6), 515–538.
Nguyen, K. T., & Sepasian, A. R. (2016). The inverse 1-center problem on trees with variable edge lengths under Chebyshev norm and Hamming distance. *Journal of Combinatorial Optimization, 32*(3), 872–884.
Paik, D., & Sahni, S. (1995). Network upgrading problems. *Networks, 26*(1), 45–58.
Plastria, F. (2002). Continuous covering location problems. In Z. Drezner, & H. Hamacher (Eds.), *Facility location: Applications and theory* (pp. 37–79). Springer.
Plastria, F. (2016). Up- and downgrading the Euclidean 1-median problem and knapsack Voronoi diagrams. *Annals of Operations Research, 246*, 227–251.
ReVelle, C., Scholssberg, M., & Williams, J. (2008). Solving the maximal covering location problem with heuristic concentration. *Computers & Operations Research, 35*(2), 427–435.
Sepasian, A. R. (2018). Upgrading the 1-center problem with edge length variables on a tree. *Discrete Optimization, 29*, 1–17.
Sepasian, A. R., & Monabbati, E. (2017). Upgrading min-max spanning tree problem under various cost functions. *Theoretical Computer Science, 704*, 87–91.
Sepasian, A. R., & Rahbarnia, F. (2015). Upgrading p-median problem on a path. *Journal of Mathematical Modelling and Algorithms in Operations Research, 14*(2), 145–157.
Vatsa, A. K., & Jayaswal, S. (2021). Capacitated multi-period maximal covering location problem with server uncertainty. *European Journal of Operational Research, 289*(3), 1107–1126.
Wang, Q., & Bai, Y. (2010). An efficient algorithm for reverse 2-median problem on a cycle. *Journal of Networks, 5*(10), 1169–1176.
Wu, L., Lee, J., Zhang, J., & Wang, Q. (2013). The inverse 1-median problem on tree networks with variable real edge lengths. *Mathematical Problems in Engineering, 2013*, 313868.
Yang, X., & Zhang, J. (2008). Inverse center location problem on a tree. *Journal of Systems Science and Complexity, 21*(4), 643–656.
Zhang, B., Peng, J., & Li, S. (2017). Covering location problem of emergency service facilities in an uncertain environment. *Applied Mathematical Modelling, 51*, 429–447.
Zhang, J., Yang, X., & Cai, M.-C. (1999). Reverse center location problem. In *Algorithms and computation* (pp. 279–294). Springer.
Zhang, Q., Guan, X., & Pardalos, P. M. (2021). Maximum shortest path interdiction problem by upgrading edges on trees under weighted l_1 norm. *Journal of Global Optimization, 79*(4), 959–987.