Omega 119 (2023) 102894

Contents lists available at ScienceDirect

Omega

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

The discrete p-center location problem with upgrading[☆]

Laura Anton-Sanchez[a,b,*], Mercedes Landete[a,b], Francisco Saldanha-da-Gama[c,d]

[a] Departamento de Estadística, Matemáticas e Informática, Universidad Miguel Hernández, Elche, Alicante 03202, Spain
[b] Centro de Investigación Operativa, Universidad Miguel Hernández, Elche, Alicante 03202, Spain
[c] Departamento de Estatística e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, Lisboa 1749-016, Portugal
[d] Centro de Matemática, Aplicações Fundamentais e Investigação Operacional, Faculdade de Ciências, Universidade de Lisboa, Lisboa 1749-016, Portugal

A R T I C L E I N F O

Article history:
Received 1 July 2022
Accepted 2 May 2023
Available online 3 May 2023

Keywords:
Discrete facility location
p-center
Connection upgrading
Center upgrading
MILP models

A B S T R A C T

In this paper, different upgrading strategies are investigated in the context of the p-center problem. The possibility of upgrading a set of connections to different centers is considered as well as the possibility of upgrading entire centers, i.e., all connections made to them. Two variants for these perspectives are analyzed: in the first, there is a limit on the number of connections or centers that can be upgraded; in the second, an existing budget is assumed for the same purpose. Different mixed-integer linear programming models are introduced for those problems as well as data-driven lower and upper bounds. In most cases, an optimal solution can be obtained within an acceptable computing time using an off-the-shelf solver. Nevertheless, this is not the case for one particular family of problems. This motivated the development of a math-heuristic seeking high-quality feasible solutions in that specific case. Extensive computational experiments are reported highlighting the relevance of upgrading connections or centers in the context of the p-center problem.

1. Introduction

Given a set of nodes in a metric space, the p-center problem consists of determining at most p points in such a way that the maximum distance between the given nodes and the closest centers is minimized. This is a minmax problem that has been widely studied [5].

The p-center problem on a network gained much notoriety and momentum with the work by Hakimi [15]. This is a problem that consists of selecting p points (centers) in a network so as to minimize the maximum weighted distance of the nodes of the network to the selected points.

The classical p-center problem and its variants have many applications among which we can point out those in telecommunications, emergency facility location, and logistics. p-Center problems are particularly appropriate for situations when equity is important, as in a disaster management environment (see, e.g. Akgün et al. [1], Dönmez et al. [11], Stienen et al. [25]) or in the context of strategic defense sites (see, e.g., Bell et al. [3]). The interested reader can refer to the overviews provided by Calik et al. [5], Fadda et al. [13] and Wang et al. [26] as well as to the references therein. What is more, the same problem can be used in applications where measures other than distances are of relevance when connecting demand nodes and centers, such as travel times or transportation costs. For this reason, to make our manuscript more general, hereafter we use the term "cost" or "allocation cost" to refer to the measure of interest when connecting a demand node and a center.

Different variants of the p-center problem have been dealt with in the literature triggered by practical needs. In this work, we focus on the case in which all nodes have identical weights and a finite set of possibilities have been identified for locating the centers. This allows casting the problem as a discrete minmax facility location problem, which in turn can be formulated as an integer programming problem.

To the best of the authors' knowledge, the literature on p-center problems assumes that the costs for connecting the demand nodes and the open centers are known beforehand and do not change. Nevertheless, in practice, one may ask whether a better solution can be ultimately achieved if we can somehow compress or reduce beforehand the allocation costs, thus obtaining what could be coined as *upgraded connections*. Such compression can materialize in different ways.

[☆] This manuscript was processed by Associate Editor Prof. Benjamin Lev

[*] Corresponding author at: Departamento de Estadística, Matemáticas e Informática, Universidad Miguel Hernández, Elche, Alicante 03202, Spain.
E-mail address: l.anton@umh.es (L. Anton-Sanchez).

https://doi.org/10.1016/j.omega.2023.102894
0305-0483/

One possibility regards individual connections. For instance, if a connection corresponds to a road, compression may be achieved by redesigning the road (e.g. straightening some curves if possible) or simply by selecting a different trajectory such as a highway instead of a secondary road. In the context of logistics and transportation, a more interesting possibility is to focus on travel time. In this case, changing the transportation mode (e.g. to a faster vehicle) may be a way to decrease the travel time thus upgrading the connection. The reader can refer to the recent paper by Baldomero-Naranjo et al. [2] for other examples of upgrading connections between demand nodes and centers in covering-type facility location problems.

Another possibility for compressing the allocation costs consists of working directly at a center level—*center upgrade*. When a center is upgraded, all the connections to it are considered so as to decrease the allocation cost for all of them. For instance, in the case of mobile centers, a technological upgrade or the use of more skilled human resources may lead to a service being provided faster to all demand nodes allocated to the center.

The possibility of improving some parameter values “before optimizing” to further improve the optimal solutions is not new in Operations Research and Management Science. The best-known case is possibly the compression of execution times in project management and machine scheduling problems (see, e.g. Lamberson and Hocking [21], Shioura et al. [24], Yang [27]). In that case, by assigning more resources to some (critical) activities or jobs it may be possible to reduce their execution time thus reducing the makespan of a project or batch production. Similarly in flight scheduling problems the flight upgrade depends on some timing flexibility indicators (see, e.g. Katsigiannis and Zografos [20]).

In the context of facility location problems, Blanco and Marín [4] investigated cost compression in the so-called tree of hubs location problem [7,8]. The goal is to upgrade hubs (by upgrading all the connections to a hub) to improve the optimal distribution cost. Two enhanced mixed-integer linear programming (MILP) models are derived and empirically compared for the problem. The authors review the literature on connection upgrading in the context of network optimization problems, which include shortest path problems, minimum cost spanning tree problems, and the 1-center problem. The latter is investigated in a network by Sepasian [23]. In all cases, as in the case of the discrete p-center problem that we are investigating in the current paper, the goal is to choose the best-after-changes solution.

In the context of network design and optimization, we also cite the work by Ibaraki et al. [16] who seek to reduce the eccentricity of a network by upgrading some nodes, i.e., reducing the lengths of the edges incident to such nodes. The authors consider separately continuous- and discrete-upgrading strategies.

More recently, Baldomero-Naranjo et al. [2] investigated edge upgrading in the context of maximal covering facility location. Unlike the p-center problem, in which the coverage radius is endogenous, a maximum coverage radius is initially imposed and the goal is to install a certain number of facilities so that the maximum possible demand is covered. For the upgraded version of the problem, the authors propose and compare different mixed-integer programming models.

Throughout this paper, we analyze two perspectives when it comes to upgrading: (i) there is a given number of components (connections or centers) that can be upgraded; (ii) there is a budget that limits the upgrades that can be made.

Upgrading in the context of the p-center problem when possible is actually a means to ensure *a priori* that better service quality will be achieved. By seeking an *upgraded solution* we aim at finding connections or centers whose cost reduction implies an improvement of the system. We should note that other possibilities have been considered in the literature such as positioning the commodities closer to where they are required (see, e.g., Corberán et al. [9] for such a possibility in the context of a minsum facility location problem).

Our work lies in a stream of research aiming at developing models and techniques for extensions of the classical discrete p-center problem triggered by practical needs (see Calik et al. [5], Calik and Tansel [6]). We also refer to Kahr [17], Karatas and Eriskin [18], Pelegrín and Xu [22], and Wang et al. [26] on the role of achieving an optimal demand covering in the context of Logistics problems.

The main contributions of this work can be summarized as follows:

i. Four different extensions of the classical p-center problem are introduced, namely,
upgrading individual connections and upgrading centers, both combined with a maximum number of upgrades or a limited budget for upgrading.
ii. Different optimization models are derived for the above variants.
iii. Lower and upper bounds as well as optimal solution properties are discussed.
iv. A math-heuristic approach is designed and implemented for budget-constrained center upgrading.
v. Extensive computational experiments are conducted. Instances with a number of nodes ranging from 100 to 900 are solved and the results are thoroughly reported, which gives strong evidence that a significant decrease in the optimal covering cost can be achieved through upgrading.

The remainder of this paper is organized as follows. In Section 2 we revisit several modeling aspects related to the discrete p-center problem to ensure a self-contained manuscript. In Section 3 we look into the possibility of upgrading a set of individual connections. In Section 4 we focus on upgrading centers. In Section 5 we propose a math-heuristic approach for the hardest problems to solve. In Section 6 we report on extensive computational experiments performed to empirically assess the relevance of upgrading. Finally, we provide some discussion and conclusions in Section 7.

2. The discrete p-center problem

To make this manuscript self-contained, we review several well-known aspects related to the discrete p-center problem. Let I be the set of potential center locations and J the set of demand nodes. Consider a cost, say c_{ij}, for allocating node $j \in J$ to center $i \in I$. As discussed in the previous section, this cost may correspond to distance (e.g. road, euclidean), travel time, fuel consumption, vehicle utilization, *et cetera*. We assume that costs are non-negative and satisfy the triangle inequality. Consider the following two sets of binary decision variables: y_i ($i \in I$) equal to 1 if and only if node i is selected for opening a center; x_{ij} equal to 1 if and only if demand node $j \in J$ is allocated to center $i \in I$. The objective of the problem is to select p centers to minimize the maximum allocation cost of the demand nodes to the selected centers.

Daskin [10], proposed the following integer programming model for the discrete p-center problem:

$$\text{minimize} \quad z \tag{1}$$

$$\text{subject to} \quad \sum_{i \in I} c_{ij} x_{ij} \le z \qquad \forall j \in J, \tag{2}$$

$$\sum_{i \in I} x_{ij} = 1 \qquad \forall j \in J, \tag{3}$$

$$x_{ij} \le y_i \qquad \forall i \in I,\ j \in J, \tag{4}$$

$$\sum_{i \in I} y_i \le p, \tag{5}$$

$$y_i \in \{0,1\} \qquad \forall i \in I, \tag{6}$$
$$x_{ij} \in \{0,1\} \qquad \forall i \in I,\ j \in J. \tag{7}$$

In the above model, the objective function (1) together with inequalities (2) define the minmax cost objective; constraints (3) guarantee that every demand node is allocated to one and only one center; inequalities (4) ensure that demand nodes can only be allocated to open centers; the maximum number of p centers to open is imposed by constraint (5). Finally, constraints (6) and (7) state the binary domain of the decision variables.

Other models have been proposed for the problem namely, that by Elloumi et al. [12] and the models (P3) and (P4) introduced by Calik and Tansel [6]. In these three cases, all costs $c_{ij}, i \in I,\ j \in J$, are sorted non-decreasingly ignoring duplicates. Let $\gamma_1, \dots, \gamma_\kappa$ be the resulting sorting with κ denoting the number of distinct values and define $K = \{1, \dots, \kappa\}$. For $i \in I,\ j \in J$, and $k \in K$, let a_{ijk} be a binary parameter indicating whether the cost c_{ij} is smaller than or equal to the kth cost, γ_k, i.e.,

$$a_{ijk} = \begin{cases} 1, & \text{if } c_{ij} \le \gamma_k; \\ 0, & \text{otherwise.} \end{cases}$$

Consider now a binary variable z_k equal to 1 if and only if the maximum allocation cost induced by the selected p centers is equal to γ_k. The discrete p-center problem can be formulated as follows:

$$\text{minimize} \quad \sum_{k \in K} \gamma_k z_k \tag{8}$$

$$\text{subject to} \quad (5), (6),$$

$$\sum_{i \in I} a_{ijk} y_i \ge z_k \qquad \forall j \in J,\ k \in K, \tag{9}$$

$$\sum_{k \in K} z_k = 1, \tag{10}$$

$$z_k \in \{0,1\} \qquad \forall k \in K. \tag{11}$$

In the above model, which corresponds to model (P3) introduced by Calik and Tansel [6], the objective function (8) and constraints (9) ensure that the variable z_k corresponding to the maximum allocation cost is selected and the corresponding solution value is accounted for. Constraint (10) ensures that exactly one maximum allocation cost is defined; the domain of the new z-variables is stated in constraints (11).

Given that exactly one of the variables z_k is selected as 1—constraint (10)—and that all distinct cost values are considered in increasing order, it is possible to strengthen model (P3) by replacing (9) with

$$\sum_{i \in I} a_{ijk} y_i \ge \sum_{q=1}^{k} z_q \quad \forall j \in J,\ k \in K. \tag{12}$$

This enhancement leads to model (P4) proposed by Calik and Tansel [6]. Those authors also show that the above z-variables relate straightforwardly with the u-variables introduced by Elloumi et al. [12]. In the latter work, the authors consider u_k as a binary variable equal to 1 if and only if the radius covering all demand nodes is greater than or equal to γ_k $(k \in K \setminus \{1\})$, i.e., the u-variables are all equal to 1 until γ_k is reached, then they are all equal to 0, unlike the z-variables where only one is equal to 1. Therefore, the relation between these variables is

$$u_k = \sum_{q=k}^{\kappa} z_q \quad k \in K \setminus \{1\}. \tag{13}$$

The above relation can be embedded in a set of constraints presented by Elloumi et al. [12].

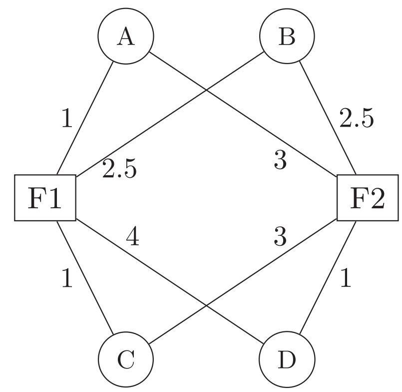


Fig. 1. A 1-center problem with a single connection upgrading.

The models just revisited namely, the model by Daskin [10] and models (P3), (P4) introduced by Calik and Tansel [6], are at the core of the developments we propose for upgrading the p-center solution.

3. Upgrading connections

In this section, we focus on the case in which up to a certain given number of connections can be upgraded. Afterwards, we assume a cost for upgrading the connections together with the existence of a budget for upgrading.

3.1. Upgrading a maximum number of connections

Let us assume that up to t connections can be upgraded. We assume that upgrading a connection between demand node $j \in J$ and potential center $i \in I$ means that the cost for allocating i to j is reduced according to a certain factor. We define the new cost as $(1-f)c_{ij}$ where $f \in [0, F_{\max}]$, and $F_{\max} < 1$. The parameter f is called the discount or compression factor.

Example 1. Consider the example depicted in Fig. 1 where A, B, C, and D are the demand nodes and F1 ad F2 are the potential centers. Assume that the values next to the edges indicate the travel time between the corresponding demand node and the potential center. Suppose the objective is to select one single center. In this case, the optimal solution calls for opening the center in F2 with a maximum travel time of 3.

Assume now that it is possible to upgrade one connection by a factor of at most 0.5. In this case, connection (D,F1) can be upgraded and the resulting travel time becomes 2. This calls for a new optimal solution: opening center F1 with a maximum travel time of 2.5.

This simple illustration shows the impact that an integrated upgrading-and-location-decision can have in the final solution.

□

Given the possibility of upgrading some connections before selecting the p centers, we consider now a mathematical model that requires the introduction of one additional set of binary decision variables. For $i \in I$ and $j \in J$ we define

$$m_{ij} = \begin{cases} 1, & \text{if connection between demand node } j \\ & \text{and location } i \text{ is upgraded;} \\ 0, & \text{otherwise.} \end{cases}$$

The p-center problem with upgraded connections can be formulated mathematically as follows:

$$
\begin{aligned}
\text{(M1)}\quad & \text{minimize} && z(1) && \\
& \text{subject to} && \sum_{i\in I}(c_{ij}x_{ij} - fc_{ij}m_{ij}) \le z && \forall j\in J, && (14)\\
& && m_{ij}\le x_{ij} && \forall i\in I,\ j\in J, && (15)\\
& && \sum_{i\in I}\sum_{j\in J} m_{ij}\le t, && && (16)\\
& && (3)-(7), && &&\\
& && m_{ij}\in\{0,1\} && \forall i\in I,\ j\in J. && (17)
\end{aligned}
$$

In the above model, constraints (14) adapt the actual maximum radius to the upgrading; constraints (15) ensure that a connection can only be upgraded if it is used; conversely, it is not useful to upgrade it. Finally, constraints (17) define the domain of the new decision variables.

Likewise, we can adapt the model (P3) introduced by Calik and Tansel [6] to the above upgrading strategy. Given that we do not know beforehand which cost will be used in a connection (the original one or its upgrade) we must consider both possibilities. Accordingly, we now sort all costs c_{ij} and $(1-f)c_{ij}$ $(i\in I,\ j\in J)$ non-decreasingly (ignoring duplicates). Let $\hat{\gamma}_1,\dots,\hat{\gamma}_{\hat{\kappa}}$ be the resulting sorting and $\hat{K}=\{1,\dots,\hat{\kappa}\}$, with $\hat{\kappa}$ denoting the total number of (different) values found.

To model the problem we need to keep track of the upgraded connections used since we are imposing a limit (t) on their number. For this reason, we keep considering a binary parameter now denoted by $\hat{a}_{ijk}$ associated with the original costs defined as follows:

$$
\hat{a}_{ijk}=\begin{cases}1, & \text{if } c_{ij}\le\hat{\gamma}_k;\\ 0, & \text{otherwise.}\end{cases}\quad i\in I,\ j\in J,\ k\in\hat{K}.
$$

Additionally, we introduce a similar parameter for the upgraded costs:

$$
\hat{b}_{ijk}=\begin{cases}1, & \text{if } (1-f)c_{ij}\le\hat{\gamma}_k;\\ 0, & \text{otherwise.}\end{cases}\quad i\in I,\ j\in J,\ k\in\hat{K}.
$$

We use again variables z_k but now considering $k\in\hat{K}$. Finally, we introduce one additional set of decision variables for keeping track of the upgraded connections used. In particular, for $j\in J$, we define

$$
s_j=\begin{cases}1, & \text{if demand node } j \text{ makes use of an upgraded connection;}\\ 0, & \text{otherwise.}\end{cases}
$$

We can now extend to our problem the model (P3) by Calik and Tansel [6], which leads to:

$$
\begin{aligned}
\text{(M2)}\quad & \text{minimize} && \sum_{k\in\hat{K}}\hat{\gamma}_k z_k && && (18)\\
& \text{subject to} && \sum_{i\in I}\hat{a}_{ijk}y_i+\sum_{i\in I}\hat{b}_{ijk}y_i\ge z_k && \forall j\in J,\ k\in\hat{K}, && (19)\\
& && \sum_{i\in I}\hat{a}_{ijk}y_i+s_j\ge z_k && \forall j\in J,\ k\in\hat{K}, && (20)\\
& && \sum_{j\in J}s_j\le t, && && (21)\\
& && \sum_{k\in\hat{K}}z_k=1, && && (22)\\
& && (5),(6), && &&\\
& && z_k\in\{0,1\} && \forall k\in\hat{K}, && (23)\\
& && s_j\in\{0,1\} && \forall j\in J. && (24)
\end{aligned}
$$

Table 1
Size of the models in terms of (maximum number of) binary variables and constraints.

Model	# 0/1 variables	# constraints
(M1)	$\lvert I\rvert+2\lvert I\rvert\lvert J\rvert$	$2(1+\lvert J\rvert+\lvert I\rvert\lvert J\rvert)$
(M2), (M3)	$\lvert I\rvert+\lvert J\rvert+2\lvert I\rvert\lvert J\rvert$	$3+4\lvert I\rvert\lvert J\rvert^2$

In the above model, the objective function (18) together with constraints (19) and (22) account for the maximum cost used (to minimize). Constraints (20) check the exact upgrades used: in case all demand points are allocated within a cost $\hat{\gamma}_k$ then z_k is equal to 1 and we know that for every demand point, there is an open center within that cost. Hence, either $\hat{a}_{ijk}=1$ or $\hat{b}_{ijk}=1$ or both for some open center i $(y_i=1)$. In this case, if we observe that for some demand point j there is no $\hat{a}_{ijk}=1$ but there is a $\hat{b}_{ijk}=1$ then we know that an upgraded connection with a cost smaller than or equal to $\hat{\gamma}_k$ is being used and thus by constraints (20) we must have $s_j=1$. Constraint (21) ensures that at most t connections are upgraded; constraint (22) reads as before. Finally, we have the maximum number of centers to open and the domain constraints.

The model (P4) proposed by Calik and Tansel [6] can also be extended to our case. The new model, that we call (M3), results from (M2) by replacing (19) and (20) with

$$
\sum_{i\in I}\hat{a}_{ijk}y_i+\sum_{i\in I}\hat{b}_{ijk}y_i\ge\sum_{q=1}^{k}z_q\quad\forall j\in J,\ k\in\hat{K}, \tag{25}
$$

and

$$
\sum_{i\in I}\hat{a}_{ijk}y_i+s_j\ge\sum_{q=1}^{k}z_q\quad\forall j\in J,\ k\in\hat{K}. \tag{26}
$$

Models (M2) and (M3) have potentially many more constraints than (M1) because in case no ties exist, the number $\hat{\kappa}$ of different costs can be $2\lvert I\rvert\lvert J\rvert$. This can be assessed in Table 1. The p-center problem with upgrading has a discrete p-center problem as a particular case and thus, not surprisingly, it is *NP*-hard (see Kariv and Hakimi [19], for the complexity of the p-center problem in the general case).

When [6] introduced their models (P3) and (P4), they reduced the model size by restricting the index set ($\hat{K}$, in our case). This can be accomplished by using valid lower and upper bounds, say lb and ub, on the optimal objective function value. The restricted set of indices is set as $\hat{K}'\equiv\{k\in\hat{K}\mid lb\le\hat{\gamma}_k\le ub\}$.

A lower and upper bound that we can directly consider are

$$
lb_1=(1-f)\mathrm{LB2}\quad\text{and}\quad ub_1=\mathrm{UB2},
$$

respectively, where LB2 and UB2 are the best lower and upper bounds proposed in Calik and Tansel [6] for the original p-center problem, i.e., without upgrading. We denote the resulting restricted set of indices by $\hat{K}_1$. Nevertheless, other alternatives can be proposed, which hopefully lead to improved bounds.

Focusing on feasibility, let $V\subseteq I$ be the set of p centers yielding ub_1 (UB2 in Calik and Tansel [6]). Each center induces a cluster of demand nodes—those allocated to that center—(ties arbitrarily broken). Let T_i be the maximum distance to center $i\in V$ from a demand node of its cluster. Naturally, $ub_1=\max_{i\in V}T_i$.

Let i^* be a cluster (center) such that $i^*\in\arg\max_{i\in V}T_i$ and let e be an edge in that cluster whose length is equal to T_{i^*}. By upgrading edge e, its cost becomes equal to $(1-f)T_{i^*}$. After doing so, we recompute T_{i^*} and $\max_{i\in V}T_i$. If $\max_{i\in V}T_i$ corresponds to the distance of some upgraded edge, then $ub_2=\max_{i\in V}T_i$ defines an improved upper bound. Otherwise, we can repeat this procedure (at most until t connections are upgraded). In this case the improved upper bound, ub_2 is given by the final value found for $\max_{i\in V}T_i$.

The above procedure is in fact a mechanism for obtaining a hopefully improved feasible solution to the problem.

In terms of the lower bound, we can also attempt to improve it as follows. We start by considering the upper bound UB1 proposed by Calik and Tansel [6]. As those authors point out, that value is smaller than or equal to two times the optimal value (see Gonzalez [14]). Hence, UB1/2 provides a lower bound on the optimal value (without upgrading). Thus, $(1-f)\text{UB1}/2$ provides a lower bound on the optimal value with upgrading. The smallest cost $\hat{\gamma}_k$ $(k \in \hat{K})$ which is greater than or equal to $(1-f)\text{UB1}/2$ yields an improved lower bound that we denote by lb_2.

Eventually, the restricted index set induced by lb_1 and ub_1 can be fine-tuned leading to $\hat{K}_2 \equiv \{k \in \hat{K} \mid lb_2 \le \hat{\gamma}_k \le ub_2\}$.

3.1.1. Pre-processing data and fixing variables

Models (M2) and (M3) have more variables and potentially many more constraints than model (M1) (Table 1). The practical success of models (M2) and (M3) stems from the use of upper and lower bounds on the optimal value to eliminate many variables and constraints. The purpose of this section is to extend the use of such bounds to fix variables in the model (M1) and also to strengthen the LP relaxation. It could happen that by eliminating variables all the models turn out to be equally competitive in computing time.

The following remark makes it explicit how to set variables to zero in the model (M1) using bounds on the optimal value.

Remark 1. Let ub and lb be upper and lower bounds on the optimal value of model (M1), respectively. Let (z^*, x^*, y^*, m^*) be an optimal solution of model (M1).

i. If $(1-f)c_{ij} > ub$, then $x^*_{ij} = m^*_{ij} = 0$.
ii. If $c_{ij} \le lb$, then $m^*_{ij} = 0$.

In other words, the remark states that (i) if the upgrade of a connection exceeds a known upper bound, then neither is this connection used nor upgraded in the optimal solution and (ii) if the cost of an edge is smaller than or equal to a known lower bound, then, surely this connection is not upgraded in the optimal solution.

The properties of the optimal solutions in Remark 1 are translated into equalities that can be added to the model (M1) without changing its optimal value.

$$x_{ij} = m_{ij} = 0 \quad \forall i \in I, j \in J : (1-f)c_{ij} > ub \tag{27}$$

$$m_{ij} = 0 \quad \forall i \in I, j \in J : c_{ij} \le lb \tag{28}$$

Apart from fixing variables to zero, upper and lower bounds can be used for modifying the cost matrix.

Remark 2. Let ub and lb be upper and lower bounds on the optimal value of model (M1), respectively. If c_{ij} is replaced by a big constant M for all $i \in I, j \in J$ such that $(1-f)c_{ij} > ub$ and c_{ij} is replaced by lb for all $i \in I, j \in J$ such that $c_{ij} \le lb$, then the optimal value of model (M1) remains the same.

It is worth noting that setting variables to zero, as indicated in Remark 1, does not modify the value of the linear relaxation of the model (M1), while modifying the values in the cost matrix (Remark 2) may change the value of the linear relaxation thus reducing the integrality gap.

A similar result to Remark 1 can be obtained by setting the variables to zero in models (M2) and (M3) as follows:

$$s_j = 0 \quad \forall j \in J : (1-f)\min_{i \in I} c_{ij} > ub, \tag{29}$$

$$s_j = 0 \quad \forall j \in J : \max_{i \in I} c_{ij} \le lb. \tag{30}$$

Note, however, that this requirement is much more demanding and less likely to be met in a cost matrix. So, we do not consider its inclusion of relevance.

3.2. Budget-constrained connection upgrading

We assume now that there is a limited budget for upgrading connections. If the discount factor is the same for all connections as in the previous section, then a budget constraint is equivalent to a limit in the number of connections that can be upgraded. Since this case has already been analyzed, we assume that the discount factor is connection-dependent and the extent of cost compression of a connection is a decision to make.

Let us assume the existence of a maximum amount, say B, that can be spent on upgrading. Let r_{ij} be a decision variable representing the discount factor to adopt in the connection between location $i \in I$ and demand node $j \in J$. If the connection is not upgraded, then $r_{ij} = 0$. Otherwise, the new cost for allocating node j to center i becomes $(1-r_{ij})c_{ij}$ where $r_{ij} \in [R_{\min}, R_{\max}]$, with $0 < R_{\min} < R_{\max} < 1$.

We assume a unit compression/reduction cost similar for all connections. This allows expressing the available budget, B, in terms of the maximum total cost units that can be reduced. The new problem can be formulated mathematically as follows:

$$\text{(M4)} \quad \text{minimize } z \tag{1}$$

$$\text{subject to } \sum_{i \in I}(c_{ij}x_{ij} - c_{ij}r_{ij}) \le z \qquad \forall j \in J, \tag{31}$$

$$(3)\text{–}(7)$$

$$r_{ij} \le x_{ij} \qquad \forall i \in I, j \in J, \tag{32}$$

$$\sum_{i \in I}\sum_{j \in J} c_{ij}r_{ij} \le B, \tag{33}$$

$$r_{ij} \in \{0\} \cup [R_{\min}, R_{\max}] \qquad \forall i \in I, j \in J. \tag{34}$$

Constraints (31) adapt the actual maximum radius to the upgrading; constraints (32) ensure that a connection can only be upgraded if it is used; constraint (33) is the budget constraint. Finally, constraints (34) define the domain of the new decision variables.

Model (M4) has $|I| + |I|\,|J|$ binary variables and $2(1 + |J| + |I|\,|J|)$ constraints.

Proposition 1. *Consider a connection between location $i \in I$ and demand node $j \in J$, and suppose that $R_{\min} = 0$. Let z^* be the optimal value of the objective function. If an optimal solution calls for upgrading connection between i and j, then $z^* = (1-\tilde{r}_{ij})c_{ij}$, where $\tilde{r}_{ij}$ denotes the value of variable r_{ij} in that optimal solution.*

Proof. Let us consider an optimal solution to the problem denoted by (z^*, x^*, y^*, r^*) such that $\tilde{r}_{ij} > 0$, i.e., the connection between location i and demand node j is upgraded. In this case, due to (32) the connection must be used and so we must have $(1-\tilde{r}_{ij})c_{ij} \le z^*$. If equality holds, then the proof is completed. Otherwise, we have $(1-\tilde{r}_{ij})c_{ij} < z^*$. In this case, we can reduce the value of variable r_{ij} from the current one $\tilde{r}_{ij}$ to the value $0 < \tilde{\tilde{r}}_{ij} < \tilde{r}_{ij}$ such that $(1-\tilde{\tilde{r}}_{ij})c_{ij} = z^*$ without changing the solution value, which completes the proof. □

When $R_{\min} = 0$ we can distinguish among three different cases according to the magnitude of $R_{\max}$. Let z^* be the optimal value of the problem and denote by c_t the t-th maximum cost in the optimal solution (c_1 is the largest cost, c_2 is the second largest, etc.). If $c_1 - c_1R_{\max} > c_2$ then we say that $R_{\max}$ is *small*; else, if $c_1 - c_1R_{\max} < c_t$ for some $t \ge 2$ we say that the parameter has a medium value; else, we say that $R_{\max}$ is large.

We provide some insights using Fig. 2. In this figure, we focus on the 6 connections and costs defining an optimal solution for some instance. The horizontal bar depicted for each cost indicates the range for that cost, starting from a maximum compression and ending with the original cost value. The figure is divided into three sub-figures corresponding to $R_{\max} = 0.1, 0.5, 0.9$, respectively.

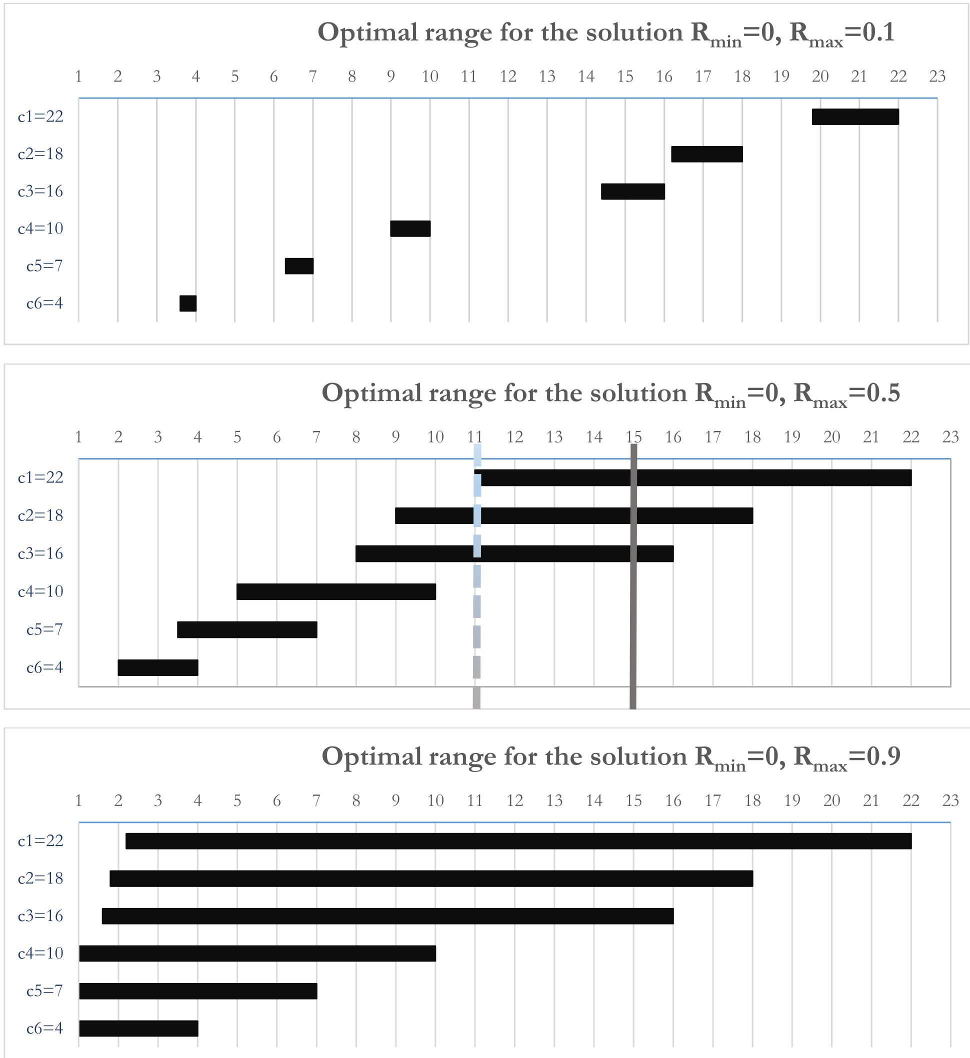


Fig. 2. Optimal solution description when $R_{\min} = 0$.

When $R_{\max}$ is small, independently from the budget, only upgrades on the most-costly connection lead to an improvement in the optimal value. This is illustrated in the upper part of Fig. 2. In this case, we can observe that even if the budget allowed the maximal decrease in the cost c_1 (2.2 units) such decrease would still render c_1, a value larger than the largest value of c_2.

If $R_{\max}$ lies in the so-called range of medium values, then the optimal solution involves upgrading connections starting from the most expensive one (corresponding to c_1) and proceeding to upgrade other connections. However, the upgrade stops either when the budget is attained or when the optimal value reaches $c_1(1 - R_{\max})$. Fig. 2, middle, illustrates this. In this case, for a budget $B = 11$, the optimal solution would be 15 (marked by the dark grey bar) while for a budget $B = 23$, it would be 11 (marked by the dashed grey bar). In particular, for any budget larger than $B = 23$ the optimal objective function value would remain equal to 11 since it cannot be less than the minimum value possible for the highest cost ($c_1(1 - R_{\max}) = 11$).

Finally, if $R_{\max}$ is *large* all connections can be upgraded. This is what we illustrate in Fig. 2 bottom.

Despite the fact that the above results and comments refer to the case $R_{\min} = 0$, we note that our models are general in the sense that we may have $R_{\min} > 0$. In fact, we explore this possibility in the computational experiments whose results are reported in Section 6.

Remark 3. Adapting the models (P3) and (P4) by Calik and Tansel [6] to budget-constrained upgrading poses a major challenge: the sorting for the costs depends on the compression decisions made and thus those models (P3) and (P4) can be adapted only if that sorting is also modeled mathematically and embedded in the model. For this reason, we do not consider adapting those models to our case since we do not foresee any particular advantage of doing so.

An interesting aspect related to the budget-constrained model just presented is related to the values of B that make sense to consider. In fact, the budget constraints become of relevance only if the existing budget is binding and allows changing the solution.

Given that all $r_{ij} \in \{0\} \cup [R_{\min}, R_{\max}]$ it is easy to conclude that the minimum cost we need to pay for implementing an upgrade is equal to

$$B^C_{\min} = R_{\min} \times \min_{i \in I,\, j \in J} \{c_{ij}\}.$$

Thus, a budget below this threshold prevents any upgrade from being feasible.

Suppose now that we ignore the budget constraint (33) in the model (M4) and solve it. In this case, we obtain the minimum objective function value, say $\tilde{z}$, independently from the budget. Consider now the following model:

$$(\widetilde{\mathrm{M4}})\quad \text{minimize} \quad \sum_{i\in I}\sum_{j\in J} c_{ij} r_{ij} \tag{35}$$

$$\text{subject to} \quad \sum_{i\in I}(c_{ij}x_{ij} - c_{ij}r_{ij}) \le \tilde{z} \qquad \forall j \in J, \tag{36}$$

$$(3)\text{–}(7), (32), (34).$$

The optimal value of ($\widetilde{\mathrm{M4}}$) gives the minimum budget that needs to be considered to make the budget constraint non-binding. Thus, the optimal value of this model sets an upper threshold $B^C_{\max}$ that is interesting to consider.

In our experiments, we set $B^C_{\min}$ and $B^C_{\max}$ as just introduced.

Remark 4. It is not necessary to solve model (M4) ignoring constraints (33) to obtain $\tilde{z}$, since $\tilde{z} = (1 - R_{\max})z^*$, where z^* is the optimal value of the original problem without upgrading. Furthermore, if we were interested in obtaining the minimum budget that ensures an improvement of $R\%$, we could solve the model ($\widetilde{\mathrm{M4}}$) by fixing $\tilde{z} = (1 - R\%)z^*$.

3.2.1. Pre-processing data and fixing variables

Variables z, x_{ij} and y_i have the same meaning in model (M4) as in model (M1). Thus, conditions in Remarks 1 and 2 can be easily adapted for model (M4). In particular, the statement in the following remark holds.

Remark 5. Let ub and lb be upper and lower bounds on the optimal value of model (M4), respectively. Let (z^*, x^*, y^*, r^*) be an optimal solution of model (M4).

i. If $(1 - R_{\max})c_{ij} > ub$, then $x^*_{ij} = r^*_{ij} = 0$.
ii. If $c_{ij} \le lb$, then $r^*_{ij} = 0$.

The number of variables that can be set to zero using Remark 5 depends very much on the value of $R_{\max}$. When $R_{\max}$ is very large, the condition $(1 - R_{\max})c_{ij} > ub$ will be fulfilled much fewer times than when this value is smaller.

Remark 2 remains the same by replacing (M1) by (M4) and f by $R_{\max}$.

In order to apply Remark 5 to model (M4), we propose in Remark 6 two bounds for this model.

Remark 6. Any of the lower bounds proposed in Section 3.1 for model (M1) with $f = R_{\max}$ is a lower bound for the optimal value of model (M4). An upper bound can be obtained by following Algorithm 1.

Algorithm 1 (M4) upper bound algorithm.

1: Let $\gamma_1, \dots, \gamma_\kappa$ be the sorted costs ;
2: For all $k \in \{1, \dots, \kappa\}$, let $\mu_k = R_{\max}\gamma_k$ be the maximum amount that can be spent on upgrading the connection with cost γ_k ;
3: Let $T + 1$ be the minimum number of μ-values that can exceed the budget: $\sum_{k=\kappa-T+1}^{\kappa} \mu_k \le B$ and $\sum_{k=\kappa-T}^{\kappa} \mu_k > B$;
4: Let v^* be the optimal value of the problem of upgrading a maximum number of T connections (optimal value of models (M1), (M2) or (M3) with $t = T$) ;
5: v^* is an upper bound of the budget-constrained upgrading model (M4).

Algorithm 1 gives an upper bound on the optimal value of the model (M4) because it obtains the objective value of a feasible solution. Value T is the minimum number of connections that a feasible solution to model (M4) would upgrade because it is obtained by assuming that all the budget is invested to upgrade the most expensive connections and that the maximum amount of budget is spent for each of these expensive connections. In other words, whatever the selected T connections are, budget B is enough to upgrade them. Any feasible solution of models (M1), (M2), or (M3) with $t = T$ is a feasible solution of model (M4), in particular the optimal one.

Table 2
Illustration of the calculation of T in Algorithm 1.

k	1	2	3	4	5	6
γ_k	1	5	7	9	10	14
μ_k	0.8	4	5.6	7.2	8	11.2
$\sum_{t=k}^{6} \mu_t$	36.8	36	32	26.4	19.2	11.2

Example 2. Let suppose that $R_{\max} = 0.8$, the budget is $B = 28$ and Algorithm 1 is applied to the data in Table 2. Then, T is 3 because $26.4 < 28 < 32$. It means that investing as much as possible in upgrading the most expensive connections, 3 connections can be upgraded.

4. Upgrading centers

We now turn our attention to the possibility of upgrading centers, i.e., by considering upgrading all the costs corresponding to allocations decided for a center. We analyze separately the case in which there is a maximum number of centers that can be upgraded and the case in which we have an exogenous budget for the upgrading.

4.1. Upgrading a maximum number of centers

Let $t < p$ be the maximum number of centers that can be upgraded. As done in the previous section when upgrading connections, we consider a fixed compression factor f. As before, when the connection between demand node $j \in J$ and location/center $i \in I$ is upgraded, the corresponding allocation cost becomes $(1 - f)c_{ij}$ where $f \in [0, F_{\max}]$, with $F_{\max} < 1$.

We consider again the m-variables already introduced in Section 3.1. For each $i \in I$ and $j \in J$, m_{ij} is equal to one if and only if the connection between demand node j and center i is upgraded. Now, we also need to consider decision variables indicating whether a center is upgraded. For every $i \in I$ we define

$$v_i = \begin{cases} 1, & \text{if center } i \in I \text{ is upgraded;} \\ 0, & \text{otherwise.} \end{cases}$$

The new problem can be formulated as follows:

$$\begin{aligned}
(\mathrm{Q1})\quad & \text{minimize } z && (1)\\
& \text{subject to } (3)-(7), (14), (15), (17), &&\\
& v_i \le y_i \qquad \forall i \in I, && (37)\\
& m_{ij} \le v_i \qquad \forall i \in I,\ j \in J, && (38)\\
& v_i + x_{ij} \le m_{ij} + 1 \qquad \forall i \in I,\ j \in J, && (39)\\
& \sum_{i\in I} v_i \le t, && (40)\\
& v_i \in \{0, 1\} \qquad \forall i \in I. && (41)
\end{aligned}$$

In addition to the constraints already introduced, we have now constraints (37) ensuring that a center is upgraded only if it is open. Inequalities (38) impose that a connection can only be upgraded if it is allocated to an upgraded center. Constraints (39) guarantee that if a center is upgraded and a demand node is assigned to it then the connection between the demand node and the center is upgraded. Without these constraints, the feasible region would contain solutions where a center is upgraded but not all of its used links are. Constraint (40) states the limit for the number of centers that can be upgraded. Finally, constraints (41) define the domain of the new v-variables.

Table 3
Size of the models in terms of (maximum number of) binary variables and constraints.

Model	# 0/1 variables	# constraints
(Q1)	$2(\|I\|+\|I\|\,\|J\|)$	$2+2\|J\|+3\|I\|\,\|J\|$
(Q2), (Q3)	$2(\|I\|+\|I\|\,\|J\|)$	$3+\|I\|+\|I\|\,\|J\|^2$

Remark 7. We note that constraints (37) are actually not necessary and thus they can be seen as an enhancement. In fact, if we ignore such constraints we allow obtaining a solution such that $v_i = 1$ with $y_i = 0$ which means that we would be upgrading a center that ends up not having any demand node connected to it. In this case, we can just neglect upgrading without neither losing feasibility or deteriorating the objective function value.

Remark 8. A close look into the above model reveals that constraints (39) can be removed from the problem without deteriorating the optimal value. In fact, in case some nodes are allocated to an ungraded center with the original cost and this does not deteriorate the objective function value then it is irrelevant to impose that the upgrade is used: the optimal value for the upgraded problem may be attained upgrading some center without explicitly upgrading all the costs for connecting demand nodes to it.

Remark 9. We note that in the model (Q1) it is possible to relax the integrality constraints for variables y_i and v_i, i.e., constraints (6) and (41), respectively.

We can now extend the model (P3) from Calik and Tansel [6] to the new problem. This is straightforward giving the above contents as well as those presented in Section 3.1.

$$\text{(Q2)}\quad \text{minimize} \sum_{k\in\hat{K}} \hat{\gamma}_k z_k \tag{18}$$

$$\text{subject to} \sum_{i\in I} \hat{a}_{ijk} y_i + \sum_{i\in I} \hat{b}_{ijk} v_i \geq z_k \quad \forall j\in J,\ k\in\hat{K}, \tag{42}$$

$$(5), (6), (22), (23),$$
$$(37), (40), (41).$$

Finally, we can adapt model (P4) in Calik and Tansel [6] to center upgrading. This can be easily achieved by replacing in the above model (Q2), constraints (42) with

$$\sum_{i\in I} \hat{a}_{ijk} y_i + \sum_{i\in I} \hat{b}_{ijk} v_i \geq \sum_{q=1}^{k} z_q \quad \forall j\in J, k\in\hat{K}. \tag{43}$$

The enhanced model will be called (Q3).

In Table 3 we can observe the dimension of the models in terms of binary variables and constraints (excluding unnecessary constraints as already explained). Similar to what we propose in Section 3.1, $\hat{K}$ can be restricted to $\hat{K}_1 \equiv \{k\in\hat{K} \mid lb_1 \leq \hat{\gamma}_k \leq ub_1\}$. Again we can try to fine-tune that set further. In fact, the lower bound lb_2 introduced in the previous section is also valid for these models.

Regarding the upper bound, a similar reasoning can be followed as before leading to an improved upper bound ub_2, which again, corresponds to the final value obtained for $\max_{i\in V} T_i$. Eventually, the restricted index set induced by lb_1 and ub_1 can be further enhanced using lb_2 and ub_2 yielding a restricted set $\hat{K}_2$.

4.1.1. Pre-processing data and fixing variables

Although variables x_{ij} and m_{ij} have the same meaning in the model (M1) and model (Q1) and the intuition calls for using the results from Section 3.1.1, there is a fact deserving special attention. Let $(z^*, x^*, y^*, m^*, v^*)$ be an optimal to model (Q1). Because of constraints (39) it can happen that $c_{ij} \leq lb$ and $m^*_{ij} = 1$, i.e., even if there is no optimal value upgrade, some connections will be upgraded only for guaranteeing that the incident connections to an upgraded center are all upgraded. On the other hand, Remark 8 states that constraints (39) can be removed without deteriorating the optimal value. Fixing to zero all the m_{ij} with $c_{ij} \leq lb$ modifies the feasible region but keeps the optimal value. Then, Remarks 1 and 2 apply to this section replacing (M1) by (Q1). The first remark indicates how to fix variables to zero and the second how to modify the cost matrix: the first enhancement potentially reduces computing time and the second potentially reduces the LP gap.

We do not consider the use of the presented bounds for tackling models (Q2) and (Q3) because, as reasoned for models (M2) and (M3) and variables s_j in Section 3.1.1, the requirements to fix the variables v_i to zero in models (Q2) and (Q3) are more demanding and unlikely to be met in a cost matrix.

4.2. Budget-constrained center upgrading

We assume now an exogenous budget of B for upgrading centers. Similarly, as for budget-constrained connection upgrading, we define a decision variable r_{ij} representing the discount factor to adopt for connection between location $i\in I$ and demand node $j\in J$. If the connection between location i and demand node j is upgraded then the cost c_{ij} becomes $(1-r_{ij})c_{ij}$ where $r_{ij} \in [R_{\min}, R_{\max}]$, with $0 < R_{\min} < R_{\max} < 1$. If center i is not upgraded, then $r_{ij} = 0$, $\forall j\in J$. From the above notation, it is relevant to emphasize that only variable costs are assumed in this work for budget-constrained center upgrading.

As done for connection upgrading, we assume a unit compression/reduction cost similar for all connections, which allows casting the available budget, B, as the maximum total cost units that can be reduced. Considering the notation already presented, the problem can be formulated as follows:

$$\text{(Q4)}\quad \text{minimize}\ z \tag{1}$$

$$\text{subject to}\ (3)-(7), (31)-(34),$$
$$(37), (41),$$
$$r_{ij} \leq v_i \qquad \forall i\in I, \tag{44}$$
$$v_i + x_{ij} \leq 2 + r_{ij} - R_{\min} \qquad \forall i\in I, j\in J, \tag{45}$$
$$r_{ij} - r_{it} \leq 2 - x_{ij} - x_{it} \qquad \forall i\in I, j,t\in J. \tag{46}$$

The novelty in the above model stems from constraints (45), which impose that in case a center i is upgraded and demand node j is allocated to it, the corresponding discount factor r_{ij} is strictly positive (at least equal to $R_{\min}$), which, in turn, ensures that the discount factor is applied to all costs for satisfying demand nodes allocated to the center. Finally, if demand nodes j and t are both allocated to center i, constraints (46) guarantee that their discount factors are equal.

Remark 3 is valid for this model. In terms of the model size, we observe a number of binary variables equal to $2|I| + |I|\,|J|$ and a number of constraints equal to $2(1+|I|+|J|) + 3|I|\,|J| + |I|\,|J|^2$.

As for the connection-upgrading problem, we may inquire about the relevant values for B, i.e., the values that may be of interest to consider since they interfere with the solution.

As before, the budget should be enough to allow at least one upgrade. Define as before $B_{\min} = \min_{i\in I,\, j\in J}\{c_{ij} \times R_{\min}\}$. Additionally, let

$$(i^*, j^*) \in \arg\left\{R_{\min} \times \min_{i\in I,\, j\in J}\{c_{ij}\}\right\}.$$

The minimum budget that we need for implementing an upgrade corresponds to having node j^* as the only demand node allocated to center i^*. Thus, we have $B^F_{\min} = B^C_{\min}$.

Now, let us assume that we relax the budget constraint in the model (Q4). In this case, we obtain an optimal objective function value, say $\tilde{z}$, which is the best we can get no matter the budget we have. As in the case of budget-constrained connection upgrading, $\tilde{z} = (1 - R_{\max})z^*$, where z^* is the optimal value of the problem without upgrading. We can now do as for connection upgrading: we can solve an auxiliary model in which we look for the minimum budget that ensures a coverage radius equal to $\tilde{z}$. This is the maximum value that makes sense to consider for the budget and thus we denote it by $B^F_{\max}$.

We have now the minimum and maximum thresholds, $B^F_{\min}$ and $B^F_{\max}$, that we consider for the budget in our analysis.

4.2.1. Pre-processing data and fixing variables

Remark 10 indicates how to adapt Remark 5 to budget-constrained center upgrading.

Remark 10. Let ub be an upper bound on the optimal value of model (Q4) and let $(z^*, x^*, y^*, r^*, v^*)$ be an optimal solution.

i. If $(1 - R_{\max})c_{ij} > ub$, then $x^*_{ij} = r^*_{ij} = 0$.

As was the case with Remark 5, the number of variables set to zero strongly depends on the value of $R_{\max}$.

Let lb be a lower bound on the optimal value of model (Q4), note that it is not true that if $c_{ij} \le lb$, then $r^*_{ij} = 0$. In model (Q4) some upgrades are forced because all the connections that are incident to an upgraded center are upgraded. Analogously, the cost matrix can only make use of an upper bound for its modification because some connections, despite having a cost smaller than or equal to lb, will be upgraded as they are incident to an upgraded center and the budget required for that will depend on the cost of the connection.

Remark 11. Let ub be an upper bound on the optimal value of the model (Q4). If c_{ij} is replaced by a big constant M for all $i \in I, j \in J$ such that $(1 - R_{\max})c_{ij} > ub$, then the optimal value of model (Q4) remains the same.

Finally, feasible solutions of the classical p-center problem without upgrading help to obtain an upper bound of the budget-constrained center-upgrading optimal value. Remark 12 proposes an auxiliary problem for obtaining an upper bound for the model (Q4).

Remark 12. Let (x, y) be a feasible solution to the p-center problem. Let $\hat{I} = \{i \in I : y_i = 1\}$ be the subset of p selected centers. For all $i \in \hat{I}$, let $\theta_i = \sum_{j \in J} c_{ij}x_{ij}$ be the total cost of the allocated demand nodes and let $\beta_i = \max_{j \in J} c_{ij}x_{ij}$ be the maximum cost of an allocated demand node. The optimal value of the following model whose variables are w and g_i is an upper bound of the budget-constrained center-upgrading optimal value. For all $i \in \hat{I}$, variable g_i represents the discount factor applied to all demand nodes allocated to center i.

$$
\begin{array}{lll}
\text{(U)} \quad \text{minimize} & w & \\
\quad\quad\; \text{subject to} & \displaystyle\sum_{i \in \hat{I}} \theta_i g_i = B & \\
& w \ge \beta_i (1 - g_i) & \forall i \in \hat{I} \\
& g_i \in \{0\} \cup [R_{min}, R_{max}] & \forall i \in \hat{I}
\end{array}
$$

5. Math-heuristic procedure for budget-constrained upgrading

As shown by the *Empirical analysis* presented in the next section, model (Q4) can be solved to proven optimality only in a very limited number of instances. This motivates the development of a heuristic for tackling that model as we propose in this section. In particular, we propose a genetic algorithm for budget-constrained center upgrading.

The structure of the procedure we propose is formalized in Algorithm 2: once an N-dimensional initial population, P_0, has been generated, crossover, mutation, and local search operators are iteratively carried out until the stopping condition occurs. We have encoded the solutions in such a way that we only save the value of the p open centers. Hence, each individual of the initial population is a combination of p centers from the $|I|$ candidates.

Algorithm 2 Genetic algorithm for budget-constrained center upgrading.

```
i ← 0;
P_i ← create_initial_population(N);
repeat
    P_i ← crossover(P_i);
    P_i ← mutation(P_i);
    P_i ← local_search(P_i);
    if random < probability then
        P_i ← intensive_local_search(P_i);
    end if
    P_{i+1} ← P_i;
    i ← i + 1;
until stopping criterion occurs
```

Initially, we randomly generate $N - 1$ combinations of p centers from the $|I|$ potential ones. Then, we also consider the optimal solution of the associated non-upgrading problem. This way, we obtain the initial N individuals.

Note that given an individual, i.e., a set of p open centers, the assignment of demand nodes to the open centers in the upgraded solution does depend on the budget. The condition in model (Q4) imposing that all the incident connections to an upgraded center have the same discount factor means that for some nodes, the best assignment may not correspond to the closest center among those that are open. For this reason, we decided to approximate the fitness of an individual (objective function value) by using the linear optimization model (U) presented in Remark 12 since it provides a good upper bound on that value. Nevertheless, we try to improve this upper bound further by reallocating some demand nodes in each individual as detailed in Algorithm 3.

Algorithm 3 Improving the fitness of an individual when tackling model (Q4).

```
Let (w, θ, g) be the solution to model (U) in Remark 12;
For each i ∈ Î, let j(i) be the demand node such that c_{ij(i)} = β_i;
for all i, s ∈ Î : β_s > β_i do
    if (1 − g_i)β_i > (1 − g_s)c_{sj(i)} then
        Reallocate demand node j(i) to center s;
        Update β_i, θ_i and θ_s and solve model (U);
    end if
end for
```

The "spirit" of Algorithm 3 is to reallocate the demand points to centers that, even though they are not the closest, they are the least costly when a discount factor is applied. If the condition $(1 - g_i)\beta_i > (1 - g_s)c_{sj(i)}$ holds, then we have that although the demand node $j(i)$ is closer to center i than to center s in the cost matrix (i.e., $\beta_i = c_{ij(i)} < c_{sj(i)}$), when both centers are upgraded, it is cheaper to allocate $j(i)$ to center s than to allocate it to center i. We repeat this check until there is no exchange that reduces the cost. Finally, the fitness value of each individual in the genetic algorithm population is the value of its largest β value.

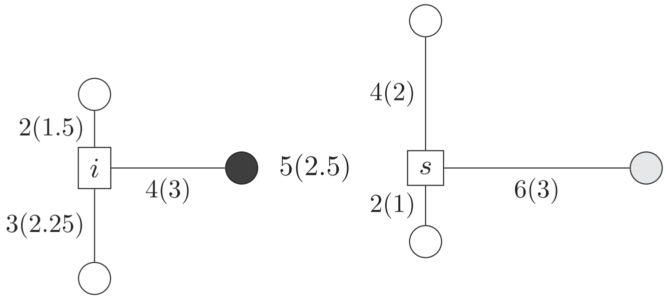


Fig. 3. Non-optimal allocation of demand nodes under center upgrading.

Example 3. Consider the example depicted in Fig. 3 where i and s are two open centers with compression factors $g_i = 0.25$ and $g_s = 0.5$ respectively. The circles represent demand nodes allocated to the centers. Next to each edge, we present the original costs and their upgrades (obtained by multiplying the former by $(1 - g_i)$ or $(1 - g_s)$, as appropriate). The black node represents the demand node served by i that is the furthest (more costly) served by this center, i.e., it represents $j(i)$. To the right of this node we present its original assignment cost to s and the corresponding upgrading. The gray node is the furthest from center s, i.e., $j(s)$. Hence, in this example we have $\beta_i = 4$ and $\beta_s = 6$. Since the distance (cost) from the black node to i is smaller than its distance from s (4 and 5, respectively), that node is initially allocated to i. However, the upgraded distance is smaller from s than from i. Algorithm 3 gives the details for this improved allocation.

As a consequence of the way the fitness of an individual is obtained, Algorithm 2 turns out to be a math-heuristic, in which the calculation of the fitness function is not trivial. The adequacy of the fitness computation to a specific instance being considered confers quality to the algorithm.

The crossover operator randomly selects two individuals of the population (parents), merges them, and assigns to each center in the union a probability proportional to the number of times it appears in the parents (each center can appear once or twice). Finally, the crossover operator generates two offspring, selecting for each of them p centers of the union with the assigned probabilities. Each offspring replaces the worst individual of the population (in terms of fitness value) if the fitness of the offspring is better (smaller) than that of the worst, as long as the offspring is not already in the population. We do not allow for duplicates of individuals in the population to avoid premature convergence of the algorithm since, after some preliminary experiments, we verified that the genetic algorithm converged very quickly.

The mutation operator starts by randomly selecting an individual from the population and a center in the selected individual. That center is replaced with a different one not in the individual. The mutated individual replaces the worst individual in the population except if the mutated individual is already in the population. Mutating individuals in the population further improves the level of diversification.

The local search operator seeks to confer intensification to the genetic algorithm. It randomly selects an individual of the population as well as a center in this individual, and a set of centers not in the individual (and different from the selected one). The fitness values of the individuals obtained by exchanging the selected center with any of the centers in the generated set are computed. The best of these new individuals in terms of fitness value replaces the worst individual in the population except if the individual is already in the population.

Each iteration of Algorithm 2 applies the three above operators (crossover, mutation and local search) to the current population, P_i. Furthermore, an intensified local search is performed in some iterations, which is ruled by some given *probability*. This additional local search operator is similar to the one above described but, as its name indicates, more intense. It checks the fitness value when exchanging all the centers in the individual with others centers that are not in it. A relevant difference is that the intensified local search is applied not to a randomly chosen individual of the population but to all the individuals in the population with the smallest fitness values. Again, the best individual obtained by performing the intensive local search replaces the worst individual of the population except if it is already part of it.

6. Empirical analysis

In this section, we report on the computational tests performed to assess the contributions proposed in the previous sections. We start by describing the data used and also by detailing the experimental setting. Next, we present some preliminary computations that indicate the direction we should follow in terms of more intensive testing. The latter is reported in the fourth subsection.

6.1. Data and experimental setting

We use the uncapacitated p-median instances from the OR-Library which consists of 40 instances where the number of nodes, n, ranges from 100 to 900 and p ranges from 5 to $n/3$. Like usually done in literature, the all-pair shortest path Floyd's algorithm was considered for obtaining the cost matrix from the original data retrieved from the OR-Library. For all the mathematical models discussed in this work we used IBM ILOG CPLEX 20.1.0.0 as the off-the-shelf solver. The computational tests were performed in an Intel(R) Xeon(R) CPU E5-2650 v3 @ 2.30 GHz. The default parameter values of the solver were considered although a time limit of 6 h was imposed when tackling each instance.

In the case of upgrading a maximum number of connections—models (M1), (M2), and (M3), we set t equal to 5%, 10%, and 25% of the total number of nodes in an instance. For the problem consisting of upgrading a maximum number of centers—models (Q1), (Q2) and (Q3), we set t equal to 5%, 10% and 25% of the value of p in each instance. In addition, for the last three models, we also include in the analysis the case $t = 1$ to see how the solution changes if only one center can be upgraded. In all cases, for each value of t we consider a discount factor f equal to 0.2, 0.4, 0.6, and 0.8.

In the budget-constrained upgrading models we analyze different values of B between $B^C_{\min}$ and $B^C_{\max}$ in model (M4) and between $B^F_{\min}$ and $B^F_{\max}$ in model (Q4). For these models we set $R_{\min} = 0.2$ and $R_{\max} = 0.8$.

Instances pmed1 to pmed5 from the OR-library ($n = 100$) were considered when using each of the models presented in this work for all the described combinations of parameters. Furthermore, models (M2), (M3), (Q2) and (Q3) were handled considering two restricted sets $\hat{K}$, according to the lower and upper bounds discussed: $\hat{K}_1 \equiv \{k \in \hat{K} \mid lb_1 \leq \hat{\gamma}_k \leq ub_1\}$ and $\hat{K}_2 \equiv \{k \in \hat{K} \mid lb_2 \leq \hat{\gamma}_k \leq ub_2\}$. We look into how the pre-processing procedures described in Sections 3.1.1, 3.2.1, 4.1.1, and 4.2.1 behave in the proposed models. In the results reported in this section, a more exhaustive study of models (M2) and (Q2) is also carried out for the 40 available p-median instances, analyzing in detail the use of the proposed bounds.

To make this paper self-contained and also to allow the readers to fully reproduce our results, the results presented in this section are complemented with an electronic appendix where we present for all instances tested, their optimal value as well as the new lower and upper bounds lb_2 and ub_2. Finally, we note that all

Table 4
Connection-upgrading problem: average computing time (seconds) required to solve the six tested models for instances pmed1–pmed5.

	M1	M1′	M2 + $\hat{K}_1$	M2 + $\hat{K}_2$	M3+$\hat{K}_1$	M3+$\hat{K}_2$
pmed1	540.9	277.6	149.2	180.1	224.1	176.1
pmed2	365.3	139.7	136.7	118.7	204.5	197.9
pmed3	362.3	110.6	176.6	165.4	252.6	265.0
pmed4	235.8	41.0	59.5	66.5	133.3	131.0
pmed5	73.9	27.5	27.8	21.6	101.3	55.7
Average	315.7	119.3	110.0	110.4	183.1	165.1

Table 5
Connection-upgrading problem: average LP gap (%) achieved by the models for the instances based upon pmed1–pmed5.

	M1	M1′	M2	M3
pmed1	30.5	26.9	48.2	17.9
pmed2	35.4	25.8	44.5	15.3
pmed3	34.4	27.2	50.5	17.4
pmed4	44.2	31.8	52.5	18.6
pmed5	57.3	32.7	44.2	19.7
Average	40.3	28.9	48.0	17.8

the instances as well as the source codes will be made available to any interested reader upon a request to the authors.

6.2. Preliminary results—connection upgrading

We start with the connection-upgrading problem when up to a certain number t of connections can be upgraded. Models (M2) and (M3) were solved considering two restricted sets $\hat{K}$, using different bounds. Regarding model (M1), it was solved as presented in Section 3.1 and, in addition, it was also solved using the pre-processing procedure described in Section 3.1.1. Specifically, we incorporated equalities (27) and (28) using $ub2$ and $lb2$ as upper and lower bounds, respectively. Moreover, the cost matrix was modified with these bounds following Remark 2 and setting $M = 500$, a value greater than any of the costs involved in the instances considered. We call this model (M1′). Recall that we ended up with six different models: model (M1), model (M1′), and models (M2) and (M3) restricting $\hat{K}$ according to $\hat{K}_1$ and $\hat{K}_2$. In what follows we denote these models as M1, M1′, M2 + $\hat{K}_1$, M2 + $\hat{K}_2$, M3+$\hat{K}_1$ and M3+$\hat{K}_2$.

In this section, we present some preliminary results using the instances based upon pmed1–pmed5 ($n = 100$). The first aspect of interest to analyze concerns the performance of the models namely in comparison with each other. Table 4 presents the average computing time (in seconds) required by each model. Each row in the table averages 12 values (3 values of t and 4 values of f). The detailed results for each instance can be found in the Appendix (Tables A1–A5), where the results are presented according to the cost compression factor (f) and the maximum number of connections that can be upgraded (t). We conclude directly that models (M1′) and (M2) outperform the other models although it is not clear whether model (M2) is easier to tackle when $\hat{K}_1$ is used or else when the choice goes to $\hat{K}_2$. We can also conclude that the pre-processing procedure carried out in model (M1′), drastically reduces the computing time. In the Appendix (Table A6), we present the percentage of variables that are fixed to zero during the pre-processing procedure for each instance, where we can see that, in some cases, it exceeds 80%.

The results observed in Tables A1–A5 also do not reveal any dominance in terms of a specific combination of f and t. Nevertheless, we see that the computing time is clearly dependent on the instance even when the same dimension is considered. In fact, recall that instances pmed1–pmed5 all consider 100 nodes whereas p takes values 5, 10, 10, 20, and 33, respectively.

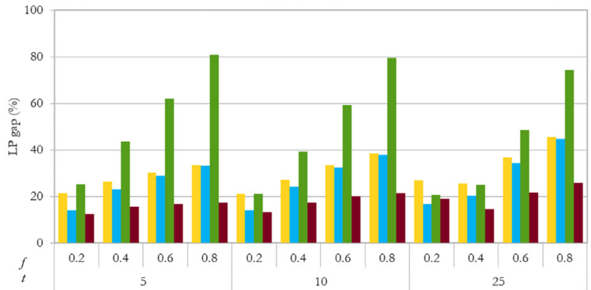
Another relevant information concerns the gap provided by the linear relaxation of the different models in use. Such gap is computed according to $100 \times (z^* - LR)/z^*$ where z^* denotes the optimal value of the problem and LR the optimal value of the linear relaxation. The information is summarized in Fig. 4. The values used in this figure are detailed in the Appendix (Tables A7–A11). Concerning the LP gap, we conclude that model (M3) outperforms by far the other models, followed by model (M1′). Interestingly, we see that (M1) is often better than (M2) although this does not impact the computing time required to solve the model to proven optimality. This is an indication that although (M1) seems to lead to a better polyhedral description of the problem feasibility set, this is still not good enough to boost the solver. In Table 5 we can see the average LP gap achieved with each model. In this table, the above conclusions become clearer. Regarding the pre-processing procedure carried out in model (M1′), not only is it effective in greatly improving the computing time (Table 4), but also it improves considerably the linear relaxation of the model, going from an average LP gap of 40.3% in the model (M1) to 28.9% in the model (M1′).

An important aspect of our problem concerns the decrease in the optimal covering cost by upgrading connections. Fig. 5 depicts this information for the 12 instances based upon pmed1–pmed5. The detailed values are presented in the Appendix (Table A12). In this figure, we observe quite significant improvements in the solution. Furthermore, as expected, this improvement increases both with the cost reduction factor and with the maximum number of connections that can be upgraded. In all 60 instances built from pmed1–pmed5 apart from four, the number of connections upgraded in the optimal solution leading to the decreased optimal values reported in the Appendix is the maximum possible. The four exceptions are observed for instances in which the maximum number of connections that can be upgraded is equal to 25. In particular, we observe 92%, 96%, 96%, and 96% of the maximum number being upgraded in instances pmed1 ($t = 25$, $f = 0.2$), pmed2 ($t = 25$, $f = 0.2$; $t = 25$, $f = 0.6$), and pmed5 ($t = 25$, $f = 0.4$). These percentages depend on the optimal solution and thus we should be careful in analyzing them since these instances might have alternative optimal solutions.

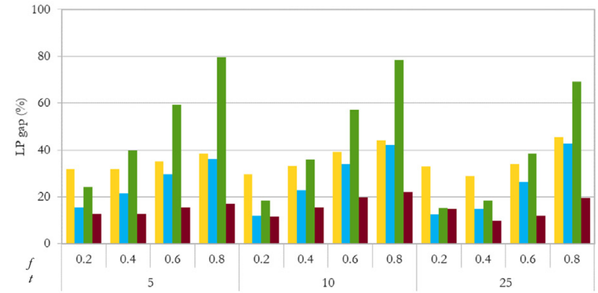
6.3. Preliminary results—center upgrading

We perform now a similar analysis as above but for center upgrading. Models (Q2) and (Q3) were solved considering two restricted sets $\hat{K}$, using different bounds for center upgrading. Regarding model (Q1), it was solved as shown in Section 4.1 and, in addition, it was also solved using the pre-processing procedure described in Section 4.1.1. Specifically, as in the case of connection upgrading, we incorporated equalities (27) and (28) using $ub2$ and $lb2$ for center upgrading as upper and lower bounds, respectively. Again, the cost matrix was modified with these bounds following Remark 2 and setting $M = 500$. Note that to apply this pre-processing procedure it is necessary to remove constraints (39) from model (Q1). We call this model (Q1′). As pointed out in Remark 9, it is possible to relax the integrality constraints for variables y_i and v_i in the model (Q1). We run some experiments to test the performance of this relaxed model and, in most cases, the computing times were similar or even worse in the relaxed model. For this reason, we decided not to include it in the analysis.

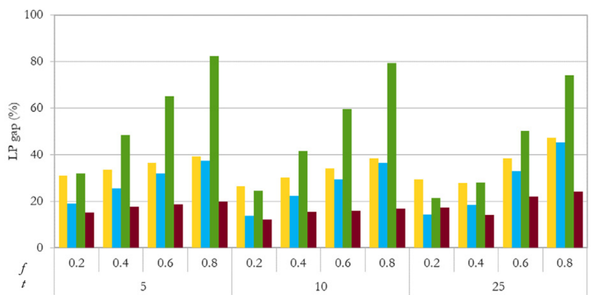
With all of the above, in the case of center upgrading we have the following six models: Q1, Q1′, Q2+$\hat{K}_1$, Q2+$\hat{K}_2$, Q3+$\hat{K}_1$ and Q3+$\hat{K}_2$. Again, we start by assessing the performance of those models namely, in comparison with each other, using instances pmed1–pmed5.



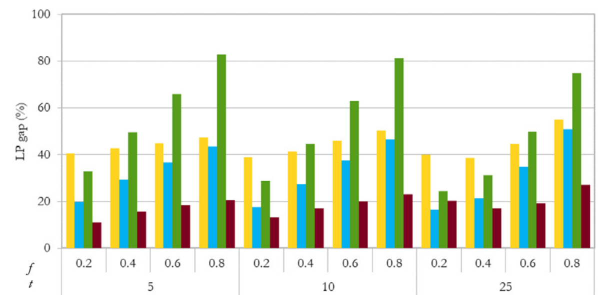
(a) pmed1.



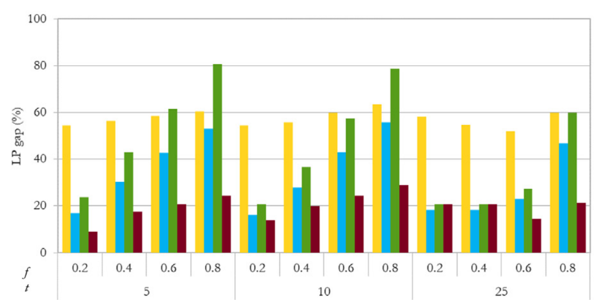
(b) pmed2.



(c) pmed3.



(d) pmed4.



(e) pmed5.

Fig. 4. Connection upgrading: bound provided by the linear relaxation (%) for the instances based upon pmed1–pmed5.

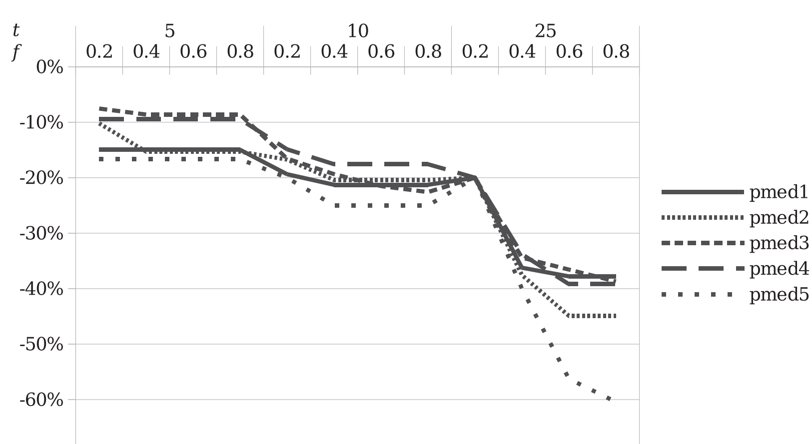


Fig. 5. Percentage decrease in the optimal cost for upgrading connections.

Table 6
Center upgrading: average computing time (seconds) required by the six tested models for instances pmed1–pmed5.

	Q1	Q1′	Q2+$\hat{K}_1$	Q2+$\hat{K}_2$	Q3+$\hat{K}_1$	Q3+$\hat{K}_2$
pmed1	1306.6	249.4	76.4	62.3	283.5	179.5
pmed2	674.6	354.8	34.0	40.9	152.2	211.9
pmed3	1453.4	508.0	47.0	51.1	153.5	222.1
pmed4	390.7	184.0	30.5	16.8	100.1	116.1
pmed5	205.9	93.8	16.3	9.4	53.6	68.7
Average	806.2	278.0	40.8	36.1	148.6	159.6

Table 7
Center upgrading: average LP gap (%) achieved by the models for instances pmed1–pmed5.

	Q1	Q1′	Q2	Q3
pmed1	28.3	18.9	30.0	13.5
pmed2	35.9	20.4	30.8	14.1
pmed3	36.7	24.8	41.0	17.9
pmed4	45.1	28.1	43.7	19.1
pmed5	57.1	27.2	35.5	19.7
Average	40.6	23.9	36.2	16.8

Table 6 presents the average computing time (in seconds) required by each model for the first five pmed instances. Each row averages the values for the different values of t and f. The detailed results can be found in the Appendix (Tables B1–B5), where the results are detailed according to the cost compression factor (f) and the maximum number of centers that can be upgraded (t). We conclude that model (Q2) outperforms the other models although, again, it is not clear whether the model is easier to tackle when $\hat{K}_1$ is used or else when the choice goes to $\hat{K}_2$. We can also observe that, although the pre-processing procedure carried out in model (Q1′) greatly improves the computing time, it is still far from the computing time provided by other models such as (Q2). Table B6 in the Appendix shows the percentage of variables that are fixed to zero during the pre-processing procedure of the model (Q1′) for each instance.

The results observed in Tables B1–B5 also do not reveal any dominance in terms of a specific combination of f and t. Nevertheless, as for connection upgrading, the computing time seems to be dependent on the instance even when the same dimension is considered. In fact, recall that instances pmed1–pmed5 all consider 100 nodes whereas p takes values 5, 10, 10, 20, and 33, respectively.

Another relevant information concerns the gap provided by the linear relaxation of the different models used computed as above explained. This information is summarized in Fig. 6. The values used in this figure are detailed in the Appendix (Tables B7–B11). Observing the figure, we see that model (Q3) outperforms the other models by far. Furthermore, model (Q1′) is always better than model (Q2) although this does not reflect in terms of the computing time required to solve the model to proven optimality. In Table 7 we can see the average LP gap achieved with each model.

An important aspect of our problem concerns the decrease in the maximum cost due to upgrading centers, i.e., upgrading all connections made to them. Fig. 7 depicts this information for the instances built from pmed1–pmed5. Recall that we have set t equal to 1 and equal to 5%, 10%, and 25% of the value of p. Particularly, for pmed1 ($p = 5$) $t \in \{1, 2\}$, for pmed2 and pmed3 ($p = 10$) $t \in \{1, 3\}$, for pmed4 ($p = 20$) $t \in \{1, 2, 5\}$, and for pmed5 ($p = 33$) $t \in \{1, 2, 3, 8\}$. The detailed values are presented in the Appendix (Table B12). In Fig. 7 we observe quite significant improvements in the solution. Furthermore, as expected, this improvement

Table 8
Average computing time (seconds) required to solve models (M1′) and (M2) for instances pmed6–pmed10.

	M1′	M2 + $\hat{K}_1$	M2 + $\hat{K}_2$
pmed6	7475.0	1703.3	1280.1
pmed7	4576.2	623.1	584.4
pmed8	4414.1	625.2	502.0
pmed9	1360.3	298.8	187.2
pmed10	581.2	96.7	78.4
Average	3681.4	669.4	526.4

increases both with the cost reduction factor and with the maximum number of centers that can be upgraded. For $t = 3$ the depicted lines for pmed2 and pmed 3 slightly differ while those depicted for pmed3 and pmed5 overlap. In all cases, the number of upgraded centers in the optimal solution is the maximum possible.

6.4. Additional results

As seen in the previous section, model (Q2) clearly outperforms in terms of computing time the other models for center upgrading. In the case of connection upgrading, models (M1′) and (M2) outperform the other models, providing, in many cases, similar computing times. However, after some previous experiments, we decided not to include model (M1′) in the analysis of larger instances, since its computing time skyrockets as the size of the instances grows. As an example, Table 8 shows the average computing time of models (M1′) and (M2) to solve the instances based upon pmed6–pmed10 ($n = 200$).

For all the above reasons, we select models (M2) and (Q2) for connection upgrading and center upgrading, respectively, to perform a more exhaustive analysis with larger instances. Furthermore, in neither case, it is clear whether these models are easier to tackle when $\hat{K}_1$ or $\hat{K}_2$ are used. Hence, we analyze now the computing time required to solve the 40 p-median instances from the OR-Library using models (M2) and (Q2) with both bounds.

Fig. 8 depicts the relative deviation in terms of the computing time when using $\hat{K}_2$ instead of $\hat{K}_1$ with models (M2) (connection-upgrading problem) and (Q2) (center-upgrading problem). The presented results correspond to average results for all instances associated with each pmed instance (the corresponding value of p and n is specified). As we can observe in this figure, most of the dots are negative. This indicates that in general, it is faster to solve the models when bounds lb_2 and ub_2 are used. Nevertheless, we see some cases in which this was not the case. In connection-upgrading problems, an overall average improvement in computing time of 9.4% is achieved when using $\hat{K}_2$ instead of $\hat{K}_1$. In some instances with a high number of nodes ($n \geq 500$), we see that bounds lb_2 and ub_2 do not seem to work well. In the case of center-upgrading problems, an overall average improvement of 14.7% is achieved when bounds lb_2 and ub_2 are used. For these problems, we see that very few instances fail to improve their computing time when using the new proposed bounds. Interestingly, looking into detail when this happens, it is mainly for small values of p.

Fig. 9 depicts the computing time (in seconds) required to solve all the instances that are considered in this work. Average results are presented for the tested combinations of t and f. The figure shows that the computing time increases considerably with the size of the problem. We also observe that for the same number of nodes the models tend to become more tractable by the solver when p increases. In the Appendix, Table C1 details the values used to draw the figures.

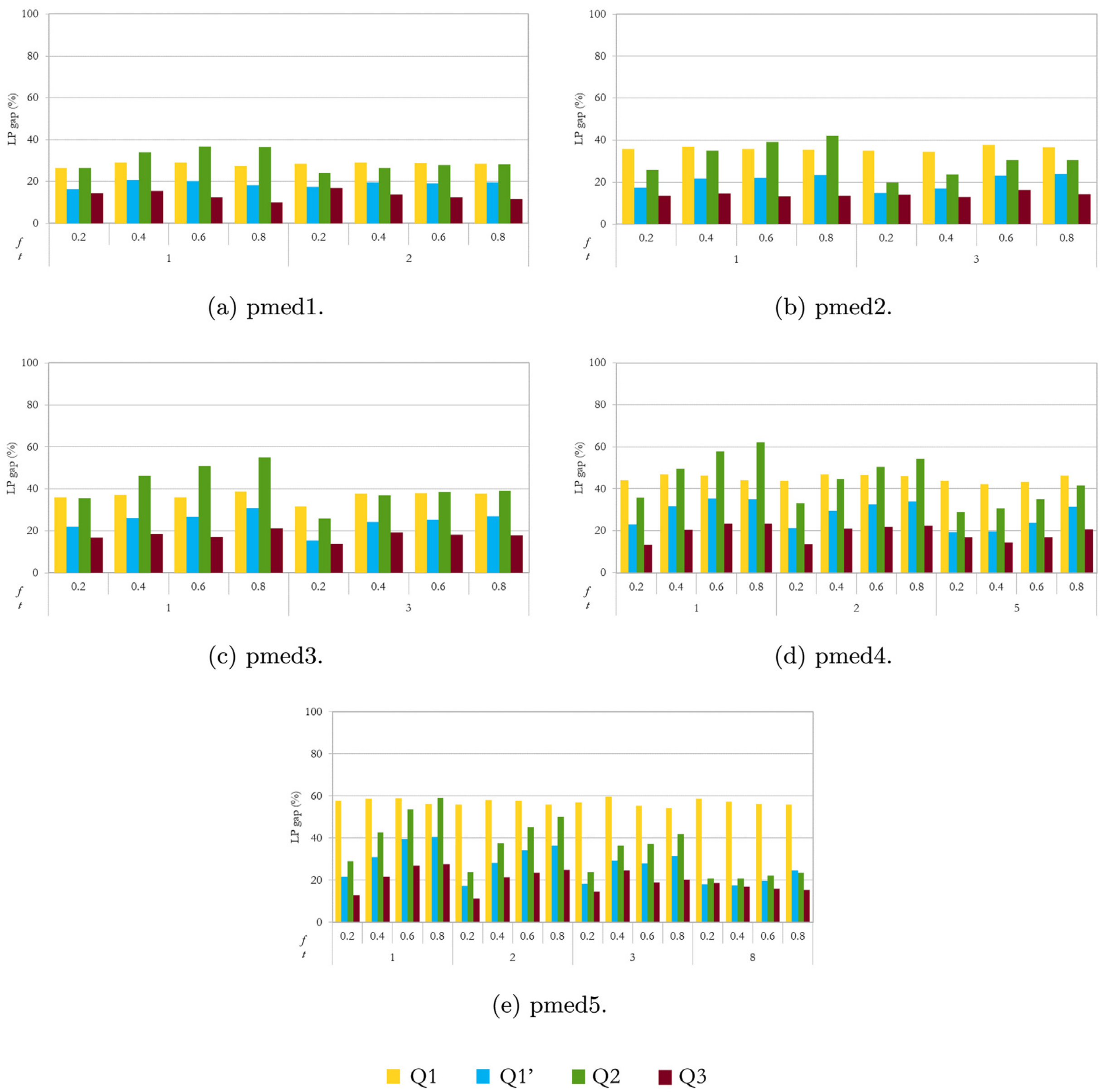


Fig. 6. Center upgrading: bound provided by the linear relaxation (%) for the instances based upon pmed1–pmed5.

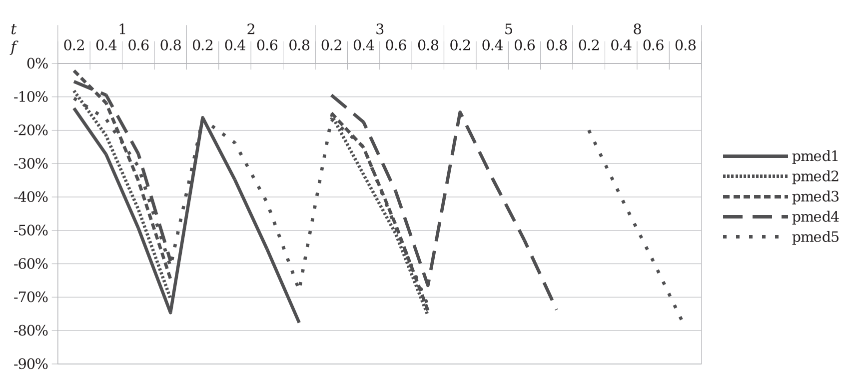


Fig. 7. Percentage decrease in the optimal cost for upgrading centers.

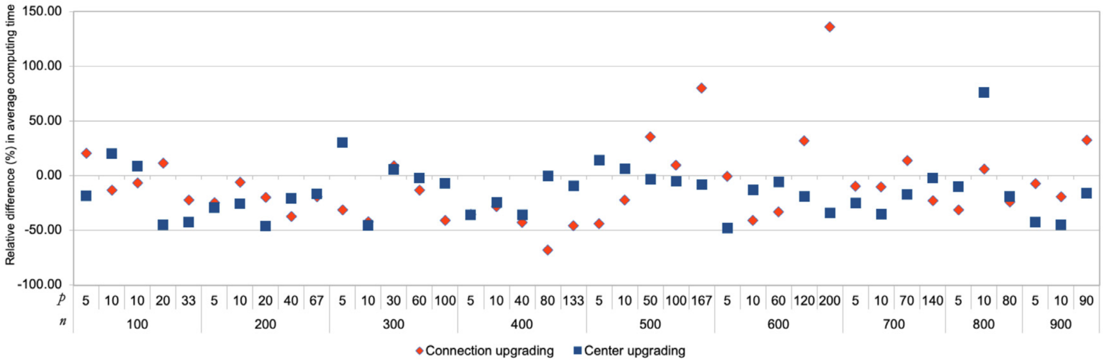
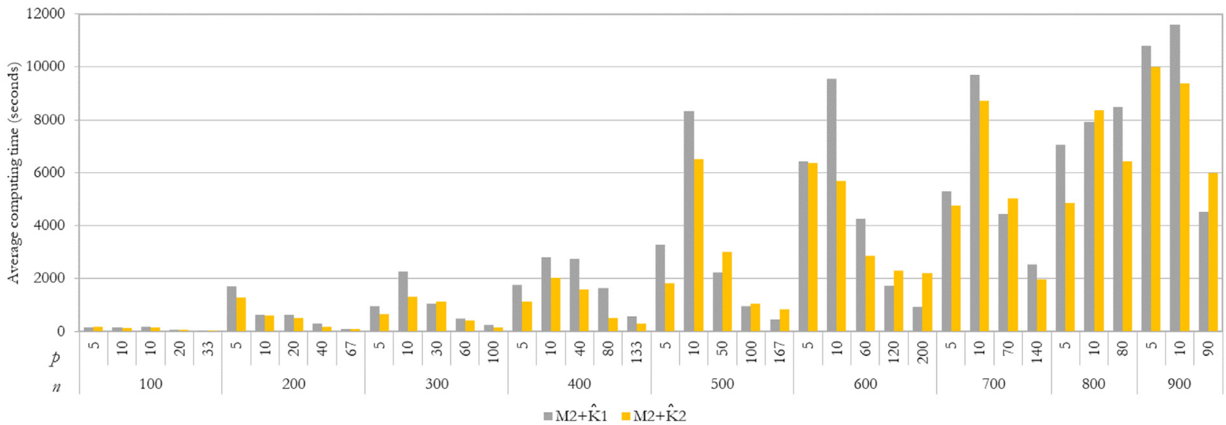
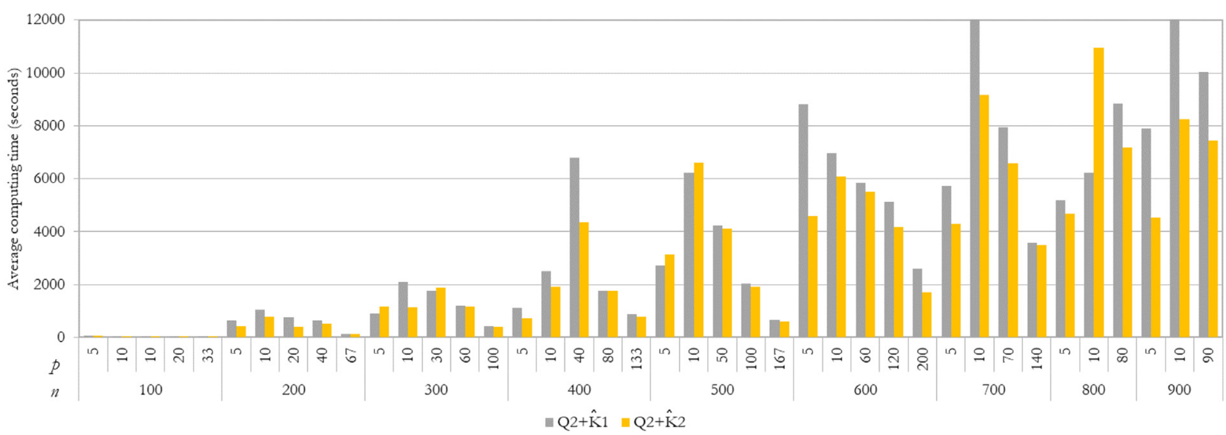


Fig. 8. Relative deviation in the average computing time when using $\hat{K}_2$ instead of $\hat{K}_1$ ($100 \times (\mathrm{CPU2} - \mathrm{CPU1})/\mathrm{CPU1}$).



(a) Connection upgrading.



(b) Center upgrading.

Fig. 9. Average computing time (seconds) for solving the instances based upon pmed1–pmed40.

6.5. Results for budget-constrained upgrading

The preliminary experiments that we executed revealed that the budget-constrained upgrading models are quite difficult to solve to proven optimality. In particular, we cannot expect to solve large-scale models using an off-the-shelf solver as done in the previous section. Additionally, given the existence of a budget constraint, it is worth performing a sensitivity analysis considering different values of the budget. For the above reasons, we now focus our analysis again on the 100-node instances namely, those built from pmed1–pmed5.

For these instances, we started by computing the minimum and maximum thresholds that are meaningful in terms of the budget (see the discussion presented at the end of Sections 3 and 4). Recall that values are denoted by $B^C_{\min}$ and $B^C_{\max}$ (for connection upgrading) and $B^F_{\min}$ and $B^F_{\max}$ (for center upgrading). Let us define

$$\Delta^C_B = B^C_{\max} - B^C_{\min},$$

for the first case and

$$\Delta^F_B = B^F_{\max} - B^F_{\min},$$

Table 9
Reference values for the budget-constrained problems.

Instance	B^C_{min}	B^C_{max}	B^C_{min} $+\frac{3}{4}\Delta^C_B$	$+\frac{1}{2}\Delta^C_B$	$+\frac{1}{4}\Delta^C_B$
pmed1	0.2	3762.6	2822.0	1881.4	940.8
pmed2	0.2	3065.6	2299.3	1532.9	766.6
pmed3	0.2	3337.2	2503.0	1668.7	834.5
pmed4	0.6	2311.4	1733.7	1156.0	578.3
pmed5	0.2	897.0	672.8	448.6	224.4
Instance	B^F_{min}	B^F_{max}	B^F_{min} $+\frac{3}{4}\Delta^F_B$	$+\frac{1}{2}\Delta^F_B$	$+\frac{1}{4}\Delta^F_B$
pmed1	0.2	4699.4	3524.6	2349.8	1175.0
pmed2	0.2	3740.1	2805.2	1870.2	935.2
pmed3	0.2	3903.1	2927.4	1951.6	975.9
pmed4	0.6	2668.9	2001.8	1334.7	667.7
pmed5	0.2	1103.3	827.5	551.7	276.0

for the second one. For each instance, four values are investigated for the budget:

$$B^C_{\min} + \frac{1}{4}\Delta^C_B,\ B^C_{\min} + \frac{1}{2}\Delta^C_B,\ B^C_{\min} + \frac{3}{4}\Delta^C_B,\ \text{and } B^C_{\min} + \Delta^C_B = B^C_{\max},$$

for connection upgrading, and

$$B^F_{\min} + \frac{1}{4}\Delta^F_B,\ B^F_{\min} + \frac{1}{2}\Delta^F_B,\ B^F_{\min} + \frac{3}{4}\Delta^F_B,\ \text{and } B^F_{\min} + \Delta^F_B = B^F_{\max},$$

for center upgrading. In Table 9 we present the values considered. Note that, for each instance, the highest value considered for the budget corresponds to $B^C_{\max}$ and $B^F_{\max}$ for connection and center upgrading, respectively. Note also that $B^C_{\min} = B^F_{\min}$ for each instance.

For connection upgrading, model (M4) was solved as presented in Section 3.2 and, in addition, it was also solved using the pre-processing procedure described in Section 3.2.1. Specifically, we considered $lb2$ with $f = R_{\max}$ as the lower bound and we obtained upper bounds for this problem using the procedure described in Algorithm 1. In step 4 of Algorithm 1 we used model (M2) because it is the fastest. As described in previous models, the cost matrix was modified using these bounds and setting $M = 500$ (Remark 2). We call this model (M4′).

Table 10 details the results for budget-constrained connection problems. In the first column, z^* denotes the optimal value of the instances without upgrading and the third column shows the optimal value with upgrading. The improvement that it represents over the initial optimal value is shown in the fourth column. Table 10 also shows the computing time of model (M4) and other elements necessary to apply the pre-processing procedure described in Section 3.2.1, that is, values of T and lower and upper bounds. As has been commented, we considered model (M2) to obtain the upper bounds ub shown in the table and, later, we solved model (M4′) using those bounds. Therefore, the total computing time to obtain the optimal solution with upgrading is the sum of the computing time needed to solve model (M2), plus the computing time needed to solve model (M4′), which is shown in the last column of Table 10. If we compare this last column with the computing times of model (M4), we can observe that they are, in most cases, much smaller, being on average more than 3 times faster using the pre-processing procedure. This happens despite the fact that the percentage of variables set to zero is rather small (Table 11) and, therefore, so is the number of values in the cost matrix that are modified. Note that the value of $R_{\max}$ directly influences the number of variables that are fixed to zero (see Section 3.2.1). We have considered a high value of this parameter ($R_{\max} = 0.8$) which means that there are few costs c_{ij} such that $(1 - R_{\max})c_{ij} > ub$. For smaller values of $R_{\max}$, the number of variables that would be fixed to zero could be much larger.

Similarly, for center upgrading, model (Q4) was solved as presented in Section 4.2 and, in addition, it was also solved using the pre-processing procedure described in Section 4.2.1. In this case, we obtained upper bounds using the model (U) described in Remark 12, starting from an optimal solution to the problem without upgrading. As before, the cost matrix was modified using the upper bounds and setting $M = 500$. Note that, in this case, we cannot use lower bounds either to fix variables to zero or to modify the cost matrix, because some connections, despite having a small cost, will be upgraded as they are incident to an upgraded center. We call this model (Q4′).

Table 12 details the results for budget-constrained center upgrading. Again, in the first column, z^* denotes the optimal value of the instances without upgrading. The third and fourth columns show the solution with upgrading and the improvement that it implies over the initial optimal value, respectively. Table 12 also shows the computing time of the model (Q4). Note that most problems exhaust the time limit of 6 h, so the final gap (%) is also shown.

To obtain the upper bound ub that is used for the pre-processing procedure of the model (Q4′), it is necessary to have a solution to the original problem without upgrading. For instances pmed1–pmed5 we can obtain an optimal solution to work within 4 seconds (see, e.g. Calik and Tansel [6]). Regarding the computing time required by the model (U) from Remark 12, it was less than 0.2 s in all cases. Given the negligible above values, we do not include in Table 12 the computing time necessary to obtain the upper bound. That same table shows the solutions, computing time, and final gap (%) of the model (Q4′). Although the pre-processing procedure is not as effective in the model (Q4′) as in other cases (there are still many instances that exhaust the time limit of 6 h), we can see really important improvements. For example, for pmed3 and $B = 3903.1$, we can obtain the optimal solution with both model (Q4) and model (Q4′) but their computing times are 7054.6 s and 15.5 s, respectively. We also see four other instances for which model (Q4′) was tackled reaching a final gap below 0.01%. Table 13 shows the number of variables set to zero in the pre-processing procedure of the model (Q4′).

Given that the number of instances that can be solved to proven optimality, both with models (Q4) and (Q4′) is small, we also address the budget-constrained center-upgrading model with the math-heuristic approach introduced in Section 5. After some preliminary experiments, we run the genetic algorithm for each instance in Table 12 for one hour (stopping condition in Algorithm 2) and we set $N = 25$, $probability = 0.1$, and the number of centers to exchange each selected center in the local searches equal to 10. In Table 12 we present the execution time in seconds required by our math-heuristic to find the best solution for each instance. Those best solutions and the improvement over the initial optimal value without upgrading are also shown.

As can be observed in the last three columns of Table 12, for the instances based upon pmed1 the math-heuristic manages to improve the best solution found by models (Q4) and (Q4′) for the four considered budgets. Moreover, this is accomplished in less than one minute. The same occurs with the instances based upon pmed2, pmed3, and pmed4, with average times of 5.6, 4.0, and 14.6 minutes, respectively. In particular, the math-heuristic is able to find the three optimal values that are known in instances pmed3 and pmed4.

Regarding pmed5, this is more demanding for the math-heuristic since each individual of the population is a combination of $p = 33$ centers, which makes the search much more difficult. However, in one of the instances it manages to improve the best solution found by models (Q4) and (Q4′) ($B = 276.0$). In another one, it equals the known optimal value ($B = 827.5$). Finally, in the other two instances, it obtains values very close to those provided

Table 10
Results for the budget-constrained connection upgrading. All computing times are displayed in seconds.

Base instance	Budget	Upgraded z^*	Improvement (%)	Time M4	T	lb	ub	Time M2	Time M4′	Time M2+M4′
pmed1	940.8	71.5	43.7	3831.5	3	20.2	113.0	201.7	4277.0	4478.8
$z^* = 127$	1881.4	50.0	60.6	1195.2	8	20.2	105.0	100.5	623.4	723.8
	2822.0	35.1	72.3	541.1	12	20.2	95.0	111.3	239.6	350.9
	3762.6	25.4	80.0	588.9	16	20.2	90.0	103.5	344.1	447.6
pmed2	766.6	52.1	46.8	1624.9	3	16.6	88.0	107.3	274.9	382.2
$z^* = 98$	1532.9	35.5	63.8	981.8	6	16.6	82.0	93.0	390.6	483.6
	2299.3	23.6	75.9	970.5	9	16.6	79.0	119.2	410.9	530.1
	3065.6	19.6	80.0	1458.9	13	16.6	73.0	105.1	456.7	561.9
pmed3	834.5	52.7	43.3	19795.5	3	14.6	89.0	161.2	2948.2	3109.4
$z^* = 93$	1668.0	34.7	62.7	969.5	7	14.6	77.0	195.2	533.1	728.3
	2503.0	21.8	76.6	4732.3	10	14.6	72.0	112.6	31.0	143.6
	3337.2	18.6	80.0	804.6	14	14.6	68.0	149.5	505.7	655.2
pmed4	578.3	42.7	42.3	4820.6	2	11.2	73.0	106.4	281.5	388.0
$z^* = 74$	1156.0	28.0	62.2	1130.4	4	11.2	70.0	103.8	154.9	258.7
	1733.7	18.0	75.6	1016.9	7	11.2	66.0	88.9	31.9	120.8
	2311.4	14.8	80.0	373.5	10	11.2	61.0	94.6	108.2	202.8
pmed5	224.4	28.4	40.8	6213.0	0	7.6	48.0	8.1	1466.8	1474.9
$z^* = 48$	448.6	19.3	59.8	300.1	1	7.6	46.0	59.8	101.6	161.4
	672.8	13.0	72.8	130.1	2	7.6	44.0	69.6	28.6	98.1
	897.0	9.6	80.0	180.2	3	7.6	40.0	63.7	52.0	115.7
Avg.				2583.0	Avg.					770.8

Table 11
Percentage of variables x_{ij}, r_{ij} that are set to zero in model (M4′) for instances pme1–pmed5.

pmed1			pmed2			pmed3			pmed4			pmed5		
Budget	x_{ij}	r_{ij}	Budget	x_{ij}	r_{ij}	Budget	x_{ij}	r_{ij}	Budget	x_{ij}	r_{ij}	Budget	x_{ij}	r_{ij}
940.8	0.0	1.8	766.6	0.0	1.8	834.5	0.0	1.6	578.3	0.0	1.3	224.4	0.7	2.2
1881.4	0.0	1.8	1532.9	0.0	1.8	1668.7	0.0	1.7	1156.0	0.0	1.3	448.6	1.3	2.7
2822.0	0.0	1.8	2299.3	0.0	1.8	2503.0	0.0	1.7	1733.7	0.0	1.3	672.8	2.0	3.5
3762.6	0.0	1.8	3065.6	0.0	1.8	3337.2	0.1	1.7	2311.4	0.1	1.4	897.0	4.7	6.1

Table 12
Results for the budget-constrained center upgrading. The computing times that do not exhaust the time limit of 6 h are displayed in seconds.

		Model Q4					Model Q4′				Math-heuristic		
Base instance	Budget	Upgraded z^*	Improvement (%)	Time	Gap (%) at termination	ub	Upgraded z^*	Improvement (%)	Time	Gap (%) at termination	Upgraded z^*	Improvement (%)	Time
pmed1	1175.0	102.2	19.5	> 6 h	43.4	110.0	102.2	19.5	> 6 h	43.4	91.0	28.4	57.9
$z^* = 127$	2349.8	71.0	44.1	> 6 h	44.4	75.1	71.0	44.1	> 6 h	44.4	64.7	49.0	46.5
	3524.6	46.2	63.6	> 6 h	44.5	52.1	46.1	63.7	> 6 h	46.5	43.8	65.5	48.6
	4699.4	26.6	79.1	> 6 h	27.3	29.1	29.0	77.1	> 6 h	33.7	25.4	80.0	32.1
pmed2	935.2	73.2	25.3	> 6 h	44.6	78.0	72.4	26.2	> 6 h	44.0	66.7	32.0	264.1
$z^* = 98$	1870.2	50.2	48.8	> 6 h	45.5	56.5	49.1	49.9	> 6 h	44.2	45.1	54.0	678.5
	2805.2	29.0	70.5	> 6 h	42.2	38.7	28.5	70.9	> 6 h	40.7	27.3	72.2	294.8
	3740.2	20.8	78.8	> 6 h	12.5	21.1	19.9	79.7	> 6 h	2.5	19.6	80.0	104.2
pmed3	975.9	68.6	26.2	> 6 h	40.2	85.0	68.6	26.2	> 6 h	40.2	65.7	29.4	137.2
$z^* = 93$	1951.6	48.0	48.4	> 6 h	44.9	54.8	48.0	48.4	> 6 h	43.1	44.7	51.9	639.9
	2927.4	26.9	71.1	> 6 h	38.4	36.8	26.9	71.1	> 6 h	38.4	26.7	71.3	184.1
	3903.1	18.6	80.0	7054.6	0.0	18.8	18.6	80.0	15.5	0.0	18.6	80.0	0.0
pmed4	667.7	53.0	28.4	> 6 h	43.6	57.0	58.0	21.6	> 6 h	48.1	52.0	29.7	496.3
$z^* = 74$	1334.7	36.0	51.4	> 6 h	41.7	41.7	37.0	50.0	> 6 h	43.2	35.0	52.7	1287.3
	2001.8	21.4	71.1	> 6 h	38.9	28.4	20.8	71.8	20772.1	0.0	20.8	71.8	1368.9
	2668.9	15.8	78.6	> 6 h	20.7	15.4	14.8	80.0	12.1	0.0	14.8	80.0	359.1
pmed5	276.0	35.0	27.1	> 6 h	56.9	37.0	34.6	27.8	> 6 h	56.4	33.0	31.3	407.2
$z^* = 48$	551.7	23.0	52.1	> 6 h	52.9	27.2	22.6	52.9	> 6 h	50.5	22.7	52.8	2663.7
	827.5	14.2	70.4	> 6 h	49.1	19.7	14.2	70.5	2346.5	0.0	14.2	70.5	1448.6
	1103.3	9.6	79.9	> 6 h	4.5	12.7	9.6	80.0	246.8	0.0	9.8	79.6	1381.9

Table 13
Percentage of variables x_{ij}, r_{ij} that are set to zero in model (Q4′) for instances pme1–pmed5.

pmed1			pmed2			pmed3			pmed4			pmed5		
Budget	x_{ij}	r_{ij}	Budget	x_{ij}	r_{ij}	Budget	x_{ij}	r_{ij}	Budget	x_{ij}	r_{ij}	Budget	x_{ij}	r_{ij}
1175.0	0.0	0.0	935.2	0.0	0.0	975.9	0.0	0.0	667.7	0.4	0.4	276.0	8.7	8.7
2349.8	0.0	0.0	1870.2	0.2	0.2	1951.6	1.4	1.4	1334.7	15.9	15.9	551.7	38.4	38.4
3524.6	0.9	0.9	2805.2	14.7	14.7	2927.4	20.1	20.1	2001.8	60.8	60.8	827.5	69.1	69.1
4699.4	47.6	47.6	3740.1	72.8	72.8	3903.1	82.2	82.2	2668.9	91.1	91.1	1103.3	89.2	89.2

Table 14
Average computing time (minutes) of the math-heuristic, according to the value of n, for $p = 5$, 10.

p	$n = 100$	$n = 200$	$n = 300$	$n = 400$	$n = 500$	$n = 600$	$n = 700$	$n = 800$	$n = 900$
5	0.8	9.8	11.0	92.1	51.8	62.5	58.3	189.8	136.7
10	5.6	16.0	71.1	75.0	143.3	149.5	212.0	301.3	225.3

by models (Q4) and (Q4′), on an average computing time of 24.6 minutes.

The promising results achieved by the math-heuristic for the instances based upon pmed1–pmed5 ($n = 100$), encouraged testing it using the larger instances. For the latter we keep using four different values for the available budget in each instance. Note, however, that for instances with $n > 100$ the values of $B^F_{\max}$ cannot be obtained in a reasonable computing time using the auxiliary model based on model (Q4) and explained at the end of Section 4.2. For this reason, we decided for the use of an auxiliary model based on the linear optimization model (U) presented in Remark 12. In particular, for each instance, we obtain an upper bound of $B^F_{\max}$ by solving the following linear program:

$$\begin{aligned} \text{minimize} \quad & \sum_{i \in \hat{I}} \theta_i g_i & \\ \text{subject to} \quad & (1 - R_{max}) z^* \geq \beta_i (1 - g_i) & \forall i \in \hat{I}, \\ & g_i \in \{0\} \cup [R_{min}, R_{max}] & \forall i \in \hat{I}, \end{aligned}$$

where z^* is the optimal value without upgrading and $\hat{I} = \{i \in I : y_i = 1\}$ is the subset of p open centers also in that case.

All the math-heuristic parameters are maintained as previously described for instances based upon pmed1–pmed5, except for the stopping condition. In fact, the large-scale instances are more challenging. Accordingly, we set the stopping criterion (time limit) dependent on the value of n. After some preliminary experiments, we set a time limit of two hours for instances with $n = 200$, three hours for $n = 300$, procedding likewise and ending with a time limit of nine hours for $n = 900$.

Table 14 shows the average computing time required by the math-heuristic to find the best solution for all problem sizes considering $p = 5$ and $p = 10$—the values of p used in all values of n. As can be observed, the computing time increases considerably both with the value of p and n. Detailed results of all instances with $n > 100$ can be found in the Appendix, Tables D1 and D2. In these tables we conclude that, in most cases, the math-heuristic did not exhaust the time limit to find the provided solution. Specifically, in only 16% of the instances the time exceeds 95% of the time limit given as a stopping criterion. Additionally, most of these cases occur for intermediate budget values, where it is more difficult for the algorithm to reach a good solution. In Tables D1 and D2, the execution time to find the best solution is zero in most cases when the budget allows an improvement of 80% because the algorithm does not improve the upgraded solution associated to the non-upgrading problem that we include at the initial population. For all instances, the results provided by the math-heuristic, even if they are not optimal solutions, provide hopefully good upper bounds for the problems.

Fig. 10 shows the average computing time of the math-heuristic for solving all the instances based upon pmed1–pmed40. Again, it can be observed how the time increases with the value of n, as well as that, when the value of p increases, the computing time stabilizes for a given n.

In Fig. 11 we summarize the results obtained for the budget-constrained problems for the instances based upon pmed1–pmed5. Both the connection-upgrading problems and the center-upgrading problems are represented in each sub-figure. For center upgrading, since the optimal value is unknown in some cases, the best solution found (Table 12) is used in Fig. 11. We mark with a dot each combination “budget” vs. “percentage improvement in the optimal value”. Since four values are considered for the budget in each case we observe four dots in each sub-figure. We decided to connect the dots with lines to see if some stabilization trend could be observed. This is in fact the case: in every sub-figure, we can ob-

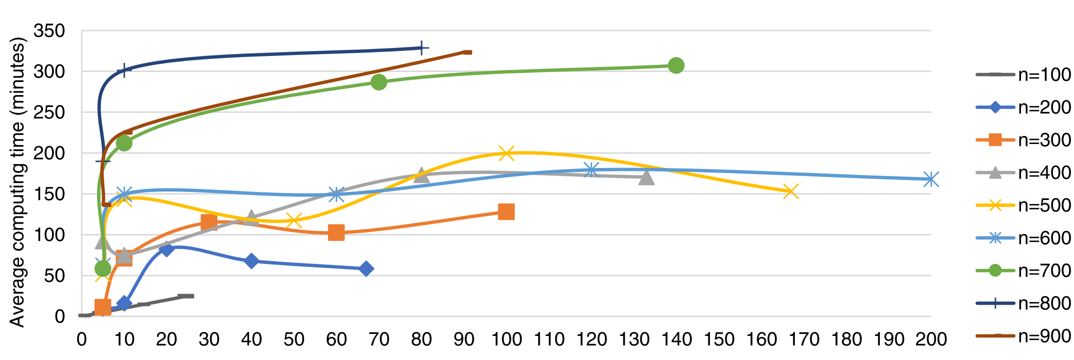
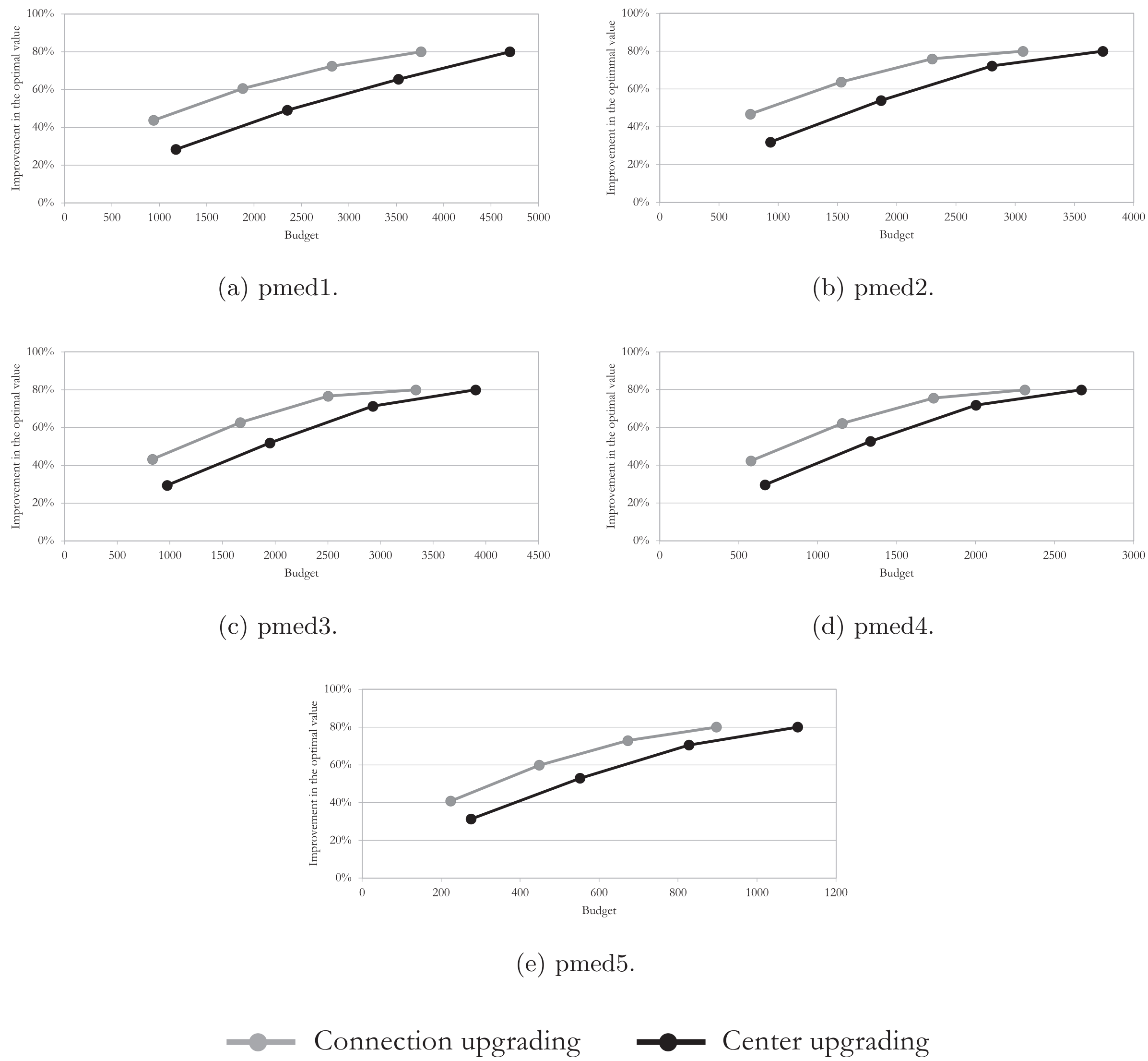


Fig. 10. Average computing time (minutes) of the math-heuristic for solving the instances based upon pmed1–pmed40.



(a) pmed1.

(b) pmed2.

(c) pmed3.

(d) pmed4.

(e) pmed5.

Fig. 11. Improvement observed in the optimal value for different budget values.

serve a stabilization in the improvement of the optimal cost when the budget grows large. This is very insightful information for a decision-maker who can decide on a maximum budget of interest.

7. Conclusions

In this paper, we investigated different upgrading strategies for the discrete p-center problem. In particular, we looked into whether a better solution can be achieved by reducing in some way the allocation costs, thus obtaining the so-called upgraded solutions. We considered the possibility of upgrading a set of connections to different centers as well as the possibility of upgrading entire centers, that is, upgrading all connections to an open center. Two variants of these problems were considered: (i) a limit is imposed on the number of connections or centers that can be upgraded; (ii) a budget exists that limits the upgrades that can be made. MILP models were introduced for the problems and their variants. Furthermore, lower and upper bounds and optimal solution properties were discussed. The models and the new bounds were tested using benchmark instances. Due to the difficulty in tackling the budget-constrained center-upgrading model investigated in this paper, we also proposed a math-heuristic approach to this problem. Specifically, we developed a genetic algorithm for finding good feasible solutions in a short time.

The major conclusion drawn from all the work done is that a significant decrease in the optimal cost can be attained by upgrading connections or centers. Therefore, the information provided by the new models proposed in this work can be extremely useful to a decision-maker because together with the location decision, the models directly seek to find underlying structures of the problem that can be "upgraded" in such a way that a better after-upgrading solution is obtained.

The research done in this work indicates different directions for future work in the topic. First, despite the instances that could be solved to proven optimality using the models proposed, it is important to deepen the polyhedral analysis of these models by deriving new valid inequalities that can strengthen them. This is crucial for later deriving more comprehensive models that can capture features of practical relevance (e.g. time-dependent decisions). Second, results reported in the paper show that there is room for improving both the lower and upper bounds for this type of problem. Finally, a close look into the budget-constrained models reveals a bi-criteria flavor: in fact, the models proposed seek to minimize the maximum cost for satisfying the demand nodes by imposing a limit on the cost compression. Thus, a relevant direction for further research involves studying specifically a bi-objective setting for the problem when a budget constraint exists.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

CRediT authorship contribution statement

Laura Anton-Sanchez: Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Mercedes Landete:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing. **Francisco Saldanha-da-Gama:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Methodology, Validation, Visualization, Writing – original draft, Writing – review & editing.

Data availability

Data will be made available on request.

Acknowledgments

This work was partially supported by the grants PID2021-122344NB-I00, PID2019-105952GB-I00/ AEI /10.13039/501100011033, and PGC2018-099428-B-100 by the Spanish Ministry of Science and Innovation, PROMETEO/2021/063 by the governments of Spain and the Valencian Community, project CIGE/2021/161 by the Valencian Community, and UIDB/04561/2020 by National Funding from FCT — Fundação para a Ciência e Tecnologia, Portugal.

The authors thank the anonymous reviewers for the very useful insights they provided that strongly contributed to an improved manuscript.

Appendix A. Detailed results for connection upgrading

In this Appendix we detail the results of the experiments performed using models M1, M1′, M2 + $\hat{K}_1$, M2 + $\hat{K}_2$, M3+$\hat{K}_1$ and M3+$\hat{K}_2$ for the instances generated from pmed1 to pmed5.

Table A1
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed1. The optimal value without upgrading is 127.0.

t	f	z^*	M1	M1′	M2 + $\hat{K}_1$	M2 + $\hat{K}_2$	M3+$\hat{K}_1$	M3+$\hat{K}_2$
5	0.2	108.0	748.5	232.2	36.6	30.2	176.9	139.7
	0.4	108.0	523.0	631.8	93.1	189.9	211.7	270.3
	0.6	108.0	367.4	145.3	296.4	183.0	152.0	164.9
	0.8	108.0	323.8	200.3	279.7	334.0	185.4	210.6
10	0.2	102.4	415.0	102.7	75.4	93.3	146.7	136.8
	0.4	100.0	488.0	237.3	106.4	151.0	185.4	202.6
	0.6	100.0	638.3	323.5	235.3	211.6	212.4	183.5
	0.8	100.0	1087.7	261.4	256.8	377.4	265.1	134.7
25	0.2	101.6	192.0	142.1	47.7	103.6	415.0	147.4
	0.4	81.0	506.1	417.6	70.5	140.1	248.5	172.1
	0.6	79.0	696.0	242.2	69.2	149.0	293.3	210.7
	0.8	79.0	504.6	394.3	223.7	197.6	196.4	139.6
Avg.			540.9	277.6	149.2	180.1	224.1	176.1

Table A2
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed2. The optimal value without upgrading is 98.0.

t	f	z^*	M1	M1′	M2 + $\hat{K}_1$	M2 + $\hat{K}_2$	M3+$\hat{K}_1$	M3+$\hat{K}_2$
5	0.2	88.0	856.1	124.9	22.8	95.0	175.3	213.5
	0.4	83.0	367.5	223.0	164.0	142.1	193.0	276.2
	0.6	83.0	460.5	320.0	136.8	152.1	218.2	243.5
	0.8	83.0	409.5	115.8	179.2	203.0	232.0	243.0
10	0.2	81.6	240.2	12.8	73.8	48.1	126.1	102.4
	0.4	78.0	263.3	112.1	60.5	75.1	275.1	273.2
	0.6	78.0	458.7	230.7	175.2	161.4	262.6	268.8
	0.8	78.0	465.1	212.4	564.0	234.1	213.1	190.2
25	0.2	78.4	89.0	11.8	18.5	34.6	148.4	73.2
	0.4	61.2	84.9	102.9	58.0	59.8	172.0	89.8
	0.6	54.0	375.1	135.2	61.9	75.7	234.3	213.8
	0.8	54.0	314.1	74.9	125.1	144.2	203.4	187.7
Avg.			365.3	139.7	136.7	118.7	204.5	197.9

Table A3
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed3. The optimal value without upgrading is 93.0.

t	f	z^*	M1	M1′	M2 + $\hat{K}_1$	M2 + $\hat{K}_2$	M3+$\hat{K}_1$	M3+$\hat{K}_2$
5	0.2	86.0	56.8	92.0	101.4	50.1	83.0	114.9
	0.4	85.0	463.1	95.4	172.3	124.2	155.6	152.3
	0.6	85.0	321.4	251.6	140.9	353.5	257.2	440.7
	0.8	85.0	430.6	114.3	172.2	380.5	299.4	256.9
10	0.2	77.6	271.6	94.7	65.7	41.3	119.6	137.6
	0.4	75.0	622.8	28.9	211.4	128.6	196.9	203.4
	0.6	73.0	441.1	41.5	151.7	213.7	344.5	343.9
	0.8	72.0	329.2	245.1	380.1	189.2	347.3	333.3
25	0.2	74.4	174.9	23.6	36.2	14.6	93.4	74.2
	0.4	61.0	409.0	109.2	60.4	83.3	279.2	160.3
	0.6	59.0	410.5	106.3	222.9	171.1	449.3	587.4
	0.8	57.0	417.0	124.7	404.1	234.0	405.8	374.7
Avg.			362.3	110.6	176.6	165.4	252.6	265.0

Table A4
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed4. The optimal value without upgrading is 74.0.

t	f	z^*	M1	M1′	M2 + $\hat{K}_1$	M2 + $\hat{K}_2$	M3+$\hat{K}_1$	M3+$\hat{K}_2$
5	0.2	67.0	78.5	11.7	8.9	6.2	27.2	36.6
	0.4	67.0	220.6	33.4	18.6	10.6	54.2	77.0
	0.6	67.0	69.8	23.7	65.3	75.3	156.3	176.6
	0.8	67.0	490.7	177.2	156.6	122.4	276.2	258.4
10	0.2	63.0	80.2	2.0	5.6	7.4	27.9	16.1
	0.4	61.0	362.4	33.7	17.4	24.7	45.1	35.3
	0.6	61.0	447.6	57.6	70.6	78.1	135.8	135.7
	0.8	61.0	419.3	81.6	139.2	205.3	221.9	293.5
25	0.2	59.2	75.2	4.9	3.8	2.6	24.4	10.5
	0.4	49.0	130.1	10.1	16.1	8.5	55.4	30.5
	0.6	45.0	157.4	11.5	103.8	50.2	157.4	82.9
	0.8	45.0	298.3	44.5	108.4	206.1	417.6	418.7
Avg.			235.8	41.0	59.5	66.5	133.3	131.0

Table A5
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed5. The optimal value without upgrading is 48.0.

t	f	z^*	M1	M1′	M2 + $\hat{K}_1$	M2 + $\hat{K}_2$	M3+$\hat{K}_1$	M3+$\hat{K}_2$
5	0.2	40.0	106.0	5.7	3.5	2.0	22.1	12.0
	0.4	40.0	58.8	3.6	8.6	4.5	41.7	25.8
	0.6	40.0	57.2	15.2	34.4	17.4	100.4	52.6
	0.8	40.0	81.2	131.7	82.3	82.9	291.2	161.5
10	0.2	38.4	54.8	2.4	2.3	1.4	24.4	6.6
	0.4	36.0	65.8	14.4	9.6	4.1	61.2	13.4
	0.6	36.0	79.8	6.6	31.0	13.6	88.7	42.6
	0.8	36.0	85.0	115.7	89.1	79.7	199.5	177.1
25	0.2	38.4	59.4	1.2	3.5	1.4	21.6	10.6
	0.4	28.8	62.6	1.9	4.9	2.3	45.9	15.5
	0.6	21.0	69.4	6.5	14.0	5.3	78.1	26.1
	0.8	19.0	107.2	24.9	50.7	44.5	241.2	125.1
Avg.			73.9	27.5	27.8	21.6	101.3	55.7

Table A6
Percentage of variables x_{ij}, m_{ij} that are set to zero in model (M1′) for instances pme1–pmed5, using bounds $lb2$ and $ub2$ for connection upgrading.

		pmed1		pmed2		pmed3		pmed4		pmed5	
t	f	x_{ij}	m_{ij}	x_{ij}	m_{ij}	x_{ij}	m_{ij}	x_{ij}	m_{ij}	x_{ij}	m_{ij}
5	0.2	12.1	24.5	18.1	27.8	32.5	39.5	80.3	83.7	83.1	86.4
	0.4	0.4	7.5	1.4	7.0	6.8	11.5	59.4	61.9	63.5	66.2
	0.6	0.0	3.4	0.0	3.2	0.1	3.0	13.0	14.7	22.0	24.1
	0.8	0.0	1.8	0.0	1.8	0.0	1.6	0.0	1.3	0.0	1.5
10	0.2	12.9	25.3	22.7	32.4	39.4	46.4	81.9	85.3	84.7	88.1
	0.4	0.6	7.6	2.5	8.1	9.9	14.5	62.8	65.3	69.1	71.8
	0.6	0.0	3.4	0.0	3.2	0.4	3.3	16.4	18.1	29.4	31.5
	0.8	0.0	1.8	0.0	1.8	0.0	1.6	0.0	1.3	0.0	1.5
25	0.2	17.0	29.3	37.5	47.2	49.4	56.4	86.7	90.1	84.7	88.1
	0.4	4.0	11.0	16.0	21.6	20.6	25.2	75.9	78.4	81.3	84.0
	0.6	0.0	3.4	0.1	3.3	1.4	4.3	35.6	37.3	49.3	51.4
	0.8	0.0	1.8	0.0	1.8	0.0	1.6	0.0	1.3	0.4	1.9

Table A7
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed1.

t	f	z^*	M1	M1′	M2	M3
5	0.2	108.0	21.4	14.1	25.1	12.2
	0.4	108.0	26.2	22.9	43.6	15.8
	0.6	108.0	30.1	28.9	62.1	16.8
	0.8	108.0	33.5	33.1	80.8	17.4
10	0.2	102.4	21.0	13.9	21.0	13.3
	0.4	100.0	27.3	24.1	39.2	17.3
	0.6	100.0	33.6	32.4	59.2	20.1
	0.8	100.0	38.5	37.9	79.4	21.3
25	0.2	101.6	26.8	16.8	20.4	18.9
	0.4	81.0	25.5	20.2	25.0	14.5
	0.6	79.0	36.6	34.2	48.6	21.6
	0.8	79.0	45.5	44.7	74.2	25.7
Avg.			30.5	26.9	48.2	17.9

Table A8
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed2.

t	f	z^*	M1	M1′	M2	M3
5	0.2	88.0	32.0	15.3	24.3	12.7
	0.4	83.0	31.7	21.5	39.8	12.7
	0.6	83.0	35.2	29.5	59.5	15.5
	0.8	83.0	38.3	36.1	79.6	17.1
10	0.2	81.6	29.6	12.0	18.4	11.6
	0.4	78.0	33.2	22.9	36.0	15.6
	0.6	78.0	39.1	33.9	57.1	20.0
	0.8	78.0	44.1	42.1	78.4	22.1
25	0.2	78.4	32.8	12.3	15.3	15.1
	0.4	61.2	28.7	14.8	18.6	9.6
	0.6	54.0	34.1	26.3	38.3	11.9
	0.8	54.0	45.6	42.7	69.1	19.5
Avg.			35.4	25.8	44.5	15.3

Table A9
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed3.

t	f	z^*	M1	M1′	M2	M3
5	0.2	86.0	30.9	19.0	31.8	15.0
	0.4	85.0	33.5	25.6	48.2	17.6
	0.6	85.0	36.6	32.0	65.0	18.8
	0.8	85.0	39.3	37.3	82.4	19.7
10	0.2	77.6	26.5	13.8	24.5	12.2
	0.4	75.0	30.3	22.3	41.5	15.3
	0.6	73.0	34.1	29.5	59.5	16.1
	0.8	72.0	38.4	36.5	79.4	16.8
25	0.2	74.4	29.5	14.4	21.4	17.2
	0.4	61.0	27.7	18.5	28.1	14.1
	0.6	59.0	38.5	32.9	50.2	21.9
	0.8	57.0	47.3	45.2	74.2	24.2
Avg.			34.4	27.2	50.5	17.4

Table A10
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed4.

t	f	z^*	M1	M1′	M2	M3
5	0.2	67.0	40.5	19.7	33.0	11.1
	0.4	67.0	42.8	29.2	49.6	15.8
	0.6	67.0	45.0	36.7	66.1	18.4
	0.8	67.0	47.3	43.5	82.9	20.7
10	0.2	63.0	38.8	17.5	28.8	13.2
	0.4	61.0	41.4	27.4	44.7	17.0
	0.6	61.0	45.9	37.6	62.9	20.1
	0.8	61.0	50.3	46.7	81.3	23.1
25	0.2	59.2	40.1	16.6	24.3	20.2
	0.4	49.0	38.7	21.5	31.3	17.0
	0.6	45.0	44.6	34.9	49.9	19.3
	0.8	45.0	55.1	50.9	74.9	27.3
Avg.			44.2	31.8	52.5	18.6

Table A11
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed5.

t	f	z^*	M1	M1′	M2	M3
5	0.2	40.0	54.3	17.0	23.7	8.9
	0.4	40.0	56.4	30.3	42.9	17.4
	0.6	40.0	58.4	42.7	61.6	20.9
	0.8	40.0	60.3	53.1	80.7	24.3
10	0.2	38.4	54.4	16.2	20.7	14.0
	0.4	36.0	55.7	27.7	36.6	19.9
	0.6	36.0	59.8	43.0	57.4	24.4
	0.8	36.0	63.4	55.8	78.6	29.1
25	0.2	38.4	58.3	18.2	20.8	20.8
	0.4	28.8	54.7	18.4	20.8	20.8
	0.6	21.0	51.8	23.0	27.4	14.6
	0.8	19.0	60.0	46.6	59.7	21.3
Avg.			57.3	32.7	44.2	19.7

Table A12
Percentage decrease in the optimal cost by upgrading connections.

t	f	pmed1	pmed2	pmed3	pmed4	pmed5
5	0.2	−15.0	−10.2	−7.5	−9.5	−16.7
	0.4	−15.0	−15.3	−8.6	−9.5	−16.7
	0.6	−15.0	−15.3	−8.6	−9.5	−16.7
	0.8	−15.0	−15.3	−8.6	−9.5	−16.7
10	0.2	−19.4	−16.7	−16.6	−14.9	−20.0
	0.4	−21.3	−20.4	−19.4	−17.6	−25.0
	0.6	−21.3	−20.4	−21.5	−17.6	−25.0
	0.8	−21.3	−20.4	−22.6	−17.6	−25.0
25	0.2	−20.0	−20.0	−20.0	−20.0	−20.0
	0.4	−36.2	−37.6	−34.4	−33.8	−40.0
	0.6	−37.8	−44.9	−36.6	−39.2	−56.3
	0.8	−37.8	−44.9	−38.7	−39.2	−60.4

Appendix B. Detailed results for center upgrading

In this Appendix we detail the results of the experiments performed using models Q1, Q1′, Q2+$\hat{K}_1$, Q2+$\hat{K}_2$, Q3+$\hat{K}_1$ and Q3+$\hat{K}_2$ for the instances generated from pmed1 to pmed5.

Table B1
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed1. The optimal value without upgrading is 127.0.

t	f	z^*	Q1	Q1′	Q2+$\hat{K}_1$	Q2+$\hat{K}_2$	Q3+$\hat{K}_1$	Q3+$\hat{K}_2$
1	0.2	110.0	2186.8	460.0	35.8	117.2	206.5	84.7
	0.4	92.4	1106.9	392.3	205.6	42.3	380.2	122.1
	0.6	64.4	520.6	217.9	148.7	62.0	203.1	114.3
	0.8	32.2	237.3	112.8	37.9	37.8	106.6	90.6
2	0.2	106.4	682.0	507.6	74.3	67.4	977.8	466.4
	0.4	82.8	2217.9	87.2	47.6	33.7	156.1	175.7
	0.6	56.4	3228.0	105.7	23.6	92.9	139.6	209.8
	0.8	28.4	273.5	111.7	37.8	45.5	98.2	172.1
Avg.			1306.6	249.4	76.4	62.3	283.5	179.5

Table B2
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed2. The optimal value without upgrading is 98.0.

t	f	z^*	Q1	Q1′	Q2+$\hat{K}_1$	Q2+$\hat{K}_2$	Q3+$\hat{K}_1$	Q3+$\hat{K}_2$
1	0.2	90.0	702.8	44.8	13.3	20.6	58.9	104.1
	0.4	76.8	1372.6	593.9	25.4	27.4	68.7	115.7
	0.6	55.2	634.7	157.3	34.2	49.9	83.1	130.8
	0.8	29.0	220.0	118.9	35.3	58.6	71.6	163.8
3	0.2	83.0	410.6	123.5	58.2	49.0	534.9	626.8
	0.4	65.4	148.4	720.2	23.3	23.6	122.0	206.5
	0.6	48.0	176.5	842.9	29.0	49.5	130.5	157.0
	0.8	24.0	1731.2	236.7	53.5	48.3	147.8	190.2
Avg.			674.6	354.8	34.0	40.9	152.2	211.9

Table B3
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed3. The optimal value without upgrading is 93.0.

t	f	z^*	Q1	Q1′	Q2+$\hat{K}_1$	Q2+$\hat{K}_2$	Q3+$\hat{K}_1$	Q3+$\hat{K}_2$
1	0.2	91.0	934.0	195.9	34.3	24.5	57.1	114.7
	0.4	82.0	1018.1	600.0	102.7	86.6	114.2	166.7
	0.6	60.4	1326.4	304.4	51.9	43.7	165.1	205.7
	0.8	33.0	387.7	145.0	66.7	60.3	192.9	339.2
3	0.2	79.2	865.6	16.6	34.3	24.7	162.7	199.0
	0.4	69.6	1262.6	988.8	17.0	77.1	132.3	302.6
	0.6	48.0	1148.7	642.7	29.4	39.9	180.9	232.5
	0.8	24.2	4683.8	1170.6	39.4	52.2	222.8	216.4
Avg.			1453.4	508.0	47.0	51.1	153.5	222.1

Table B4
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed4. The optimal value without upgrading is 74.0.

t	f	z^*	Q1	Q1′	Q2+$\hat{K}_1$	Q2+$\hat{K}_2$	Q3+$\hat{K}_1$	Q3+$\hat{K}_2$
1	0.2	70.0	99.1	49.2	4.1	3.2	10.4	11.7
	0.4	67.0	653.9	330.3	12.8	8.9	23.9	20.2
	0.6	54.0	1213.4	224.8	61.8	34.2	105.9	76.1
	0.8	30.2	248.4	156.2	50.5	40.9	135.4	192.9
2	0.2	67.0	217.6	61.5	5.7	3.1	11.6	10.6
	0.4	61.0	390.4	233.2	23.2	8.1	35.0	38.2
	0.6	45.6	125.7	326.2	25.1	17.2	103.6	129.3
	0.8	24.8	435.2	362.7	35.8	30.7	172.3	313.5
5	0.2	63.2	418.4	2.8	4.2	3.7	88.7	78.3
	0.4	48.6	91.9	12.0	9.3	7.8	63.1	72.3
	0.6	34.8	371.8	48.9	84.1	14.4	310.4	229.7
	0.8	19.4	423.0	400.2	49.1	29.1	141.1	219.9
Avg.			390.7	184.0	30.5	16.8	100.1	116.1

Table B5
Optimal objective function value (z^*) and computing time (seconds) for the instances generated from pmed5. The optimal value without upgrading is 48.0.

t	f	z^*	Q1	Q1′	Q2+$\hat{K}_1$	Q2+$\hat{K}_2$	Q3+$\hat{K}_1$	Q3+$\hat{K}_2$
1	0.2	43.0	88.4	36.6	4.1	2.6	12.3	11.1
	0.4	40.0	530.1	93.5	9.5	5.2	22.4	27.8
	0.6	33.2	160.7	111.3	12.4	17.7	73.4	66.5
	0.8	18.8	176.4	150.6	34.8	20.7	106.2	126.5
2	0.2	40.0	59.6	13.8	3.3	2.5	9.9	9.6
	0.4	36.6	216.0	20.0	7.6	4.8	22.4	37.3
	0.6	28.0	362.1	116.6	27.9	14.0	56.6	66.2
	0.8	15.4	203.8	285.3	38.4	25.1	109.1	193.7
3	0.2	40.0	74.4	12.9	3.4	2.1	9.3	12.1
	0.4	36.0	339.1	36.4	8.8	5.2	43.3	44.3
	0.6	24.4	130.0	55.2	20.6	8.4	49.8	45.9
	0.8	13.2	329.4	376.1	35.0	19.2	106.9	192.1
8	0.2	38.4	91.4	2.1	3.0	1.6	32.9	33.5
	0.4	28.8	95.2	9.3	4.6	2.8	41.5	23.5
	0.6	19.6	146.6	87.8	11.1	5.0	59.3	69.6
	0.8	10.0	290.7	93.7	35.8	13.5	102.1	139.8
Avg.			205.9	93.8	16.3	9.4	53.6	68.7

Table B6
Percentage of variables x_{ij}, m_{ij} that are set to zero in model (Q1′) for instances pme1–pmed5, using bounds $lb2$ and $ub2$ for center upgrading.

		pmed1			pmed2			pmed3			pmed4			pmed5	
f	t	x_{ij}	m_{ij}	t	x_{ij}	m_{ij}	t	x_{ij}	m_{ij}	t	x_{ij}	m_{ij}	t	x_{ij}	m_{ij}
0.2	1	9.9	22.3	1	17.5	27.2	1	25.3	32.3	1	80.6	84.1	1	77.3	80.7
0.4		0.2	7.3		1.2	6.8		4.3	8.9		60.8	63.4		54.4	57.1
0.6		0.0	3.4		0.0	3.2		0.0	2.9		14.3	16.0		13.0	15.1
0.8		0.0	1.8		0.0	1.8		0.0	1.6		0.0	1.3		0.0	1.5
0.2	2	10.9	23.2	3	24.6	34.3	3	37.8	44.8	3	81.0	84.5	2	80.7	84.0
0.4		0.3	7.3		3.3	8.9		9.0	13.7		61.4	64.0		59.4	62.1
0.6		0.0	3.4		0.0	3.2		0.3	3.2		15.1	16.8		17.8	19.9
0.8		0.0	1.8		0.0	1.8		0.0	1.6		0.0	1.3		0.0	1.5
0.2										5	82.2	85.6	3	80.7	84.0
0.4											63.9	66.5		59.4	62.1
0.6											17.4	19.1		17.8	19.9
0.8											0.0	1.3		0.0	1.5
0.2													8	84.7	88.1
0.4														67.2	69.9
0.6														27.2	29.3
0.8														0.0	1.5

Table B7
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed1.

t	f	z^*	Q1	Q1′	Q2	Q3
1	0.2	110.0	26.5	16.4	26.5	14.3
	0.4	92.4	29.0	20.6	34.1	15.7
	0.6	64.4	29.0	20.2	36.7	12.6
	0.8	32.2	27.4	18.3	36.4	10.1
2	0.2	106.4	28.6	17.4	24.0	17.0
	0.4	82.8	28.9	19.5	26.5	14.0
	0.6	56.4	28.8	19.2	27.9	12.5
	0.8	28.4	28.6	19.6	28.3	11.7
Avg.			28.3	18.9	30.0	13.5

Table B8
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed2.

t	f	z^*	Q1	Q1′	Q2	Q3
1	0.2	90.0	35.6	17.3	25.9	13.4
	0.4	76.8	36.9	21.7	34.9	14.6
	0.6	55.2	35.8	22.0	39.1	13.2
	0.8	29.0	35.6	23.3	42.0	13.6
3	0.2	83.0	34.9	14.8	19.8	13.9
	0.4	65.4	34.3	17.1	23.7	13.0
	0.6	48.0	37.7	23.2	30.4	16.3
	0.8	24.0	36.7	24.1	30.4	14.3
Avg.			35.9	20.4	30.8	14.1

Table B9
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed3.

t	f	z^*	Q1	Q1′	Q2	Q3
1	0.2	91.0	36.1	22.1	35.4	16.9
	0.4	82.0	37.2	26.2	46.3	18.5
	0.6	60.4	36.2	26.7	50.9	17.0
	0.8	33.0	38.7	30.8	55.0	21.2
3	0.2	79.2	31.7	15.6	25.9	13.9
	0.4	69.6	37.7	24.2	36.9	19.4
	0.6	48.0	37.9	25.4	38.6	18.3
	0.8	24.2	37.9	27.0	39.1	18.0
Avg.			36.7	24.8	41.0	17.9

Table B10
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed4.

t	f	z^*	Q1	Q1′	Q2	Q3
1	0.2	70.0	44.1	23.0	35.9	13.4
	0.4	67.0	46.8	31.7	49.5	20.4
	0.6	54.0	46.4	35.3	57.9	23.5
	0.8	30.2	44.0	35.1	62.2	23.6
2	0.2	67.0	43.8	21.2	33.0	13.5
	0.4	61.0	47.0	29.6	44.6	21.2
	0.6	45.6	46.7	32.7	50.3	21.9
	0.8	24.8	46.0	33.9	54.2	22.3
5	0.2	63.2	43.8	19.3	29.0	17.0
	0.4	48.6	42.2	19.7	30.6	14.6
	0.6	34.8	43.4	23.7	35.1	16.9
	0.8	19.4	46.4	31.5	41.8	20.7
Avg.			45.1	28.1	43.7	19.1

Table B11
Optimal objective function value (z^*) and LP gap (%) for the instances generated from pmed5.

t	f	z^*	Q1	Q1′	Q2	Q3
1	0.2	43.0	57.7	21.7	28.9	12.8
	0.4	40.0	58.5	31.0	42.8	21.5
	0.6	33.2	58.8	39.3	53.7	26.9
	0.8	18.8	56.0	40.5	59.1	27.5
2	0.2	40.0	56.0	17.2	23.7	11.1
	0.4	36.6	58.2	28.1	37.5	21.4
	0.6	28.0	57.7	34.2	45.2	23.6
	0.8	15.4	55.8	36.4	50.1	24.9
3	0.2	40.0	57.0	18.3	23.7	14.4
	0.4	36.0	59.8	29.2	36.5	24.6
	0.6	24.4	55.4	28.0	37.2	19.0
	0.8	13.2	54.1	31.4	41.9	20.2
8	0.2	38.4	58.7	18.0	20.8	18.6
	0.4	28.8	57.3	17.6	20.8	16.8
	0.6	19.6	56.2	19.7	22.2	16.0
	0.8	10.0	55.8	24.5	23.7	15.3
Avg.			57.1	27.2	35.5	19.7

Table B12
Percentage decrease in the optimal cost by upgrading centers.

f	t	pmed1	t	pmed2	t	pmed3	t	pmed4	t	pmed5
0.2	1	−13.4	1	−8.2	1	−2.2	1	−5.4	1	−10.4
0.4		−27.2		−21.6		−11.8		−9.5		−16.7
0.6		−49.3		−43.7		−35.1		−27.0		−30.8
0.8		−74.6		−70.4		−64.5		−59.2		−60.8
0.2	2	−16.2	3	−15.3	3	−14.8	3	−9.5	2	−16.7
0.4		−34.8		−33.3		−25.2		−17.6		−23.8
0.6		−55.6		−51.0		−48.4		−38.4		−41.7
0.8		−77.6		−75.5		−74.0		−66.5		−67.9
0.2							5	−14.6	3	−16.7
0.4								−34.3		−25.0
0.6								−53.0		−49.2
0.8								−73.8		−72.5
0.2									8	−20.0
0.4										−40.0
0.6										−59.2
0.8										−79.2

Appendix C. Detailed results for all instances

In this Appendix we present detailed results concerning the computing time (seconds) required by models (M2) and (Q2). Each row in Table C1 presents average results for the instances built from the pmed instance heading the row. We remove from the average the instances such that one of the models or both exhaust the time limit of 6 h, i.e., we do not take into account the computing time of these instances for any of the models when obtaining the average values. In this table we also present the percentage deviation of the average time using the lower and upper bounds lb_2 and ub_2 compared to lb_1 and ub_1. A positive deviation indicates that by using the latter bounds, a smaller average computing time was obtained.

Table C1
Average computing time (seconds) using models (M2) (connection upgrading) and (Q2) (center upgrading).

Instance	n	p	M2 + $\hat{K}_1$	M2 + $\hat{K}_2$	Deviation (%)	Q2+$\hat{K}_1$	Q2+$\hat{K}_2$	Deviation (%)
pmed1	100	5	149.2	180.1	20.7	76.4	62.3	−18.4
pmed2		10	136.7	118.7	−13.1	34.0	40.9	20.1
pmed3		10	176.6	165.4	−6.4	47.0	51.1	8.8
pmed4		20	59.5	66.5	11.6	30.5	16.8	−44.9
pmed5		33	27.8	21.6	−22.4	16.3	9.4	−42.3
pmed6	200	5	1703.3	1280.1	−24.8	625.9	443.2	−29.2
pmed7		10	623.1	584.4	−6.2	1061.1	787.3	−25.8
pmed8		20	625.2	502.0	−19.7	745.7	401.3	−46.2
pmed9		40	298.8	187.2	−37.4	651.6	517.9	−20.5
pmed10		67	96.7	78.4	−18.9	143.0	119.7	−16.3
pmed11	300	5	965.2	662.9	−31.3	893.4	1166.8	30.6
pmed12		10	2264.5	1312.7	−42.0	2107.2	1149.8	−45.4
pmed13		30	1051.4	1146.2	9.0	1772.1	1879.3	6.1
pmed14		60	485.2	421.1	−13.2	1200.5	1179.0	−1.8
pmed15		100	246.3	146.1	−40.7	434.7	405.8	−6.7
pmed16	400	5	1744.6	1124.1	−35.6	1122.3	722.0	−35.7
pmed17		10	2796	2009.4	−28.1	2515.0	1909.9	−24.1
pmed18		40	2736.1	1574.3	−42.5	6777.1	4344.2	−35.9
pmed19		80	1639.6	520.6	−68.2	1763.6	1761.7	−0.1
pmed20		133	576.0	311.5	−45.9	873.8	794.3	−9.1
pmed21	500	5	3259.4	1821.0	−44.1	2730.9	3123.3	14.4
pmed22		10	8341.4	6501.6	−22.1	6236.0	6624.0	6.2
pmed23		50	2217.4	2999.4	35.3	4239.2	4103.0	−3.2
pmed24		100	963.8	1057.5	9.7	2022.9	1924.5	−4.9
pmed25		167	457.7	823.3	79.9	667.7	613.7	−8.1
pmed26	600	5	6414.6	6363.0	−0.8	8811.6	4593.7	−47.9
pmed27		10	9545.0	5668.8	−40.6	6974.0	6085.4	−12.7
pmed28		60	4251.4	2852.0	−32.9	5840.7	5501.5	−5.8
pmed29		120	1733.3	2290.7	32.2	5137.3	4165.4	−18.9
pmed30		200	932.7	2201.8	136.1	2586.5	1713.5	−33.8
pmed31	700	5	5297.1	4774.4	−9.9	5731.6	4291.3	−25.1
pmed32		10	9706.3	8725.7	−10.1	14112.1	9182.7	−34.9
pmed33		70	4419.0	5040.7	14.1	7948.8	6573.3	−17.3
pmed34		140	2538.3	1959.1	−22.8	3578.2	3497.2	−2.3
pmed35	800	5	7044.4	4863.3	−31.0	5180.0	4670.0	−9.8
pmed36		10	7906.5	8369.2	5.9	6224.1	10954.4	76.0
pmed37		80	8472.7	6439.0	−24.0	8851.2	7184.7	−18.8
pmed38	900	5	10787.5	9988.9	−7.4	7888.6	4523.0	−42.7
pmed39		10	11595.3	9371.0	−19.2	14939.0	8245.7	−44.8
pmed40		90	4507.8	5985.7	32.8	10029.3	7451.7	−15.7

Appendix D. Math-heuristic results for instances with $n > 100$

In this Appendix we detail the results of the experiments performed using the math-heuristic proposed in Section 5. Tables D1 and D2 show the results for the instances based upon pmed6–pmed40 ($n > 100$). For each instance, the number of nodes, n, the number of open centers, p, and the optimal value without upgrading, z^*, are shown. Each instance was solved with four different budgets, therefore the tables also show, for each instance, the solution with upgrading for each budget, the improvement that it implies over the initial optimal value, and the execution time in seconds to find the best solution.

Table D1
Math-heuristic results for the instances based upon pmed6–pmed25.

n	p	Base instance	Budget	Upgr. z^*	Impr. (%)	Time	n	p	Base instance	Budget	Upgr. z^*	Impr. (%)	Time
200	5	pmed6	1783.3	61.3	27.0	738.7	300	5	pmed11	1895.4	40.0	32.2	262.7
		$z^* = 84$	3566.5	43.1	48.7	1147.3			$z^* = 59$	3790.6	26.2	55.7	1332.8
			5349.6	25.7	69.4	474.5				5685.8	14.6	75.3	1033.0
			7132.7	16.8	80.0	0.0				7581.0	11.8	80.0	0.0
	10	pmed7	1278.9	47.9	25.1	2019.5		10	pmed12	1622.1	35.0	31.4	2200.4
		$z^* = 64$	2557.5	31.6	50.6	1140.2			$z^* = 51$	3243.9	23.1	54.8	9421.6
			3836.1	19.2	69.9	680.3				4865.8	13.2	74.1	5449.3
			5114.8	12.8	80.0	0.0				6487.6	10.2	80.0	0.0
	20	pmed8	1075.8	35.1	36.2	5850.1		30	pmed13	1082.8	26.0	27.8	7893.9
		$z^* = 55$	2151.4	23.6	57.1	7056.6			$z^* = 36$	2165.4	17.0	52.9	9286.9
			3227.0	14.2	74.1	6883.1				3248.0	9.2	74.5	10361.1
			4302.6	11.0	80.0	0.0				4330.6	7.2	80.0	0.0
	40	pmed9	691.6	25.0	32.4	5326.5		60	pmed14	707.9	18.0	30.8	4108.5
		$z^* = 37$	1383.0	15.6	57.8	6478.5			$z^* = 26$	1415.7	12.1	53.4	9713.7
			2074.4	8.8	76.2	4465.4				2123.4	7.0	73.1	10740.8
			2765.8	7.4	80.0	0.0				2831.1	5.2	80.0	0.0
	67	pmed10	275.3	14.0	30.0	278.9		100	pmed15	424.8	12.6	30.2	10514.4
		$z^* = 20$	550.4	8.8	56.1	6661.4			$z^* = 18$	849.5	8.2	54.5	9944.9
			825.5	5.3	73.5	7039.9				1274.1	4.9	73.0	10248.6
			1100.6	4.0	80.0	0.0				1698.7	3.6	80.0	0.0
400	5	pmed16	1747.2	34.0	27.7	606.4	500	5	pmed21	2105.3	28.1	29.8	4534.0
		$z^* = 47$	3494.1	23.6	49.7	8360.3			$z^* = 40$	4210.4	19.0	52.6	3428.7
			5241.1	15.2	67.6	13140.0				6315.5	11.3	71.7	4459.9
			6988.1	9.4	80.0	0.0				8420.6	8.0	80.0	0.0
	10	pmed17	1621.5	28.0	28.2	1372.0		10	pmed22	2041.8	26.0	31.5	16650.1
		$z^* = 39$	3242.9	19.3	50.4	5849.1			$z^* = 38$	4083.4	17.7	53.3	13394.3
			4864.2	11.1	71.6	10778.8				6125.0	10.6	72.1	4339.5
			6485.5	7.8	80.0	0.0				8166.6	7.6	80.0	0.0
	40	pmed18	1223.0	21.0	25.1	12539.9		50	pmed23	1210.0	17.0	22.7	1006.2
		$z^* = 28$	2445.9	13.1	53.1	13405.2			$z^* = 22$	2419.8	10.4	52.7	17938.8
			3668.7	8.0	71.4	3157.0				3629.6	6.4	70.9	9289.0
			4891.5	5.6	80.0	0.0				4839.4	4.4	80.0	0.0
	80	pmed19	709.7	13.7	23.8	13793.5		100	pmed24	739.2	11.0	26.7	13484.5
		$z^* = 18$	1419.2	8.4	53.3	14153.3			$z^* = 15$	1478.2	7.6	49.3	17399.2
			2128.7	5.0	72.4	13617.8				2217.2	4.5	69.7	17038.4
			2838.2	3.6	80.0	0.0				2956.2	3.0	80.0	0.0
	133	pmed20	401.4	9.5	26.8	13593.1		167	pmed25	432.2	8.0	27.3	2602.0
		$z^* = 13$	802.7	6.7	48.7	13723.3			$z^* = 11$	864.2	5.4	51.2	17658.4
			1203.9	4.3	66.9	13641.7				1296.2	3.4	69.1	16453.7
			1605.2	2.6	80.0	0.0				1728.1	2.2	80.0	0.0

Table D2
Math-heuristic results for the instances based upon pmed26–pmed40.

n	p	Base instance	Budget	Upgr. z^*	Impr. (%)	Time	n	p	Base instance	Budget	Upgr. z^*	Impr. (%)	Time
600	5	pmed26	2412.4	25.9	31.7	11873.4	700	5	pmed31	2156.5	23.0	23.3	3023.6
		$z^* = 38$	4824.7	17.2	54.8	1689.9			$z^* = 30$	4312.8	16.4	45.3	1784.8
			7236.9	9.6	74.7	1436.9				6469.0	10.3	65.5	9183.8
			9649.1	7.6	80.0	0.0				8625.3	6.0	80.0	0.0
	10	pmed27	1925.2	23.0	28.1	3785.4		10	pmed32	2289.3	20.4	29.6	24942.1
		$z^* = 32$	3850.3	15.5	51.6	17600.4			$z^* = 29$	4578.5	13.6	53.0	19545.1
			5775.3	9.1	71.5	14485.1				6867.6	7.4	74.4	6391.1
			7700.3	6.4	80.0	0.0				9156.7	5.8	80.0	0.0
	60	pmed28	1136.6	14.0	22.2	1064.4		70	pmed33	1148.7	11.7	21.9	20365.6
		$z^* = 18$	2273.1	8.6	52.3	21354.7			$z^* = 15$	2297.3	8.0	46.8	23948.6
			3409.5	5.2	71.1	13419.8				3446.0	4.7	68.5	24500.7
			4546.0	3.6	80.0	0.0				4594.7	3.0	80.0	0.0
	120	pmed29	738.4	10.0	23.1	1875.7		140	pmed34	755.9	8.2	25.9	24773.7
		$z^* = 13$	1476.6	6.8	48.0	20240.1			$z^* = 11$	1511.9	5.7	48.0	24729.3
			2214.8	4.0	69.2	20923.3				2267.8	3.5	67.8	24203.1
			2953.1	2.6	80.0	0.0				3023.8	2.2	80.0	0.0
	200	pmed30	465.8	7.0	22.2	0.0							
		$z^* = 9$	931.4	4.9	45.8	19763.6							
			1397.0	3.1	65.0	20566.8							
			1862.6	1.8	80.0	0.0							
800	5	pmed35	2274.9	21.3	29.0	16108.7	900	5	pmed38	2632.0	21.0	27.6	728.8
		$z^* = 30$	4549.7	14.9	50.4	15148.1			$z^* = 29$	5264.0	13.2	54.4	17760.7
			6824.6	9.0	69.9	13709.7				7896.0	8.1	72.2	14322.4
			9099.4	6.0	80.0	582.0				10528.0	5.8	80.0	0.0
	10	pmed36	2207.9	20.0	25.9	19755.5		10	pmed39	2183.2	17.0	26.1	29499.1
		$z^* = 27$	4415.8	13.7	49.2	22984.6			$z^* = 23$	4366.5	11.7	49.2	14251.3
			6623.7	8.6	68.1	20554.0				6549.7	6.9	70.0	10325.3
			8831.6	5.4	80.0	9010.9				8733.0	4.6	80.0	0.0
	80	pmed37	1307.7	11.0	26.7	26633.0		90	pmed40	1265.8	10.0	23.1	15049.1
		$z^* = 15$	2615.4	7.7	48.8	25193.9			$z^* = 13$	2531.7	6.9	46.8	30973.5
			3923.0	4.3	71.1	27070.3				3797.5	4.0	69.2	31594.6
			5230.7	3.0	80.0	0.0				5063.3	2.6	80.0	0.0

Supplementary material

Supplementary material associated with this article can be found, in the online version, at doi:10.1016/j.omega.2023.102894.

References

[1] Akgün I, Gümüşbuğa F, Tansel B. Risk based facility location by using fault tree analysis in disaster management. Omega 2015;52:168–79. doi:10.1016/j.omega.2014.04.003.
[2] Baldomero-Naranjo M, Kalcsics J, Marín A, Rodríguez-Chía AM. Upgrading edges in the maximal covering location problem. Eur J Oper Res 2022;303(1):14–36. doi:10.1016/j.ejor.2022.02.001.
[3] Bell JE, Griffis SE, Cunningham WA, Eberlan JA. Location optimization of strategic alert sites for homeland defense. Omega 2011;39(2):151–8. doi:10.1016/j.omega.2010.05.004.
[4] Blanco V, Marín A. Upgrading nodes in tree-shaped hub location. Comput Oper Res 2019;102:75–90. doi:10.1016/j.cor.2018.10.007.
[5] Calik H, Labbé M, Yaman H. *p*-Center problems. In: Laporte G, Nickel S, Saldanha da Gama F, editors. Location science. Springer, International Publishing; 2019. p. 51–65. doi:10.1007/978-3-030-32177-2.
[6] Calik H, Tansel B. Double bound method for solving the *p*-center location problem. Comput Oper Res 2013;40:2991–9. doi:10.1016/j.cor.2013.07.011.
[7] Contreras I, Fernández E, Marín A. Tight bounds from a path based formulation for the tree of hub location problem. Comput Oper Res 2009;36:3117–27. doi:10.1016/j.cor.2008.12.009.
[8] Contreras I, Fernández E, Marín A. The tree of hubs location problem. Eur J Oper Res 2010;202:390–400. doi:10.1016/j.ejor.2009.05.044.
[9] Corberán A, Landete M, Peiró J, Saldanha-da-Gama F. The facility location problem with capacity transfers. Transp Res Part E 2020;138:101943. doi:10.1016/j.tre.2020.101943.
[10] Daskin M. Network and discrete location: models, algorithms, and applications. 2nd ed. John Wiley & Sons, Ltd; 2013. doi:10.1002/9781118537015.
[11] Dönmez Z, Kara BY, Özlem K, Saldanha-da-Gama F. Humanitarian facility location under uncertainty: critical review and future prospects. Omega 2021;102:102393. doi:10.1016/j.omega.2021.102393.
[12] Elloumi S, Labbé M, Pochet Y. A new formulation and resolution method for the *p*-center problem. INFORMS J Comput 2004;16:84–94. doi:10.1287/ijoc.1030.0028.
[13] Fadda E, Manerba D, Cabodi G, Camurati PE, Tadei R. Comparative analysis of models and performance indicators for optimal service facility location. Transp Res Part E 2021;145:102174.
[14] Gonzalez TF. Clustering to minimize the maximum intercluster distance. Theor Comput Sci 1985;38:293–306. doi:10.1016/j.tre.2020.102174.
[15] Hakimi S. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. Oper Res 1965;13:462–75. doi:10.1287/opre.13.3.462.
[16] Ibaraki T, Vaxés Y, Yang XG. Lowering eccentricity of a tree by node upgrading. Networks 2005;45:232–9.
[17] Kahr M. Determining locations and layouts for parcel lockers to support supply chain viability at the last mile. Omega 2022;113:102721.
[18] Karatas M, Eriskin L. Linear and piecewise linear formulations for a hierarchical facility location and sizing problem. Omega 2023;118:102850.
[19] Kariv O, Hakimi S. An algorithmic approach to network location problems. SIAM J Appl Math 1979;37:513–38.
[20] Katsigiannis F, Zografos K. Optimising airport slot allocation considering flight-scheduling flexibility and total airport capacity constraints. Transp Res Part B 2021;146:50–87. doi:10.1016/j.trb.2021.02.002.
[21] Lamberson L, Hocking R. Optimum time compression in project scheduling. Manag Sci 1970;16:597–606. doi:10.1287/mnsc.16.10.B597.
[22] Pelegrín M, Xu L. Continuous covering on networks: improved mixed integer programming formulations. Omega 2023;117:102835.
[23] Sepasian A. Upgrading the 1-center problem with edge length variables on a tree. Discrete Optim 2018;29:1–17. doi:10.1016/j.disopt.2018.02.002.
[24] Shioura A, Shakhlevich N, Strusevich V. Scheduling problems with controllable processing times and a common deadline to minimize maximum compression cost. J Global Optim 2020;76:471–90. doi:10.1007/s10898-018-0686-2.
[25] Stienen V, Wagenaar J, den Hertog D, Fleuren H. Optimal depot locations for humanitarian logistics service providers using robust optimization. Omega 2021;104:102494. doi:10.1016/j.omega.2021.102494.
[26] Wang W, Wu S, Wang S, Zhen L, Qu X. Emergency facility location problems in logistics: status and perspectives. Transp Res Part E 2021;154:102465. doi:10.1016/j.tre.2021.102465.
[27] Yang B. Single machine rescheduling with new jobs arrivals and processing time compression. Int J Adv Manuf 2007;34:378–84. doi:10.1007/s00170-006-0590-7.