



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

On the complexity of the upgrading version of the Maximal Covering Location Problem

Citation for published version:

Baldomero-Naranjo, M, Kalcsics, J & Rodriguez-Chia, AM 2024, 'On the complexity of the upgrading version of the Maximal Covering Location Problem', *Networks*. <https://doi.org/10.1002/net.22207>

Digital Object Identifier (DOI):

[On the complexity of the upgrading version of the maximal covering location problem 10.1002/net.22207](https://doi.org/10.1002/net.22207)

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

Networks

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



On the complexity of the upgrading version of the Maximal Covering Location Problem

Marta Baldomero-Naranjo^{1*}, Jörg Kalcsics², Antonio M. Rodríguez-Chía³

¹ Departamento de Estadística y Ciencia de los Datos, Universidad Complutense de Madrid, Facultad de Estudios Estadísticos, Madrid, 28040, Spain, martbald@ucm.es, 0000-0002-0788-7412

² School of Mathematics, University of Edinburgh, Edinburgh, EH9 3FD, United Kingdom, joerg.kalcsics@ed.ac.uk, 0000-0002-5013-3448

³ Departamento de Estadística e Investigación Operativa, Universidad de Cádiz, Facultad de Ciencias, Puerto Real (Cádiz), 11510, Spain, antonio.rodriguezchia@uca.es, 0000-0002-1708-2108

January 18, 2024

Abstract

In this paper, we study the complexity of the upgrading version of the maximal covering location problem with edge length modifications on networks. This problem is NP-hard on general networks. However, in some particular cases, we prove that this problem is solvable in polynomial time. The cases of star and path networks combined with different assumptions for the model parameters are analysed.

In particular, we obtain that the problem on star networks is solvable in $O(n \log n)$ time for uniform weights and NP-hard for non-uniform weights. On paths, the single facility problem is solvable in $O(n^3)$ time, while the p -facility problem is NP-hard even with uniform costs and upper bounds (maximal upgrading per edge), as well as, integer parameter values. Furthermore, a pseudo-polynomial algorithm is developed for the single facility problem on trees with integer parameters.

Keywords: Location; covering problems; networks; upgrading problems; complexity.

Statements and Declarations

The authors do not have any financial and personal relationships with other people or organizations that could inappropriately influence (bias) our work.

*Corresponding author: martbald@ucm.es (M. Baldomero-Naranjo)

1 Introduction

The Maximal Covering Location Problem (MCLP) is a well-known location problem analysed by several researchers in the last decades since its introduction in Church and ReVelle (1974). The aim of this problem is, given a set of demand points, to locate a fixed number of facilities to maximise the amount of covered demand. A demand point is considered to be covered if its distance to a facility is smaller than or equal to a given coverage radius.

Since the seminal paper by Church and ReVelle (1974), many variants of this problem have been analysed depending on the supporting space, e.g., recently in the discrete version (Arana-Jiménez et al., 2020, Cordeau et al., 2019, García and Marín, 2019, Marín et al., 2018), on networks (Baldomero-Naranjo et al., 2021, Berman et al., 2016, Fröhlich et al., 2020), and in a continuous space (Bansal and Kianfar, 2017, Blanco and Gázquez, 2021, Yang et al., 2020).

In this paper, we follow an algorithmic approach to deal with the *upgrading* version of the MCLP (Up-MCLP). The upgrading maximal covering location problem aims to find the best locations for p service facilities and the edge length reduction within the given budget to cover the maximum demand. In this context, upgrading an edge means reducing its length (within certain limits) at a given cost which is proportional to the extent of the reduction. Thus, it has to be decided on which edges to invest (allocate part of the budget) in order to maximise the coverage. This problem has several practical applications.

One real-life application of this problem is when a government wants to enhance the accessibility of public services for its citizens, such as healthcare centers, educational institutions, or social welfare facilities. Since improving accessibility is closely related to distances, one approach to achieving this goal is to invest in infrastructure to reduce travel times to these services. This typically involves a combination of opening new facilities and improving transportation infrastructure, e.g., upgrading roads to highways, adding new lanes, and enhancing public transportation by incorporating high-speed lines, dedicated bus routes, and increasing service frequency.

In the private sector, telecommunication, gas, and electricity companies face a similar challenge when they aim to expand their coverage. Further application of this problem can be found in areas like shopping centers or airports. The objective is to strategically position services such as defibrillators, ATM, information posts, etc., while simultaneously building additional passenger conveyors or escalators. The goal is to ensure that the maximum number of people are within a reasonable walking distance of these facilities.

As a consequence of its wide range of applications, studying the upgrading version of classical problems is a recent trend that interests many researchers. The motivation of upgrading problems is that in some applications, the parameters of a network can be modified so as to obtain better solutions. Two key parameters of a network are demand

weights and edge (arc) lengths, and allowing to adjust them means that they are decision variables of the model. Accordingly, we can consider different versions of upgrading, as follows.

- Upgrading nodes considering that the weight of the nodes can be modified (usually decreased) subject to a prespecified budget. One possible application is the problem of locating a hospital to cover the demand of the cities of a region. The demand in each city for hospital services is represented by a weight and the demand may be reduced if a primary care center is located in the city (which involves a cost and there is a budget for this purpose). The following are a few examples of problems where this version has been applied: the 1-median problem (Gassner, 2007), the 1-center problem (Gassner, 2009), the Euclidean 1-median problem (Plastria, 2016), the p -median problem (Sepasian and Rahbarnia, 2015), and the network delay minimization problem (Medya et al., 2018).
- Upgrading nodes such that the length of all edges incident to this node is reduced. One of the applications is to improve the access to a facility subject to a cost. This type of upgrading has recently been studied in relation to the p -center problem (Anton-Sanchez et al., 2023). In this paper, the authors do not consider an underlying network and its individual edges, but instead, they only consider direct connections between customers and facilities as a whole and upgrade those (additionally, upgrading a connection for one customer has no effect on the connection of another customer). This type of upgrading has also been applied in other combinatorial optimization problems, such as: the spanning tree problem (Álvarez-Miranda and Sinnl, 2017), in communication and signal flow problems (Paik and Sahni, 1995), and in the hub-location problem (Blanco and Marín, 2019), among others.
- Discrete upgrades over edges (arcs) considering that the edge (arc) length can be reduced incrementally in fixed steps, with each step reducing the length by a (potentially) different amount and at a different cost. For example, each step could reduce the edge length by 10%, up to a maximum of 50%, with each upgrade step becoming progressively more expensive to implement. If there is just one step, then the decision is merely whether or not to upgrade the edge (arc). This type of upgrading has recently been studied in relation to the p -center problem with direct connection upgrades (Anton-Sanchez et al., 2023). Discrete upgrades have also been applied for other combinatorial optimization problems, such as: the minimum cost flow problem (Büsing et al., 2017), the minimization of the maximum travel time (Campbell et al., 2006), the accessibility arc problem (Maya Duque et al., 2013), and the Graphical TSP (Landete et al., 2023), among others.

- Continuous upgrades over edges (arcs) considering that the length of the edges (arcs) can be modified by any increment subject to a prespecified budget. For covering problems, this has recently been considered by Baldomero-Naranjo et al. (2022), who proposed several mixed-integer programming formulations to solve the problem exactly. Other facility location problems where this approach has been studied include: the 1-center problem (Sepasian, 2018), the p -center problem with direct connection upgrades (Anton-Sanchez et al., 2023), and the p -median problem (Afrashteh et al., 2020, Espejo and Marín, 2023). Also, this has been applied in combinatorial optimization problems, such as: the min-max spanning tree problem (Sepasian and Monabbati, 2017), the minimum flow cost problems (Demgensky et al., 2002), the maximal shortest path interdiction problem (Zhang et al., 2021).

To the best of our knowledge, this is the first attempt to analyse the complexity of the MCLP with continuous edge upgrades for different types of graphs. It is known that the MCLP is NP-hard on a general network even without upgrades (Megiddo et al., 1983). Therefore, the Up-MCLP is also NP-hard on a general network, since the MCLP is a special case of the Up-MCLP that has the maximal upgrade per edge equal to zero for all edges. However, the classical MCLP is polynomial-time solvable on trees (Megiddo et al., 1983) and consequently also on stars and paths (Hassin and Tamir, 1991). The open question we address in this work is whether it is possible to prove similar results for the Up-MCLP.

Therefore, in this paper, we turn our attention to particular networks such as stars, paths, and trees and study the complexity of the Up-MCLP and provide polynomial and pseudo-polynomial time algorithms under different assumptions for the parameters. In particular, we prove the following results.

- On star networks, the p -facility problem is polynomial time solvable for uniform weights, while for non-uniform weights already the single facility problem is NP-hard even with uniform costs and upper bounds as well as integer parameter values.
- On paths, the single facility problem is solvable in polynomial time, while the p -facility problem is NP-hard even with uniform costs and upper bounds as well as integer parameter values.
- On trees, the single facility weighted problem with integer parameters has a pseudo-polynomial algorithm; note that this problem is NP-hard (as a star is also a tree).

Similar results have been obtained for various location problems, including covering problems (Bhattacharya and Nandy, 2013, Hartmann et al., 2022, Hassin and Tamir, 1991, Megiddo et al., 1983), center problems (Wang and Zhang, 2021), median problems (Afrashteh et al., 2019, Oudjit and Stallmann, 2021), or other facility location problems (Kalcsics

et al., 2015, Lari et al., 2018, Nguyen and Teh, 2023, Puerto et al., 2018), as well as flow/shortest path problems (Danilovic et al., 2021, Hoppmann-Baum, 2022, Šuvak et al., 2021).

A summary of the obtained results can be found in Table 1, where the first column indicates the complexity of the Up-MCLP, the second column shows the graph type, the third column points out the number of facilities, the three following columns provide information on the parameters of the problem: uniformity of weights (unif. w), upgrading costs (unif. c), and upper bounds on the amount of upgrade on an edge (unif. u), respectively. The last column indicates the theorem in which the result is proven. Note that “Y” represents that the uniformity on the respective parameters is assumed, “N” represents that the non-uniformity on the respective parameters is assumed, and blank space means that no assumptions are made on the corresponding uniformity of the parameters.

| Complexity | Graph | Num. facilities | Unif. w | Unif. c | Unif. u | Result |
|------------|-------|-----------------|-----------|-----------|-----------|-----------|
| Polynomial | Star | p | Y | | | Theorem 1 |
| | Path | 1 | | | | Theorem 4 |
| NP-hard | Star | 1 | N | Y | Y | Theorem 3 |
| | Path | p | N | Y | Y | Theorem 5 |

Table 1: Summary of complexity results.

The rest of the paper is structured as follows. In Section 2 the problem is introduced. Section 3 analyses the problem on star networks, a polynomial time algorithm is derived for uniform weights and its NP-hardness is proven for non-uniform weights. In Section 4, a polynomial time algorithm is presented for the single facility problem on path networks. Moreover, we prove that the p -facility problem is NP-hard even with uniform costs and upper bounds and integer parameters values. In Section 5 a pseudo-polynomial algorithm is presented for the single facility problem on tree networks with integer parameters. Lastly, our conclusions are presented in Section 6.

2 Definitions and Problem Description

Let $N = (V, E, \ell)$ be an undirected network with node set $V = \{1, \dots, n\}$ and edge set E , where $|E| = m$. Slightly abusing notation, sometimes we denote the nodes also as v_1, \dots, v_n instead of $1, \dots, n$. Every edge $e = (k, q) = (q, k) \in E$, $k, q \in V$, has a positive length $\ell_e = \ell_{(k,q)}$. For $i, j \in V$, $d(i, j)$ is the length of the shortest path connecting i with j . Furthermore, we are given a fixed coverage radius $R > 0$. We say that a node $i \in V$ is *covered* by a facility at node j if $d(i, j) \leq R$. Finally, for each node $i \in V$ we are given a non-negative *weight* w_i that specifies the demand at the node.

The length ℓ_e of each edge $e \in E$ can be reduced by an amount lower than or equal to $u_e \in [0, \ell_e]$, $e \in E$. Without loss of generality, we assume that $\ell_e - u_e \leq R$, for $e \in E$ (if that were not the case, i.e., there was an edge $e \in E$ such that $\ell_e - u_e > R$, then e can be removed from the network without affecting the optimal solution). Moreover, reducing the length of edge $e \in E$ by an amount $\delta_e \in [0, u_e]$ comes at a positive cost of $\delta_e c_e$, for $c_e \in \mathbb{R}^+$, and there is a budget constraint $B \in \mathbb{R}^+$ on the overall cost of reduction. Again, without loss of generality, we assume that $c_e u_e \leq B$, for $e \in E$ (if that were not the case, i.e., there was a cost c_e for $e \in E$ such that $c_e u_e > B$, then u_e can be substituted by $u_e = B/c_e$ without affecting the optimal solution since it is not possible to pay for a reduction whose cost is greater than the budget). For short, the labels (ℓ_e, u_e, c_e) have been given to the edges representing the length, the maximal reduction, and the cost per unit of reduction of each edge. Finally, we assume that facilities can only be located at nodes.

The upgrading maximal covering location problem (Up-MCLP) aims to locate p service facilities covering the maximum demand taking into account that the total cost for the edge length reductions is within the given budget.

Let $\delta = (\delta_e)_{e \in E}$ denote a vector of edge length reductions, $0 \leq \delta_e \leq u_e$, for $e \in E$. For $i, j \in V$ and reductions δ , $d(i, j, \delta)$ is the length of a shortest path between i and j in the network after the reductions δ have been applied. Moreover, we call i reachable from j (and vice versa), if we can find an upgrade δ of the edges such that i is covered by j , i.e., $d(i, j, \delta) \leq R$. Finally, for $p \in \mathbb{N}$, let $X_p \subseteq V$ denote a set of p nodes, let (X_p, δ) represent a feasible solution of the Up-MCLP, and let $C(X_p, \delta) = \{i \in V \mid \exists j \in X_p : d(i, j, \delta) \leq R\}$ denote the set of all nodes covered by a facility in X_p after the edge upgrades δ . Then, the p -Up-MCLP can be formulated as:

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

For a given weight threshold $T \in \mathbb{R}^+$, we define the decision version p -Up-MCLP-D of p -Up-MCLP as:

Input: Network $N = (V, E, \ell)$, number of facilities p , coverage radius R , upgrading bounds and costs $(u_e)_{e \in E}$ and $(c_e)_{e \in E}$, respectively, and budget B .

Question: Does there exist a set $X_p \subseteq V$ and edge upgrades $\delta = (\delta_e)_{e \in E} \in \times_{e \in E} [0, u_e]$ with $|X_p| = p$ and $\sum_{e \in E} c_e \delta_e \leq B$ such that $\sum_{i \in C(X_p, \delta)} w_i \geq T$.

As previously discussed, the Up-MCLP is NP-hard on a general network. However, the classical MCLP is polynomial time solvable on trees and paths. Indeed, Megiddo et al. (1983) present an $O(n^2 p)$ algorithm for this problem on trees and Hassin and Tamir (1991) derive an $O(np)$ algorithm on paths. The objective of this paper is to analyse

the upgrading version of this problem in star, path, and tree networks to determine for which cases and under which assumptions on the demand weights w_i , upgrading costs c_e , or upgrading bounds u_e the problem can be solved in polynomial time. Hereinafter, the following expressions mean:

- *Uniform weights:* $w_i = w$ for all $i \in V$ and some $w \in \mathbb{R}^+$.
- *Uniform costs:* $c_e = c$ for all $e \in E$ and some $c \in \mathbb{R}^+$.
- *Uniform bounds:* $u_e = u$ for all $e \in E$ and some $u \in \mathbb{R}^+$.

In the next section, we focus on the star network case.

3 Star networks

In this section, we establish several complexity results for the problem on star networks depending on the characteristics of the input parameters. First, we present a result for the unweighted single facility problem.

Theorem 1 *The single facility maximal covering problem with edge length variations, 1-Up-MCLP, can be solved in $O(n \log n)$ time for uniform weights (unweighted case) on star networks.*

Proof. Let $N = (V, E, \ell)$ be a star network with central node v_0 and satellite nodes v_1, \dots, v_n . See Figure 1 for an illustration.

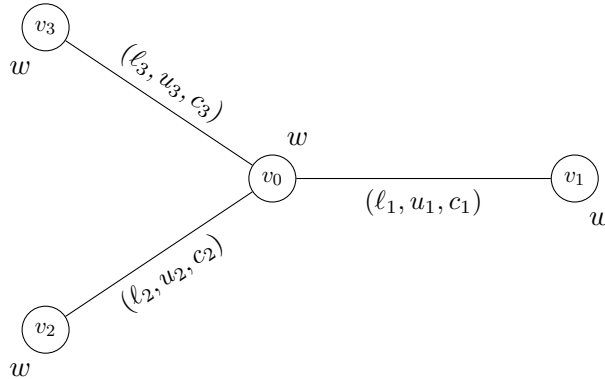


Figure 1: Illustration for the uniform case in Theorem 1, with edge labels (ℓ_j, u_j, c_j) .

First, we can assume without loss of generality that the facility is located in the central node, v_0 . Indeed, since the weights are uniform, locating a facility in a central node is at least as good as locating it in a satellite node as a facility located in a satellite node will never be able to cover more nodes than one that is located in the central node.

Then, we can find an optimal upgrading of the edges in polynomial time as follows. Let $\Delta_i = \ell_{(0,i)} - R$. If $\Delta_i \leq 0$, then v_i is automatically covered and can be discarded. If $\Delta_i > u_i$, then v_i is unreachable and can also be discarded. All remaining satellite nodes are sorted by non-decreasing minimal upgrading costs, i.e., by $c_i \Delta_i$. Then, starting with the first node v_i in the list, we upgrade the edge by $\delta_i = \min\{B/c_i, \Delta_i\}$ and update the budget $B = B - c_i \delta_i$. If $B > 0$, we continue with the next node in the list. Otherwise, we stop.

Indeed, this procedure provides an optimal solution. If this was not the case, then an optimal upgrade would provide a set of covered nodes \hat{V} , such that there exist $v_i \in \hat{V}$ and $v_k \in V \setminus \hat{V}$ satisfying $c_i \Delta_i > c_k \Delta_k$. However, upgrading edge e_k instead of e_i covers v_k without decreasing the total number of covered nodes and without increasing the amount of used budget. Concerning the last node v_j for which we upgrade the incident edge, if $B/c_j < \Delta_j$, then we cannot reduce the length of e_j enough to cover v_j . However, as we sorted the nodes by non-decreasing minimal upgrading costs, spending the remaining budget instead on some other edge e_k incident to another uncovered node v_k , $k \notin C(\{v_0\}, \delta)$, will also not allow us to cover that node. Hence, spending the remaining budget on e_j is as good as spending it on some other edge or even as good as not spending it at all, which concludes the argument.

The complexity of this procedure is dominated by the step of sorting the minimal upgrading costs, i.e., is $O(n \log n)$. All other steps are computed in constant time. Therefore, the unweighted 1-Up-MCLP can be solved in $O(n \log n)$ time on star networks. \square

After showing that the unweighted single facility problem can be solved in polynomial time, the next natural step is to study the p -facility case, i.e., p -Up-MCLP.

Theorem 2 *The unweighted p -Up-MCLP can be solved in $O(n \log n)$ time on star networks.*

Proof. Let $N = (V, E, \ell)$ be a star network with central node v_0 and satellite nodes v_1, \dots, v_n . Again, we can assume without loss of generality that in an optimal solution one of the facilities is located in the central node v_0 , as a facility located in a satellite node will never be able to cover more nodes than one that is located in the central node.

We start by using the same algorithm as in the proof of Theorem 1 to compute optimal edge upgrades δ^* for the case of just one facility being located at v_0 . Let $V^- = V \setminus C(\{v_0\}, \delta^*)$ be the set of nodes that are still uncovered after this upgrade. If $|V^-| > p - 1$, then since each node has the same weight and no facility in a satellite node can cover a node that could not already be covered by v_0 , putting facilities on $p - 1$ arbitrary nodes in V^- is optimal. If instead $|V^-| \leq p - 1$, then all nodes can be covered and the solution is trivially optimal.

Finally, for an arbitrary upgrade δ by the proof of Theorem 1 we have $|C(\{v_0\}, \delta)| \leq |C(\{v_0\}, \delta^*)|$. Therefore, even if we distribute the remaining $p - 1$ facilities optimally, we will not be able to cover more nodes using δ than using δ^* (again, because facilities in satellite nodes cannot cover more nodes than v_0 for a given upgrade).

The complexity of this construction is dominated by the step of sorting the minimal upgrading costs, i.e., $O(n \log n)$. Therefore, the unweighted p -Up-MCLP can be solved in $O(n \log n)$ time on star networks. □

Next, we discuss the case in which the node weights are not uniform.

Theorem 3 *The 1-Up-MCLP is NP-hard on star networks for non-uniform weights (weighted case) even assuming uniformity in upgrading costs and upper bounds as well as integrality of the input parameter values.*

Proof. We prove the result by reducing KNAPSACK to the decision problem 1-Up-MCLP-D. Let an instance of KNAPSACK be given with n items of positive weight $g_i \in \mathbb{N}$ and positive value $b_i \in \mathbb{N}$, $1 \leq i \leq n$, knapsack capacity $K \in \mathbb{N}$, and target value $U > 0$. A solution (i.e., a Yes-Input) to KNAPSACK is a set $M \subseteq \{1, \dots, n\}$ such that $\sum_{i \in M} g_i \leq K$ and $\sum_{i \in M} b_i \geq U$. Without loss of generality, we assume $g_i \leq K$, $1 \leq i \leq n$.

First, we observe that 1-Up-MCLP-D is in NP since a given solution for 1-Up-MCLP-D can be verified as such in polynomial time. Next, given an instance for KNAPSACK, we construct an instance of the edge upgrading problem in polynomial time as follows. Let $N = (V, E, \ell)$ be a star network with central node v_0 , satellite nodes v_1, \dots, v_n , and edges $e_i = (v_0, v_i)$. Each satellite node v_i corresponds to one item i , and the weight w_i of the node equals the item value b_i . The central node is given weight $w_0 = W > \sum_{i=1}^n b_i$. Finally, the weight threshold T for 1-Up-MCLP-D is defined as $T = W + U$.

Let c and u be the unit upgrading costs and, respectively, the upper bounds on the upgrade in each edge. We set $c = 1$, $u = \max_{i=1, \dots, n} g_i \leq K$, $R = u + 1$, and the length of each edge e_i as $\ell_i = R + g_i$. Finally, the budget equals the upgrading cost times the knapsack capacity, i.e., $B = K$. See Figure 2 for an illustration.

Let $(\{x\}, \delta)$ be a solution for 1-Up-MCLP-D. As $\sum_{i \in C(\{x\}, \delta)} w_i \geq T > W$, the central node must be covered. Therefore, we can assume without loss of generality that $x = v_0$. Let $M = C(\{v_0\}, \delta) \setminus \{v_0\}$ be the set of all satellite nodes covered by v_0 after the upgrade δ . A satellite node v_i is hereby covered if and only if $g_i \leq \delta_i \leq u$. As

$$\sum_{i \in M} \delta_i \leq \sum_{i=1}^n \delta_i \leq B = K$$

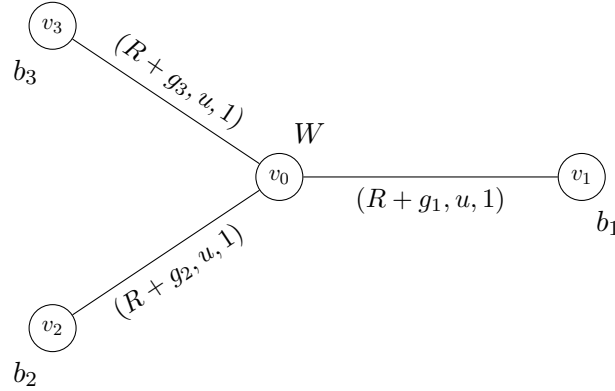


Figure 2: Illustration for the proof of Theorem 3, with edge labels (ℓ_j, u, c) .

we get $\sum_{i \in M} g_i \leq K$. Moreover, as

$$\sum_{i \in C(\{v_0\}, \delta)} w_i = W + \sum_{i \in M} w_i \geq T = W + U,$$

we obtain $\sum_{i \in M} w_i = \sum_{i \in M} b_i \geq U$ and M is a solution to KNAPSACK. Vice versa, for the same star network any solution M to KNAPSACK can easily be converted into a solution for 1-Up-MCLP-D by setting $X = \{v_0\}$ and $\delta_i = g_i$, $i \in M$.

As a result, the weighted 1-Up-MCLP-D is NP-complete on star networks and the weighted 1-Up-MCLP is NP-hard. \square

Consequently, the p -facility problem is also NP-hard. Therefore, from the analysis carried out in this section we have seen that the key feature for the Up-MCLP to be solvable or not in polynomial time on star networks is given by the uniformity or non-uniformity of weights.

4 Path networks

In this section, we analyse the complexity of Up-MCLP on path networks. First, we show that the 1-Up-MCLP is solvable in polynomial time on path networks.

The resolution process consists of checking all possible facility locations, i.e., all nodes of the path, and then calculating the resulting maximal coverage that can be obtained by upgrading edges. For each $i \in V$, we locate the facility on node i of the path. We denote this node in the following as v_0 . Let V^l and E^l (V^r and E^r) be the set of nodes and edges on the left-hand side (on the right-hand side) of v_0 , respectively. We number the nodes and edges in V^l and E^l consecutively, starting with v_1^l and e_1^l being adjacent and, respectively, incident to v_0 . Let, $V^l = \{v_1^l, v_2^l, \dots, v_s^l\}$ and $E^l = \{e_1^l, e_2^l, \dots, e_s^l\}$, where v_s^l is the left

most node in V^l . Analogously, we define $V^r = \{v_1^r, \dots, v_t^r\}$ and $E^r = \{e_1^r, \dots, e_t^r\}$ for the right-hand side, where v_t^r is the right most node. Accordingly, we adjust the notation for the edge lengths, upgrading costs, and edge length reduction bounds. A sketch of this notation is depicted in Figure 3.

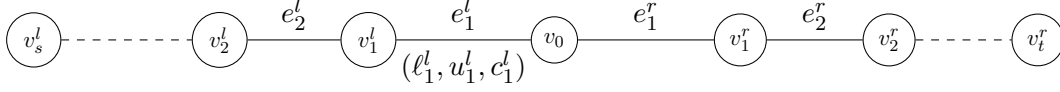


Figure 3: Notation used in Section 4

Without loss of generality, we consider only nodes in V^l and V^r that are reachable from v_0 . For example for the left-hand side, v_q^l , $1 \leq q \leq s$, is reachable if there exists a vector $\delta^l = (\delta_1^l, \dots, \delta_q^l)$, $0 \leq \delta_k^l \leq u_k^l$, $1 \leq k \leq q$, such that $d(v_q^l, v_0) \leq R + \sum_{k=1}^q \delta_k^l$ and $\sum_{k=1}^q c_k^l \delta_k^l \leq B$. We denote the minimal upgrading cost to make node v_q^l reachable by B_q^l which is computed by solving the following linear program whenever this value is smaller than or equal to B .

$$\begin{aligned}
 (\text{PL-}B_q^l) \quad & \min \sum_{k=1}^q c_k^l \delta_k^l \\
 \text{s.t.} \quad & \sum_{k=1}^q (\ell_k^l - \delta_k^l) \leq R, \\
 & 0 \leq \delta_k^l \leq u_k^l, \quad k = 1, \dots, q.
 \end{aligned}$$

Analogously, we define B_q^r for node v_q^r and these values can again be calculated by solving a linear program. Later, we will also provide a procedure to compute them efficiently by inspection.

With the facility located in v_0 , we now have to decide how much of the budget B we allocate to the sub-path to the left and how much to the sub-path to the right of v_0 . We observe that in order to be able to cover a node v_q^l on the left-hand side, we have to allocate at least the budget B_q^l to the left. Increasing the budget beyond B_q^l will obviously only have an effect, if we increase it up to B_{q+1}^l . Hence, the only useful budget allocations for the left are from the set $\{B_1^l, \dots, B_s^l\}$. We point out that the budget values are strictly increasing, the further we go to the left. Analogously we calculate these values for the right-hand side.

This observation allows us to develop an efficient procedure for calculating the maximal coverage we can achieve for a facility at v_0 . We start with the left most node v_s^l . To cover it, we have to allocate $B^l = B_s^l \leq B$ of the budget to the left-hand side. This will also cover all nodes v_1^l, \dots, v_{s-1}^l , so the total coverage is $\sum_{q \in V^l} w_q^l$. The remaining budget, $B^r = B - B^l$ can be invested in the sub-path on the right of v_0 and having computed

the values $\{B_1^r, \dots, B_t^r\}$, it is easy to determine the right most node v_q^r that can still be covered, i.e., for which $B_q^r \leq B^r$ and $B_{q+1}^r > B^r$, and the resulting weight: $\sum_{k=1}^q w_k^r$. This gives us the maximal coverage we can achieve under the assumption that v_s^l is covered. Now we can simply repeat this procedure from node v_{s-1}^l , to node v_0 , i.e., the case of $B^l = 0$. Note that this already considers all possible combinations of coverage and there is no need to repeat the same steps for the nodes to the right of v_0 . We summarize the steps in Algorithm 1, where we denote $B_0^l = 0$ ($B_0^r = 0$).

Algorithm 1: Optimal algorithm for the single facility Up-MCLP on paths.

Input: Network $N = (V, E, \ell)$; bounds u_e and upgrade costs c_e , $e \in E$; coverage radius $R > 0$; budget $B > 0$.

Output: Optimal solution (v^*, w^*, δ^*) .

```

1 Set  $w^* = 0$ .
2 foreach  $v_0 \in V$  do
3   Determine  $V^l, E^l, V^r, E^r, \{B_1^l, \dots, B_s^l\}$ , and  $\{B_1^r, \dots, B_t^r\}$ .
4   for  $q = s, \dots, 0$  do
5     Let  $B^l = B_q^l$  and  $B^r = B - B^l$ .
6     Determine the maximal  $1 \leq k \leq t$  such that  $B_k^r \leq B^r$ .
7     Calculate  $w = \sum_{i=1}^q w_i^l + \sum_{i=1}^k w_i^r$ .
8     if  $w > w^*$  then
9       | Set  $w^* = w$ ,  $\delta^* = \delta$ , and  $v^* = v_0$ .
10    end
11  end
12 end
13 return  $(v^*, w^*, \delta^*)$ .

```

Theorem 4 *The 1-Up-MCLP can be solved in $O(n^3)$ time on path networks.*

Proof. Let v_0 be fixed and consider Algorithm 1. In Step 3 we have to calculate the set of reachable nodes to the left and right of v_0 , together with the corresponding minimal upgrading cost. To do this efficiently, we start with the adjacent node v_1^l to the left of v_0 and calculate B_1^l . If $B_1^l \geq B$, we can already stop. If $B_1^l = B$ we set $s = 1$ and $V^l = \{v_1^l\}$, otherwise we set $s = 0$ and $V^l = \emptyset$. If instead $B_1^l < B$, then we calculate B_2^l for node v_2^l . We repeat this procedure until $B_q^l \geq B$ or we reach the left most node of the left sub-path. The sets and values to the right of v_0 are calculated analogously.

Although, B_q^l , $1 \leq q \leq s$, can be computed by solving (PL- B_q^l), we next describe a procedure with lower complexity. First, we sort the upgrade costs c_k^l , $1 \leq k \leq q$, in non-decreasing order:

$$c_{(1)}^l \leq c_{(2)}^l \leq \dots \leq c_{(q)}^l.$$

Let $\Delta = d(v_q^l, v_0) - R$. If $\Delta \leq 0$, then $B_q^l = 0$ and we are done. Thus, assume now that $\Delta > 0$. Pick the edge $e_{(1)}^l$ with the smallest upgrade cost $c_{(1)}^l$ and set $\delta_{(1)}^l = \min\{\Delta, u_{(1)}^l\}$, i.e., we upgrade the edge until we reach the upper bound or until we can cover v_q^l . Set $\Delta = \Delta - \delta_{(1)}^l$. If $\Delta = 0$, then $B_q^l = c_{(1)}^l \delta_{(1)}^l$ and we are done. Otherwise, we continue with the second-cheapest edge $e_{(2)}^l$. If we reach the last edge $e_{(q)}^l$ and still $\Delta > 0$, then v_q^l cannot be covered from v_0 , i.e., is unreachable. The values on the right-hand side can be determined analogously. Concerning the complexity of this step, for a given v_q^l , it takes $O(q \log q)$ time to sort the edges by non-decreasing upgrade costs from scratch. However, by starting with v_1^l and then working our way to the left, we can calculate the sorted list incrementally, by just inserting e_q^l in $O(\log q)$ time. Going through the sorted list and checking against Δ and updating it can then be done in time proportional to q , i.e., in $O(q)$ time. As we can have up to $n - 1$ nodes on the left-hand side, the total complexity of Step 3 is $O(\sum_{q=1}^{n-1} (\log q + q)) = O(n^2)$.

Step 5 requires constant time. The sum in Step 7 can be updated incrementally, resulting in a constant effort. Steps 8-10 are also constant. Finally, as B_q^r is strictly increasing and, hence, we have to look for the maximal k only to the right of the maximal k of the previous iteration, the complexity of Step 6 for all $q = s, \dots, 0$, is $O(n)$. In total, the effort for Steps 4-11 is $O(n)$.

Thus, the total complexity of the algorithm is dominated by Step 3, i.e., $O(n^2)$. Since it needs to be repeated n times, the complexity of the algorithm is $O(n^3)$. \square

At this point, we have shown that the 1-Up-MCLP on paths can be solved in polynomial time. This raises the question about the p -facility problem on those networks.

Theorem 5 *The p -Up-MCLP is NP-hard on path networks even assuming uniformity in upgrading costs and upper bounds as well as integrality in the input parameter values.*

Proof. We again prove the result by reducing KNAPSACK to the decision problem p -Up-MCLP-D. Let an instance of KNAPSACK be given with n items of positive weight $g_i \in \mathbb{N}$ and positive value $b_i \in \mathbb{N}$, $1 \leq i \leq n$, knapsack capacity $K \in \mathbb{N}$, and target value $U > 0$. We construct an instance of the upgrading problem in polynomial time as follows.

First, we observe that p -Up-MCLP-D is in NP since a given solution can be verified as such in polynomial time. Next, given an instance for KNAPSACK, we construct an instance of the edge upgrading problem in polynomial time as follows. Let $N = (V, E, \ell)$ be a path network with $2n$ nodes $V = \{v_1, v_2, \dots, v_{2n}\}$. Odd indexed nodes v_{2k-1} have weight $w_{2k-1} = W > \sum_{i=1}^n b_i$, and even indexed nodes v_{2k} have weight $w_{2k} = b_k$, $k = 1, \dots, n$. Let $V^{odd} = \{v_{2k-1} \mid k = 1, \dots, n\}$ and $V^{even} = V \setminus V^{odd}$. Moreover, let $u = \max_{i=1, \dots, n} g_i$, $R = 1 + u$, $c = 1$, $B = K$, and $T = pW + U$.

Concerning the edges, for $e_{2k-1} = (v_{2k-1}, v_{2k})$ we define $\ell_{2k-1} = R + g_k$, for $k =$

$1, \dots, n$. Moreover, for $e_{2k} = (v_{2k}, v_{2k+1})$ we define $\ell_{2k} = 2R$, for $k = 1, \dots, n - 1$. A sketch of the resulting network is depicted in Figure 4.

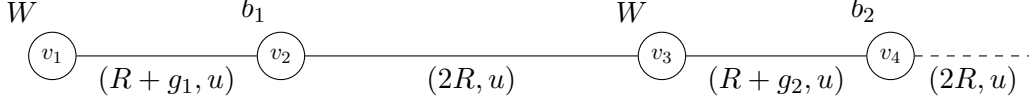


Figure 4: Notation used in Theorem 5, with edge labels (ℓ_j, u_j) .

Let $p = n$ and (X_p, δ) be a solution for p -Up-MCLP-D. As $\sum_{i \in C(X_p, \delta)} w_i \geq T > pW$, all odd-indexed nodes $v_{2k-1} \in V^{odd}$ must be covered. Therefore, we can assume without loss of generality that the $p = n$ facilities are located in the n odd-index nodes, i.e., $X_p = V^{odd}$. Let $M = C(X_p, \delta) \cap V^{even}$ be the set of all even-indexed nodes covered by X_p after the upgrade. An even-indexed node v_{2k} is covered if and only if $g_k \leq \delta_{2k-1} \leq u$. As

$$\sum_{i \in M} \delta_i \leq \sum_{i=1}^{2n} \delta_i \leq B = K$$

(recall that $c = 1$) we get $\sum_{i \in M} g_i \leq K$. Moreover, as

$$\sum_{i \in C(X_p, \delta)} w_i = pW + \sum_{i \in M} w_i \geq T = pW + U,$$

we obtain $\sum_{i \in M} w_i = \sum_{i \in M} b_i \geq U$ and M is a solution to KNAPSACK. Vice versa, for the same path network any solution M to KNAPSACK can easily be converted into a solution for p -Up-MCLP-D by setting $X_p = V^{odd}$ and $\delta_{2k-1} = g_k$, $k \in M$.

As a result, on path networks p -Up-MCLP-D is NP-complete and p -Up-MCLP is NP-hard. \square

The question whether the p -Up-MCLP with uniform weights is NP-hard or solvable in polynomial time remains open.

5 Tree networks

Theorem 3 shows that the 1-Up-MCLP is NP-hard on weighted star networks. Therefore, the 1-Up-MCLP is NP-hard on weighted tree networks, but analogously to the knapsack problem, it admits a pseudo-polynomial algorithm for the case where the input parameter values are integer.

Assumptions

A1: The parameters u_e and ℓ_e only take non-negative integer values.

A2: The parameters R and c_e only take positive integer values.

Let $S = (v_0, \delta)$ be an optimal solution of the 1-Up-MCLP. We call an edge $e \in E$ *fractional* with respect to S , if $\delta_e \notin \mathbb{N}_0 := \mathbb{N} \cup \{0\}$, otherwise it is called *integer* with respect to S . If the context is clear, then we usually omit the reference to S .

Theorem 6 *If Assumptions A1 and A2 hold, then there exists an optimal solution for the 1-Up-MCLP on trees where all edge upgrades are integer, i.e., $\delta_e \in \mathbb{N}_0$, $e \in E$.*

Proof. Let $N = (V, E, \ell)$ be a tree and $S = (v_0, \delta)$ be an optimal solution to the 1-Up-MCLP. If the solution has at least one fractional edge $\bar{e} \in E$, then we will show how to construct an equivalent optimal solution with at least one fractional edge less. In the following we interpret N as a tree rooted at v_0 . For a node $v_i \in V \setminus \{v_0\}$, $T(v_i) = T_i = (V_i, E_i)$ denotes the sub-tree rooted at v_i consisting of all descendants of v_i , including v_i itself, that are covered by v_0 with respect to δ . E_i consists of all edges which have both nodes in V_i .

Let $\bar{e} = (v_i, v_j) \in E$ be a fractional edge, i.e., $\delta_{\bar{e}} \notin \mathbb{N}_0$, such that $i \in T_j$ and there is no other fractional edge on the shortest path from v_j to v_0 . We have the following cases to consider:

Case 1: All edges in E_i are integer with respect to S , i.e., $\delta_e \in \mathbb{N}_0$, $\forall e \in E_i$.

As all edges on the shortest path from v_j to v_0 are integer, as well as all edge lengths and R , we can round down $\delta_{\bar{e}}$ to the nearest integer without losing coverage of any node in T_i , preserving the optimality of S (as $c_{\bar{e}} > 0$).

Case 2: There is at least one fractional edge in E_i .

Let $E'_i \subseteq E_i$ be the set of fractional edges in E_i such that there is no other fractional edge on the path between $e \in E'_i$ and \bar{e} . Moreover, let $V'_i \subseteq V_i$ be the set of those nodes of edges $e \in E'_i$ that are further away from v_0 than their counterpart on e , i.e., $V'_i = \{k \in V_i \mid e = (k, l) \in E'_i \text{ and } d(v_0, v_k) > d(v_0, v_l)\}$. Finally, set $C = \sum_{e \in E'_i} c_e$. Figure 5 depicts the setting with $E'_i = \{\hat{e}, \tilde{e}\}$ and $V'_i = \{v_5, v_8\}$.

We start with the case that $c_{\bar{e}} \leq C$. If we increase $\delta_{\bar{e}}$ by a small amount ϵ and decrease every δ_e , $e \in E'_i$, by the same amount, then all nodes remain covered: The distance to v_0 will decrease for every node in $V_i \setminus \bigcup_{k \in V'_i} V_k$ and will remain unchanged for each $v \in \bigcup_{k \in V'_i} V_k$. For example in Figure 5, the distance from v_0 to v_i , v_4 , and v_7 will decrease, while it will stay the same for v_5 , v_6 , and v_8 . Let $\epsilon = \min\{\lceil \delta_{\bar{e}} \rceil - \delta_{\bar{e}}, \min_{e \in E'_i} (\delta_e - \lfloor \delta_e \rfloor)\}$. As $c_{\bar{e}} \leq C$, increasing $\delta_{\bar{e}}$ by ϵ and reducing δ_e by ϵ for each $e \in E'_i$ will not exceed the budget and will preserve coverage with the same objective value, i.e., it will preserve optimality and result in one less fractional edge.

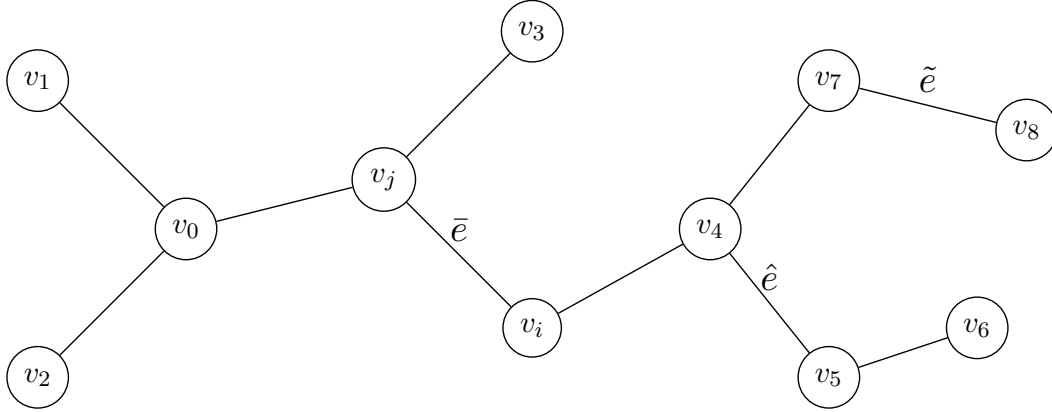


Figure 5: Illustration of the notation used in Case 2 of Theorem 6.

Next, we consider the case $C < c_{\bar{e}}$. If we decrease $\delta_{\bar{e}}$ by a small amount ϵ , then every node in $V_i \setminus \bigcup_{k \in V'_i} V_k$ will remain covered, even if the distance to v_0 increases. And if we increase δ_e for every $e \in E'_i$ by the same amount, then the distance to v_0 will be the same for each $v \in \bigcup_{k \in V'_i} V_k$. Using $\epsilon = \min\{\delta_{\bar{e}} - \lfloor \delta_{\bar{e}} \rfloor, \min_{e \in E'_i} (\lceil \delta_e \rceil - \delta_e)\}$, we can again preserve optimality while reducing the number of fractional edges by one.

This completes the proof. \square

Corollary 1 *If Assumptions A1 and A2 hold, then there exists an optimal solution for the 1-Up-MCLP on trees where the budget invested in upgrading an edge is always integer.*

Proof. The result follows immediately from the integrality of the δ -variables and the upgrading costs c_e . \square

The above two results can be extended to the p -facility case on general networks.

Theorem 7 *If Assumptions A1 and A2 hold, then there exists an optimal solution for the p -Up-MCLP on general networks, where all edge lengths are integers, i.e., $\delta_e \in \mathbb{N}_0$, $e \in E$, and the budget invested in upgrading an edge is also integer.*

Proof. Let $S = (X_p, \delta)$ be an optimal solution for p -Up-MCLP and let $e = (i, j)$ be fractional after the upgrade.

First, we assume that i and j can both be covered by the same facility $x \in X_p$, i.e., $i, j \in C(\{x\}, \delta)$. Without loss of generality, let $d(i, x, \delta) \leq d(j, x, \delta)$. We distinguish two different cases:

- If no shortest path from j to x passes i , then also no shortest path from x to any other node $k \in C(\{x\}, \delta)$ will cross e . Hence, we can reduce δ_e by a small amount ϵ

without affecting the distance from x to any $k \in C(\{x\}, \delta)$ covered by x . Therefore, we can replace δ_e by $\lfloor \delta_e \rfloor$ and we have one fractional edge less.

- If a shortest path from j to x passes i , then we compute the shortest path tree T_x rooted at x for all nodes in $C(\{x\}, \delta)$. T_x must necessarily include e . Using the same argumentation as in the proof of Theorem 6, we can reduce the number of fractional edges by one.

Next, we assume that i and j are covered by distinct facilities x and x' , respectively, $x, x' \in X_p$. Without loss of generality, we assume that x and x' are the closest facilities to i and j , respectively. As $i \in C(\{x\}, \delta)$ but $j \notin C(\{x\}, \delta)$, a shortest path from i to x cannot cross j , and vice versa a shortest path from j to x' cannot cross i . Thus, edge e is not included in the shortest path trees rooted at x or x' as described above, or, in fact, in the shortest path trees rooted at any facility in X_p . Thus, we can reduce δ_e by a small amount ϵ without affecting the distance from any node $k \in V$ to its closest facility in X_p . Replacing δ_e by $\lfloor \delta_e \rfloor$, we have one fractional edge less in our solution.

This completes the proof. \square

Based on the previous results, a pseudo-polynomial algorithm to solve the 1-Up-MCLP on trees is presented. Let $N = (V, E, \ell)$ be a tree rooted at node $v_0 \in V$. We follow the construction in Tamir (1996) to convert N into an equivalent binary tree, i.e., each node has exactly two descendants, as follows. For each non-leaf node v with only one descendant, we introduce a new node a and a new edge connecting v and a . a has weight 0, $\ell_{(v,a)} = u_{(v,a)} = 0$, and $c_{(v,a)} = 1$. For each non-leaf node v with $r \geq 3$ descendants, b_1, \dots, b_r , we introduce $r - 2$ new nodes a_2, \dots, a_{r-1} with weight 0. We remove the edges between v and b_2, \dots, b_r and replace them with $2r - 3$ new edges: (v, a_2) , (a_s, a_{s+1}) , $s = 2, \dots, r - 2$, (a_s, b_s) , $s = 2, \dots, r - 1$, and (a_{r-1}, b_r) . The first two groups of edges, i.e., (v, a_2) and (a_s, a_{s+1}) , have length 0, upgrading cost 1, and an upgrading bound of 0. The last two groups of edges, i.e., (a_s, b_s) and (a_{r-1}, b_r) , have the same parameters as the removed edges (v, b_s) , $s = 2, \dots, r$. The resulting tree is binary and has at most $2n - 3$ nodes. Moreover, it has the same optimal solution as N .

Let Assumptions A1 and A2 hold. In the following, we introduce a leaves-to-root dynamic programming algorithm to determine for a given location v_0 of the facility the optimal edge upgrades. For $n \leq m$, we use the notation $[n : m]$ to denote the set of integers between n and m , including n and m .

Let v_0 be the facility location and also the root of the tree. We denote by $L \subset V$ the set of leaves of the tree. For every $v_i \in V \setminus L$, we denote v_i^l and v_i^r the left and right child, respectively, of v_i . The corresponding edges are denoted as e_i^l and e_i^r , respectively. For simplicity, let $\ell_i^l = \ell_{e_i^l}$, $\delta_i^l = \delta_{e_i^l}$ and $c_i^l = c_{e_i^l}$; analogously for the right-hand child.

At the core of the algorithm is the function

$$f : V \times \mathbb{N}_0 \times \mathbb{N}_0 \rightarrow \mathbb{R}, (v, d, b) \mapsto f(v, d, b)$$

which gives the maximal amount of demand that can be covered in the subtree $T(v)$ rooted at v if the distance from v to v_0 is d and the budget for upgrading edges in $T(v)$ is b . Obviously, we must have $b \in [0 : B]$. Moreover, if $d > R$, then none of the nodes in $T(v)$ can be covered, including v , so the covered demand is zero.

Starting with the leaves, let $v_i \in L$, $b \in [0 : B]$, and $d \in [0 : d(v_i, v_0)]$. We have

$$f(v_i, d, b) = \begin{cases} w_i, & \text{if } d \leq R, \\ 0, & \text{otherwise.} \end{cases}$$

Let now $v_i \in V \setminus L$ be a non-leaf node. For $d \leq R$, the optimal coverage in T_i can be calculated as

$$f(v_i, d, b) = w_i + \max_{\substack{b_i^l + b_i^r + c_i^l \delta_i^l + c_i^r \delta_i^r = b \\ \delta_i^l \in [0 : u_i^l], \delta_i^r \in [0 : u_i^r] \\ b_i^l, b_i^r \in [0 : b]}} f(v_i^l, d + \ell_i^l - \delta_i^l, b_i^l) + f(v_i^r, d + \ell_i^r - \delta_i^r, b_i^r).$$

The maximum considers all possible combinations of splitting the budget b between the left and right subtree, b_i^l and b_i^r , and the two edges e_i^l and e_i^r , connecting v_i with its children. For $d > R$, $f(v_i, d, b)$ is zero.

The optimal coverage is then given by $f(v_0, 0, B)$. We summarise the steps in Algorithm 2 and provide its complexity in Theorem 8.

Theorem 8 *The 1-Up-MCLP on tree networks with integer parameters can be solved in $O(|V|^2 RB^3)$ time using Algorithm 2.*

Proof. Let v_0 be fixed. The core of the algorithm consists of computing the function f . Step 5 is computed in constant time for each $v_i \in L$, $b \in [0 : B]$, $d \in [0 : \min\{d(v_i, v_0), R\}]$.

Let $v_i \in V \setminus L$ be a non-leaf node, $d \in [0 : R]$, and $b \in [0 : B]$. There are $b + 1$ ways to split the budget between the left and right child: $(\bar{b}_i^l, \bar{b}_i^r) \in \{(b, 0), (b - 1, 1), \dots, (0, b)\}$. The budget \bar{b}_i^l for the left child can be spent on upgrading the edge e_i^l and/or upgrading edges in the subtree $T(v_i^l)$. The number of combinations depends on c_i^l and u_i^l . We have that $\delta_i^l \leq \min\{u_i^l, \lfloor \bar{b}_i^l / c_i^l \rfloor\}$, which is bounded by b . Considering all possible values for δ_i^l for each value of \bar{b}_i^l , we have in total at most

$$(b + 1) + b + (b - 1) \dots + 1 = O(b^2) = O(B^2)$$

Algorithm 2: Optimal algorithm for the single facility Up-MCLP on trees with integer parameters.

Input: Tree network $N = (V, E, \ell)$; bounds u_e and upgrade costs c_e , for $e \in E$; coverage radius $R > 0$; budget $B > 0$.

Output: Optimal solution (v^*, w^*, δ^*) .

```

1 Set  $w^* = 0$ .
2 foreach  $v_0 \in V$  do
3   Determine the binary tree rooted at  $v_0$ .
4   foreach  $v_i \in L$ ,  $b \in [0 : B]$ ,  $d \in [0 : \min\{d(v_i, v_0), R\}]$  do
5     Define  $f(v_i, d, b) = w_i$ .
6   end
7   foreach  $v_i \in V \setminus L$ ,  $b \in [0 : B]$ ,  $d \in [0 : \min\{d(v_i, v_0), R\}]$  do
8     Calculate
          
$$f(v_i, d, b) = w_i + \max_{\substack{b_i^l + b_i^r + c_i^l \delta_i^l + c_i^r \delta_i^r = b \\ \delta_i^l \in [0 : u_i^l], \delta_i^r \in [0 : u_i^r] \\ b_i^l, b_i^r \in [0 : b]}} f(v_i^l, d + \ell_i^l - \delta_i^l, b_i^l) + f(v_i^r, d + \ell_i^r - \delta_i^r, b_i^r).$$

9   end
10  if  $w = f(v_0, 0, B) > w^*$  then
11    Set  $w^* = w$ ,  $\delta^* = \delta$ , and  $v^* = v_0$ .
12  end
13 end
14 return  $(v^*, w^*, \delta^*)$ .

```

combinations to consider for the left child. Analogously for the right child. Note that whenever we call $f(\cdot)$, it takes constant time to check whether or not $d + \ell_i^l - \delta_i^l \leq R$ ($d + \ell_i^r - \delta_i^r \leq R$). Therefore, the complexity of computing Step 8 is $O(B^2)$. Step 3 takes $O(|V|)$ time (Tamir, 1996) and Step 11 takes constant time.

Therefore, going over all possible combinations of d and b for Step 8, the total effort to work through all combinations for a given node v_i is $O(RB^3)$. Doing this for every node, the overall complexity for finding the optimal upgrade for a given facility location v_0 is $O(|V|RB^3)$.

Finally, testing all nodes $v_0 \in V$ as facility locations, we can solve the 1-facility upgrading problem on trees with integer parameters in $O(|V|^2RB^3)$ time. Note that we do not have to construct the binary tree every time from scratch. Once we have obtained the equivalent binary tree for a node $v_0 \in V$, we can incrementally update it for a new root $v'_0 \in V$. For that we have to add a one new node a_0 and connect it to v_0 , and another new node a'_0 and connect it to v'_0 and two of its descendants, replacing the edges between v'_0 and those descendants (provided that none of the descendants is a node that was added in the initial transformation, in which case we can simply delete this node and its incident edge). The attributes of the new nodes and edges are set in the same way as above. This

can be done in constant time and the total number of nodes and edges of the binary tree increases by at most two and, respectively, one. \square

In the following, we provide a small example of solving an instance using Algorithm 2.

Example 1 Consider the network depicted in Figure 6. For each node, its name is printed inside the node and its weight is next to it. Let $R = 2$, $B = 3$, $\ell_j = 2$, and $c_j = u_j = 1$, for $j \in E$.

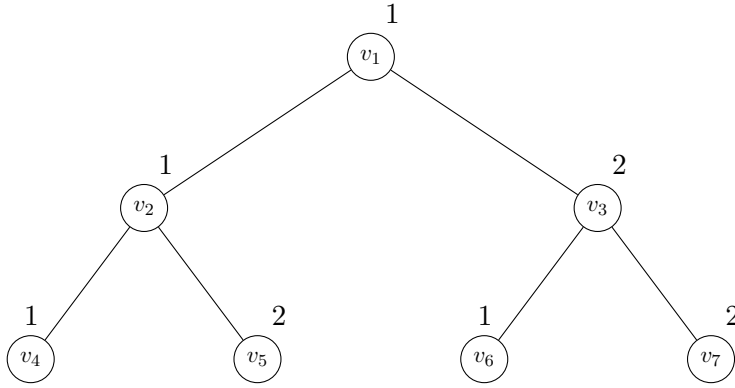


Figure 6: Binary tree with node weights in brackets.

We will describe the procedure for obtaining the optimal upgrading strategy assuming the facility's location is node v_1 (i.e., v_1 takes the role of v_0 using the notation of the algorithm). Observe that the algorithm requires that we repeat this process for each node of the tree in order to obtain the optimal location of the facility. Then, we will determine which facility location covers more demand after applying the optimal upgrading strategy.

We have $L = \{v_4, v_5, v_6, v_7\}$ as the set of leaves of the tree. For $v_i \in L$, $d \in [0 : 2]$, $b \in [0 : 3]$, we have $f(v_i, d, b) = w_i$. For other values of d and b this function has value zero.

In the following, Steps 7 and 8 of Algorithm 2 shall be carried out for v_2 and v_3 . To exemplify, we describe the process for v_2 (it is done analogously for v_3). We need to compute $f(v_2, d, b)$ for $b \in [0 : 3]$ and $d \in [0 : 2]$, i.e., 12 different combinations. Note that some combinations of parameters may not be possible to occur for a node, e.g., for v_2 we cannot have $d = 0$. To illustrate the procedure, we solve one of the 12 combinations, namely $d = 1$ and $b = 2$:

$$f(v_2, 1, 2) = 1 + \max_{\substack{b_2^l + b_2^r + \delta_2^l + \delta_2^r = 2 \\ \delta_2^l \in [0:1], \delta_2^r \in [0:1] \\ b_2^l, b_2^r \in [0:2]}} f(v_4, 3 - \delta_2^l, b_2^l) + f(v_5, 3 - \delta_2^r, b_2^r).$$

Therefore, the optimal solution of the previous problem is $\delta_2^l = \delta_2^r = 1, b_2^l = b_2^r = 0$, i.e.,

$$f(v_2, 1, 2) = w_2 + f(v_4, 2, 0) + f(v_5, 2, 0) = w_2 + w_4 + w_5 = 4.$$

After performing the same procedure for v_3 , we repeat the process for v_1 and obtain that an optimal upgrading strategy for this location of the facility is $\delta_1^l = 1, \delta_2^l = 1, \delta_2^r = 1$ and zero on all other edges (see Figure 7), then,

$$\begin{aligned} f(v_1, 0, 3) &= w_1 + f(v_2, 1, 2) + f(v_3, 2, 0) = w_1 + f(v_4, 2, 0) + f(v_5, 2, 0) + w_3 \\ &= w_1 + w_2 + w_3 + w_4 + w_5 = 7. \end{aligned}$$

Now, the described procedure has to be repeated assuming that the facility is located at

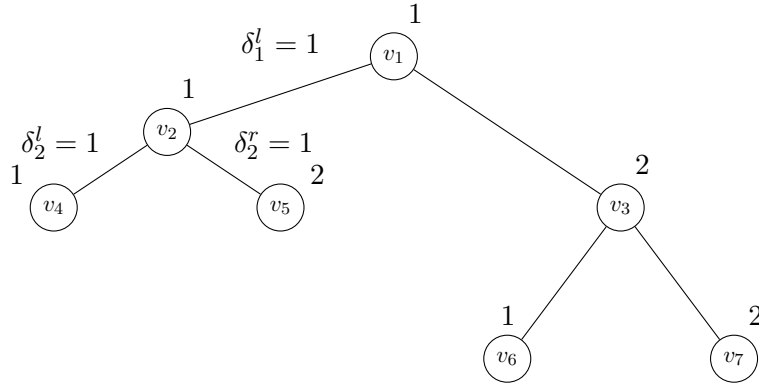


Figure 7: Optimal upgrading strategy for locating the facility at v_1

each of the other nodes in the tree (i.e., v_2, v_3, v_4, v_5, v_6 , and v_7). Note that to do so, an equivalent binary tree rooted at each of the nodes has to be built. Finally, it can be observed that the solution described above is an optimal solution to the problem.

In this section, we have developed a pseudo-polynomial algorithm for the 1-facility upgrading problem on trees with integer parameters. The remaining open question is whether the p -Up-MCLP with uniform weights is NP-hard or polynomial.

6 Concluding remarks

In this paper, we propose an algorithmic approach to deal with the upgrading version of the maximal covering location problem assuming that the edge length of the edges can be modified.

We prove that the problem on star networks is polynomial time solvable for uniform weights ($O(n \log n)$) and NP-hard for non-uniform weights. On path, we show that the

single facility problem is solvable in polynomial time ($O(n^3)$), while the p -facility problem is NP-hard even with uniform costs and upper bounds, as well as, integer parameter values. Furthermore, a pseudo-polynomial algorithm is developed for the single facility problem on trees with integer parameters ($O(|V|RB^3)$). A summary of the obtained result can be found in Table 1.

One of the remaining open question is whether the upgrading p -facility maximal covering problem with uniform weights is NP-hard or polynomial on path and tree networks.

Compliance with Ethical Standards

Funding

All authors were partially supported by research project PID2020-114594GB-C22 (Agencia Estatal de Investigación, Spain and the European Regional Development's funds (ERDF)) and research project RED2018-102363-T (Spanish Ministry of Science and Innovation).

Marta Baldomero-Naranjo was partially supported by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR through project TED2021-130875B-I00 and the BritishSpanish Society Scholarship Programme 2018 (Telefónica and the BritishSpanish Society).

Jörg Kalcsics was partially supported by Grant number EST2019-047 for research stays in Universidad de Cádiz (Plan Propio de Investigación y Transferencia de la UCA).

Antonio M. Rodríguez-Chía was partially supported by MCIN/AEI/10.13039/501100011033 and the European Union "NextGenerationEU"/PRTR through project TED2021-130875B-I00.

The authors would like to thank the anonymous reviewers for their comments and suggestions and declare that they have no conflict of interest.

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

- E. Afrashteh, B. Alizadeh, and F. Baroughi. Optimal algorithms for selective variants of the classical and inverse median location problems on trees. *Optimization Methods and Software*, 34(6):1213–1230, 2019.
- E. Afrashteh, B. Alizadeh, and F. Baroughi. Optimal approaches for upgrading selective

- obnoxious p-median location problems on tree networks. *Annals of Operations Research*, 289:153–172, 2020.
- E. Álvarez-Miranda and M. Sinnl. Lagrangian and branch-and-cut approaches for upgrading spanning tree problems. *Computers & Operations Research*, 83:13 – 27, 2017.
- L. Anton-Sanchez, M. Landete, and F. Saldanha-da Gama. The discrete p-center location problem with upgrading. *Omega*, 119:102894, 2023.
- M. Arana-Jiménez, V. Blanco, and E. Fernández. On the fuzzy maximal covering location problem. *European Journal of Operational Research*, 283(2):692 – 705, 2020.
- M. Baldomero-Naranjo, J. Kalcsics, and A. M. Rodríguez-Chía. Minmax regret maximal covering location problems with edge demands. *Computers & Operations Research*, 130:105181, 2021.
- M. Baldomero-Naranjo, J. Kalcsics, A. Marín, and A. M. Rodríguez-Chía. Upgrading edges in the maximal covering location problem. *European Journal of Operational Research*, 303(1):14–36, 2022.
- M. Bansal and K. Kianfar. Planar maximum coverage location problem with partial coverage and rectangular demand and service zones. *INFORMS Journal on Computing*, 29(1):152–169, 2017.
- O. Berman, J. Kalcsics, and D. Krass. On covering location problems on networks with edge demand. *Computers & Operations Research*, 74(C):214–227, 2016.
- B. B. Bhattacharya and S. C. Nandy. New variations of the maximum coverage facility location problem. *European Journal of Operational Research*, 224(3):477–485, 2013.
- V. Blanco and R. Gázquez. Continuous maximal covering location problems with interconnected facilities. *Computers & Operations Research*, 132:105310, 2021.
- V. Blanco and A. Marín. Upgrading nodes in tree-shaped hub location. *Computers & Operations Research*, 102:75–90, 2019.
- C. Büsing, A. Koster, S. Kirchner, and A. Thome. The budgeted minimum cost flow problem with unit upgrading cost. *Networks*, 69(1):67–82, 2017.
- A. M. Campbell, T. J. Lowe, and L. Zhang. Upgrading arcs to minimize the maximum travel time in a network. *Networks*, 47(2):72–80, 2006.
- R. Church and C. ReVelle. The maximal covering location problem. *Papers of the Regional Science Association*, 3:101–118, 1974.

- J.-F. Cordeau, F. Furini, and I. Ljubić. Benders decomposition for very large scale partial set covering and maximal covering location problems. *European Journal of Operational Research*, 275(3):882–896, 2019.
- M. Danilovic, D. Vasiljevic, and B. Cvetic. A novel pseudo-polynomial approach for shortest path problems. *Networks*, 78(2):107–127, 2021.
- I. Demgensky, H. Noltemeier, and H.-C. Wirth. On the flow cost lowering problem. *European Journal of Operational Research*, 137(2):265–271, 2002.
- I. Espejo and A. Marín. The p-median problem with upgrading of transportation costs and minimum travel time allocation. *Computers & Operations Research*, page 106354, 2023. doi: <https://doi.org/10.1016/j.cor.2023.106354>.
- N. Fröhlich, A. Maier, and H. W. Hamacher. Covering edges in networks. *Networks*, 75(3):278–290, 2020.
- S. García and A. Marín. Covering location problems. In G. Laporte, S. Nickel, and F. Saldanha da Gama, editors, *Location Science*, pages 99–119. Springer International Publishing, 2019.
- E. Gassner. Up-and downgrading the 1-median in a network. Technical Report 2007-6, Graz University of Technology, 2007.
- E. Gassner. Up-and downgrading the 1-center in a network. *European Journal of Operational Research*, 198(2):370 – 377, 2009.
- T. A. Hartmann, S. Lendl, and G. J. Woeginger. Continuous facility location on graphs. *Mathematical Programming*, 192:207–227, 2022.
- R. Hassin and A. Tamir. Improved complexity bounds for location problems on the real line. *Operations Research Letters*, 10(7):395–402, 1991.
- K. Hoppmann-Baum. On the complexity of computing maximum and minimum min-cost-flows. *Networks*, 79(2):236–248, 2022.
- J. Kalcsics, S. Nickel, J. Puerto, and A. M. Rodríguez-Chía. Several 2-facility location problems on networks with equity objectives. *Networks*, 65(1):1–9, 2015.
- M. Landete, I. Plana, J. L. Sainz-Pardo, and J. M. Sanchis. Upgrading edges in the Graphical TSP. *Computers & Operations Research*, 159:106321, 2023.
- I. Lari, J. Puerto, F. Ricca, and A. Scozzari. Uniform and most uniform partitions of trees. *Discrete Optimization*, 30:96–107, 2018.

- A. Marín, L. I. Martínez-Merino, A. M. Rodríguez-Chía, and F. S. da Gama. Multi-period stochastic covering location problems: Modeling framework and solution approach. *European Journal of Operational Research*, 268(2):432 – 449, 2018.
- P. A. Maya Duque, S. Coene, P. Goos, K. Sörensen, and F. Spieksma. The accessibility arc upgrading problem. *European Journal of Operational Research*, 224(3):458–465, 2013.
- S. Medya, P. Bogdanov, and A. Singh. Making a small world smaller: Path optimization in networks. *IEEE Transactions on Knowledge and Data Engineering*, 30(8):1533–1546, 2018.
- N. Megiddo, E. Zemel, and S. L. Hakimi. The maximum coverage location problem. *SIAM Journal on Algebraic and Discrete Methods*, 4(2):253–261, 1983.
- K. T. Nguyen and W. C. Teh. The uniform cost reverse 1-centdian location problem on tree networks with edge length reduction. *Vietnam Journal of Mathematics*, 51:345–361, 2023.
- A. Oudjit and M. F. Stallmann. Efficient algorithms for finding 2-medians of a tree. *Networks*, 77(3):383–402, 2021.
- D. Paik and S. Sahni. Network upgrading problems. *Networks*, 26(1):45–58, 1995.
- F. Plastria. Up- and downgrading the Euclidean 1-median problem and knapsack Voronoi diagrams. *Annals of Operations Research*, 246:227–251, 2016.
- J. Puerto, F. Ricca, and A. Scozzari. Extensive facility location problems on networks: an updated review. *TOP*, 26(2):187–226, 2018.
- A. R. Sepasian. Upgrading the 1-center problem with edge length variables on a tree. *Discrete Optimization*, 29:1–17, 2018.
- A. R. Sepasian and E. Monabbati. Upgrading min–max spanning tree problem under various cost functions. *Theoretical Computer Science*, 704:87–91, 2017.
- A. R. Sepasian and F. Rahbarnia. Upgrading p-median problem on a path. *Journal of Mathematical Modelling and Algorithms in Operations Research*, 14(2):145–157, 2015.
- Z. Şuvak, İ. K. Altinel, and N. Aras. Minimum cost flow problem with conflicts. *Networks*, 78(4):421–442, 2021.
- A. Tamir. An $O(pn^2)$ algorithm for the p -median and related problems on tree graphs. *Operations Research Letters*, 19:59–64, 1996.

-
- H. Wang and J. Zhang. An $O(n \log n)$ -time algorithm for the k -center problem in trees. *SIAM Journal on Computing*, 50(2):602–635, 2021.
- P. Yang, Y. Xiao, Y. Zhang, S. Zhou, J. Yang, and Y. Xu. The continuous maximal covering location problem in large-scale natural disaster rescue scenes. *Computers & Industrial Engineering*, 146:106608, 2020.
- Q. Zhang, X. Guan, and P. M. Pardalos. Maximum shortest path interdiction problem by upgrading edges on trees under weighted l_1 norm. *Journal of Global Optimization*, 79(4):959–987, 2021.