Omega 122 (2024) 102956

Contents lists available at ScienceDirect

Omega

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

Optimal coverage-based placement of static leak detection devices for pipeline water supply networks[☆]

Víctor Blanco [*], Miguel Martínez-Antón

Institute of Mathematics (IMAG), Universidad de Granada, Spain
Department Quant. Methods for Economics & Business, Universidad de Granada, Spain

ARTICLE INFO

Keywords:
Pipeline leak detection
Covering
Network
Location
Leak detection devices

ABSTRACT

In this paper, we provide a mathematical optimization-based framework to determine the location of leak detection devices along a network. Assuming that the devices are endowed with a known coverage area, we analyze two different models. The first model aims to minimize the number of devices to be located in order to (fully or partially) cover the volume of the network. In the second model, the number of devices is given, and the goal is to locate them to provide a coverage volume as broad as possible. Unlike other approaches in the literature, in our models, it is not assumed that the devices are located on the network (nodes or edges) but in the whole space and that the different segments in the networks may be partially covered, which allows for more flexible coverage. We also derive a method to construct initial solutions as well as a math-heuristic approach for solving the problem for larger instances. We report the results of a series of experiments on real-world water supply pipeline networks, supporting the validity of our models.

1. Introduction

The design of leak detection systems on water supply networks has attracted great interest due to the economic and environmental impact associated with the systematic loss of this resource. Needless to say, the important role water plays in our social and economic life system, such as in agriculture, manufacturing, the production of electricity, and sustaining human health.

In urban networks, where the supply pipeline network is buried, an average of 20% to 30% of the supply water is periodically lost [1]. This average exceeds 50% in places with less technological development where poor maintenance makes the system more vulnerable. It is also known [1] that ∼70% of the amount of wasted water is provoked by losses caused by leaks in modern networks. Pipe internal roughness or friction factors are the main causes of leakage of a water pipeline network [2,3], and as the pipelines get older, they become more susceptible to damage. In developed countries, it is expected that the annual disbursements for water leaks in their supply networks would be close to 10 billion USD, of which 2 billion would go to costs for damages due to water loss and 8 billion to social effects costs. Additionally, the International Institute of Water Management forecasts that 33% of the world's population will experience water scarcity by 2025 [4]. Thus, the efficient management of water supplies is and will be one of the main concerns of water authorities throughout the world.

Most of the efforts related to the management of water supply networks have focused on the detection of leaks once they occur. Rapid detection of the leak location is then crucial to minimize the impact of the leaks. Hamilton [5] suggests three different phases in the leak detection problem: *localization*, *location* and *pinpointing*. In the *localization* phase, the goal is to detect if a leak has occurred within a certain network segment after a suspected leak. There are several proposed Machine Learning based methodologies to estimate leak probabilities or to classify the event leak/no leak based on historic leakage datasets [6,7]. In the *location* phase, the uncertain area where the leak is localized is narrowed to ∼ 30 cm. Finally, in the *pinpointing* phase, the exact position of the leak is to be determined with a pre-specified accuracy of ∼ 20 cm by using hydrophones and/or geophones [8,9]. Previously to the determination of the position of the leak, a vast amount of literature have being devoted to modeling the determination of false/true leak alarms by the different available devices [10,11].

Another line of research on this topic is the design of control devices and methods for the accurate and quick detection of leakages. This is the case of the design of devices that accurately detect the leak within a restricted area [12]. Nevertheless, these devices are expensive and the adequate placement of the limited units must be strategically determined. One of the most popular approaches is by partitioning the network in *district-metered areas* where the flow and the pressure are monitored (leaks can be detected by a decrease of flow and pressure)

[☆] Area: Production Management, Scheduling and Logistics. This manuscript was processed by Associate Editor Yagiura.

[*] Corresponding author at: Institute of Mathematics (IMAG), Universidad de Granada, Spain.
E-mail addresses: vblanco@ugr.es (V. Blanco), mmanton@ugr.es (M. Martínez-Antón).

https://doi.org/10.1016/j.omega.2023.102956
Received 11 January 2023; Accepted 22 August 2023
Available online 24 August 2023
0305-0483/© 2023 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY-NC license (http://creativecommons.org/licenses/by-nc/4.0/).

by means of leak-detection devices at each of this areas [see e.g. 13]. However, one still has to decide the number of devices and their exact locations in each of the district-metered areas.

There are different types of devices designed to help in the different leak detection phases which are classified into static and dynamic devices. Static devices, such as sensors or data loggers, are usually located on the network, at utility holes, or directly on the ground, attached to the network. They keep a data transmission flow with a central server to detect and localize leaks. In contrast, dynamic devices are portable and used in the location and pinpointing phases on more specific areas where the leak was suspected to occur. Whereas static devices can be automated, dynamic ones must be controlled on-site by humans. Different technologies have been designed for the two different types of devices (see e.g. [14], for further details).

Most of the research on static leak detection systems is focused on the adequate estimation of the signals transmitted from the devices to the central server to detect an actual leak [15,16]. A few works analyze the optimal placement of a given number of static devices on a finite number of potential placements based on the capability of each of the potential places to detect a leak [17], or in the use of historic data to place the devices at the more convenient places [18].

This paper provides a technological decision support tool to help in the design of leak detection systems via the optimal placement of static devices. Instead of assuming that the devices are to be placed in a finite set of pre-specified potential places, they are allowed to be located in the whole space where the network lives, i.e. in the whole town, city, or district. We analyze, in this framework, two different strategies to place the devices. On the one hand, we derive a method to find the smallest number of devices (and their placements) required to detect *any* leak in the whole network or in a given percent of it. Since the devices may be costly, covering a large amount of the network might be expensive, and we also derive a method, that fixes the number of devices to be located based on a budget, and finds their optimal placements to cover as much volume of the network as possible.

The models that we propose belong to the family of Continuous Covering Location Problems. In this type of problem, the goal is to find the position of one or more services (in this case, the leak detection devices), each of them endowed with a *coverage area*, i.e., a limited region where the service/signal can be provided. Covering Location Problems are usually classified into (Partial) Set Covering Location Problems—(P)SCLP and Maximal Coverage Location Problems—MCLP. The goal of the (P)SCLP is to determine the minimum number of services (or equivalently the minimum set-up cost for them) to cover (part of) a given demand. In MCLP the number of services is known and the goal is to place them to cover as much demand as possible. These problems have been widely studied in the literature in case the given demand points to cover are finite and planar, and the coverage areas are Euclidean disks (see [19–22] for further information on this problems). Several extensions of these problems have been studied, by imposing connectivity between the services in higher dimensional spaces and different coverage areas [23], multiple types of services [24], under uncertainty [25], regional demand [26], or with ellipsoidal coverage areas [27].

We provide versions of the PSCLP and the MCLP, where instead of covering demand points, the goal is to cover lengths/volumes of the water supply pipeline networks, and the services to be located are the devices to detect leaks. The goal is either to find the number of devices and their optimal placement to fully or partially cover the whole length of the network (in the case of the PSCLP) or to find the placements of a given number of devices to maximize the length of the network which is covered by the devices. We assume that the coverage areas of the devices are ℓ_τ-norm based balls and that covering a part of the network with these shapes implies that the device is able to detect a leak there. As far as we know, this problem has never been investigated before despite its practical applications. Murray and Tong [20] analyze planar covering problems with generalized types of demand, as line segments or polygons, but where partial coverage is not allowed. In our approach, a part of the different elements taking part of the network is allowed to be covered, being the overall coverage maximized or lower bounded in our models.

We derive mathematical programming formulations for our model. First, we analyze the simple case when a single device is to be located. Next, we extend the model to the case of the simultaneous location of more than one device. We propose Mixed Integer Non-Linear Programming formulations for the problems, that are reformulated as Mixed Integer Second-Order Cone Optimization problems. We analyze some properties of the model that allow us to develop a strategy to construct initial solutions by solving an Integer Linear Optimization problem. We also design a math-heuristic approach to approximately the problem by solving, sequentially, the single-device versions of the problem. We have tested all our approaches in real-world urban water networks. In addition to analyzing the computational performance of our algorithms, we provide managerial insights about the locations obtained with our approaches, compared to the application of the classical algorithms in the literature, namely node and edge-restricted covering problems. Mathematical Optimization, the cornerstone of the developments in this paper, has been already recognized as a very powerful tool to analyze real-world pipeline networks (see e.g. [28,29]).

The rest of the paper is organized as follows. In Section 2 we introduce the problem under analysis and illustrate some of the solutions that can be obtained. Section 3 is devoted to analyzing the problem of locating a single device, which will be helpful in the development of approximation algorithms for the multi-device case. In Section 4 the general case is analyzed. We provide Mixed Integer Non-Linear Programming formulations for the maximal and partial set covering location problems and a deep study of them. We also provide a method to construct initial solutions for the problem based on the geometrical properties of the solutions and a math-heuristic approach based on solving, iteratively, single-device instances. The results of our computational experiments on real-world urban pipeline networks are reported in Section 5. Finally, in Section 6 we draw some conclusions and future research lines on the topic.

2. Length-coverage location of devices

In this section, we introduce the problem under study and fix the notation for the rest of the sections.

Let $G = (V, E; \Omega)$ be an undirected network with a set of nodes V, a set of edges E, and non-negative edge weights Ω. The graph represents an urban water pipeline network, where the weights are the diameter or roughness of each of the pipelines in the network, which together with its length will allow us to compute the covered volume of the network. We assume that the graph is embedded in $\mathbb{R}^d$, i.e., $V \subseteq \mathbb{R}^d$ and each (undirected) edge $e = \{o_e, f_e\} \in E$ is identified with a segment in $\mathbb{R}^d$, with endnodes o_e and f_e in V. Abusing notation, we identify edge $e \in E$ either with the segment induced by its end nodes, i.e., $e \equiv [o_e, f_e]$ or with the vector of $\mathbb{R}^d$ associated with them, i.e., $e \equiv f_e - o_e$.

A device located at $X \in \mathbb{R}^d$ is endowed with a ball-shaped coverage area in the form:

$$\mathbb{B}_R(X) = \{z \in \mathbb{R}^d : \|X - z\| \leq R\}$$

where $R > 0$ is the given coverage radius. We assume that $\|\cdot\|$ is an ℓ_τ-based norm with $\tau \geq 1$ or a polyhedral norm. Note that each of the devices can be endowed with a different radius and a different norm, based on their technical specifications.

For each edge $e \in E$, and a finite set of positions for the devices $\mathcal{X} \subset \mathbb{R}^d$, we denote by $\mathrm{CovWLength}_G(e, \mathcal{X})$ the weighted length of the edge covered by the devices. Let us denote by $\mathrm{TotWLength_G}$ the total weighted length of the network, i.e., $\mathrm{TotWLength_G} = \sum_{e \in E} \omega_e \|o_e - f_e\|$ with $\omega_e \in \Omega$.

We analyze in this paper two covering location problems to determine the position of the leak detection devices, namely the Partial

Set Network Length Covering Location Problem (PSNLCLP) and the Maximal Network Length Covering Location Problem (MNLCLP). In both cases, the goal is to place the different types of devices in the space to accurately detect a leak on the network.

Partial Set Network Length Covering Location Problem (PSNLCLP)
The goal of this problem is to determine the minimum number of devices and their positions in $\mathbb{R}^d$ in order to cover at least $100\gamma\%$ of the weighted length of the network, for a given $\gamma \in (0,1]$.

The PSNLCLP can be mathematically stated as:

$$\min_{\substack{\mathcal{X}\subseteq\mathbb{R}^d:\\ \sum_{e\in E}\text{CovWLength}_G(e,\mathcal{X})\geq\gamma\text{TotWLength}_{\text{G}}}} |\mathcal{X}|$$

The number of devices in the objective function can be replaced by the overall set-up costs for them, in whose case, the model read:

$$\min_{\substack{\mathcal{X}\subseteq\mathbb{R}^d:\\ \sum_{e\in E}\text{CovWLength}_G(e,\mathcal{X})\geq\gamma\text{TotWLength}_{\text{G}}}} \sum_{X\in\mathcal{X}} f_X.$$

being f_X a given set-up cost for the device $X \in \mathcal{X}$.

For the sake of simplicity, in this paper, we analyze the first model, although all the results are also valid for the second one.

Maximal Network Length Covering Location Problem (MNLCLP)
In this problem the number of devices to locate is given, $p \geq 1$, and the goal is to find their positions to maximize the weighted covered length of the network. While the MNLCLP consists of solving

$$\max_{\substack{\mathcal{X}\subseteq\mathbb{R}^d:\\ |\mathcal{X}|=p}} \sum_{e\in E} \text{CovWLength}_G(e,\mathcal{X})$$

Both in the PSNLCLP and the MNLCLP, one can provide different coverage for the different devices that want to be located. In the PSNLCLP, it is assumed that the number of available devices is unlimited (although minimized), but one can assume that different types of devices with different specifications are available. In the MNLCLP, since exactly p of them are to be located, different radii can be specified for each of them.

In the following example, we illustrate the two problems described above analyzed in a real network (see Section 5).

Example 1. Let us consider the network drawn in Fig. 1. There, each edge has a different weight indicating the diameter of the pipeline (as larger the weight, the thicker the line in the plot). Devices with identical Euclidean disk coverage areas of radius 0.5 are to be located (the network has been scaled to fit in a disk of radius 5). In Fig. 2 we show the solutions of the PSNLCLP for $\gamma = 0.75$ (right) and the solution of MNLCLP for $p = 5$ (left). There, the centers are highlighted as red stars, the covered segments of the network are colored in blue, and the coverage of the devices are the red disks.

Note that the flexible approach that we propose does not force the devices to be located at the nodes or edges of the network, being able to cover a larger amount of the volume of the network with a smaller number of devices.

Example 2. The two problems that we introduce here are defined in a very general framework (d-dimensional spaces, networks with no further assumptions, and general coverage shapes). In Fig. 3 we show solutions for the MNLCLP for the same instance that in Example 1, with $p = 5$ but in case the coverage areas are induced by ℓ_1-norm (left) and ℓ_∞-norm (right) balls.

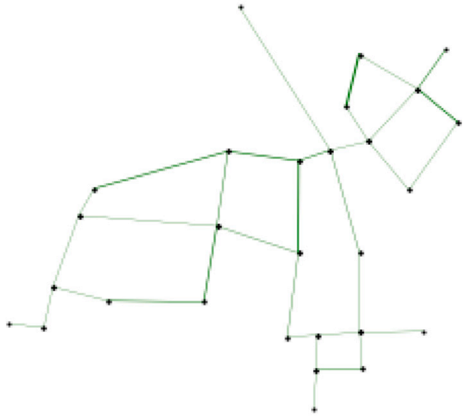


Fig. 1. Pipeline urban network of Example 1.

Remark 3. As already mentioned, most covering location problems on networks assume that the centers must be located either on the edges or the nodes of the network [30,31, see e.g.]. Here, this condition is no longer assumed, allowing the centers to be located at any place in the space where the network lives. This flexibility allows positions for the devices providing a larger coverage of the network. In Fig. 4 we show the solutions of the edge-restricted (left) and node-restricted (right) versions of the MNLCLP, where one can observe that the optimal positions of the devices are different from those obtained for the MNLCLP.

We have compared the covered lengths of the three problems (MNLCLP, edge-restricted MNLCLP, and node-restricted MNLCLP) for different values of p (2, 5, and 8), and different radii R (0.1, 0.25, and 0.5). In Fig. 5 we show a bar diagram with the average deviations (for each p) of the two restricted versions with respect to the covered length of the general approach that we propose. As can be observed, the solutions of the unrestricted MNLCLP are able to cover more than 6% than the edge-restricted problem and more than 20% than the node-restricted problem. Since undetected leaks may produce fatal consequences in an urban area and leak detection devices are expensive, the use of the solutions of our models is advisable in this situation.

3. The single-device maximal network length covering location problem

In this section, we first analyze the MNLCLP in case $p = 1$ (a single device). We provide a mathematical programming model for the problem that will be useful for the general construction of the multi-device instances of MNLCLP and PSNLCLP derived in this paper.

Let $e \in E$ be an edge in the network and $X \in \mathbb{R}^d$ a given location for a device. In case the coverage area of the device in X, $\mathbb{B}_R(X)$, does not *touch* the edge, then the covered length is clearly zero. Otherwise, since $\mathbb{B}_R(X)$ is a compact and convex body in $\mathbb{R}^d$, $\partial\mathbb{B}_R(X)$, the border of the ball, will touch the segment in two points (that may coincide in case the segment belong to a tangent hyperplane of the ball). These points belong to the segment $[o_e, f_e]$, that can be parameterized as:

$$Y_e^0 = \lambda_e^0 o_e + (1-\lambda_e^0) f_e \text{ and } Y_e^1 = \lambda_e^1 o_e + (1-\lambda_e^1) f_e$$

for some $\lambda_e^0, \lambda_e^1 \in [0,1]$. We can assume without loss of generality that Y_e^0 is closer to o_e than Y_e^1, so we restrict the λ-values to $\lambda_e^0 \leq \lambda_e^1$.

With the above parameterization, the length of the edge covered by X is $(\lambda_e^1 - \lambda_e^0)L_e$ (here, L_e denotes the length of the edge e).

To derive our mathematical programming formulation for the problem, we use the following sets of decision variables:

$$z_e = \begin{cases} 1 & \text{if edge } e \text{ intersects the device's coverage area,} \\ 0 & \text{otherwise.} \end{cases}$$

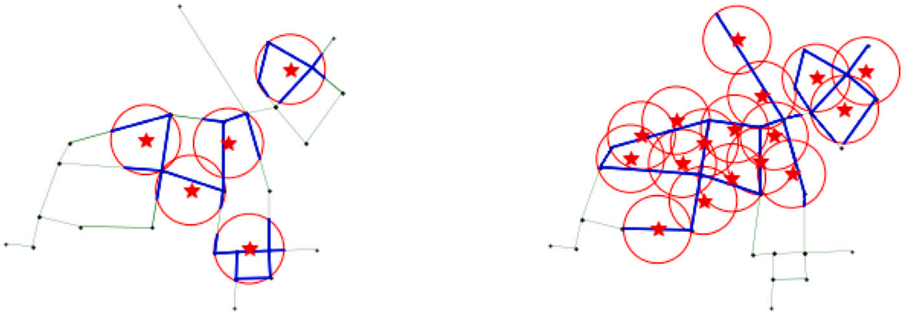


Fig. 2. Solutions of MNLCLP ($p = 5$) and PSNLCLP ($\gamma = 0.75$) of the network of Example 1.

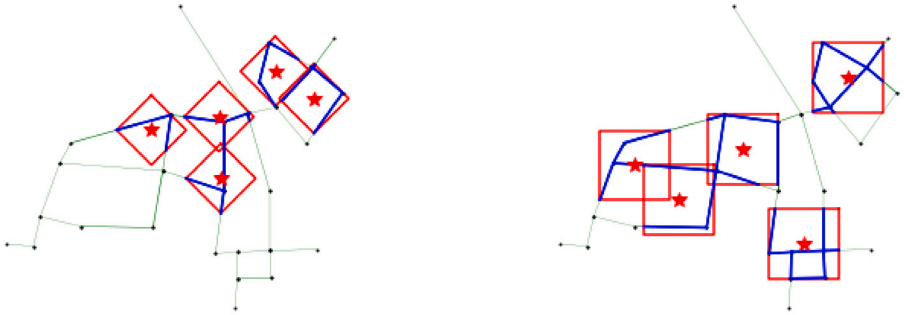


Fig. 3. Solutions of MNLCLP with $p = 5$ for coverage areas defined by ℓ_1-norm (left) and ℓ_∞-norm (right) balls.

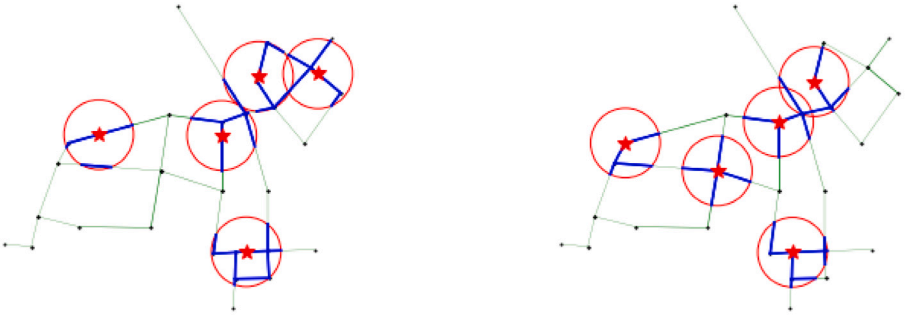


Fig. 4. Solutions of the edge-restricted and node-restricted versions of MNLCLP for $p = 5$ for the network of Example 1.

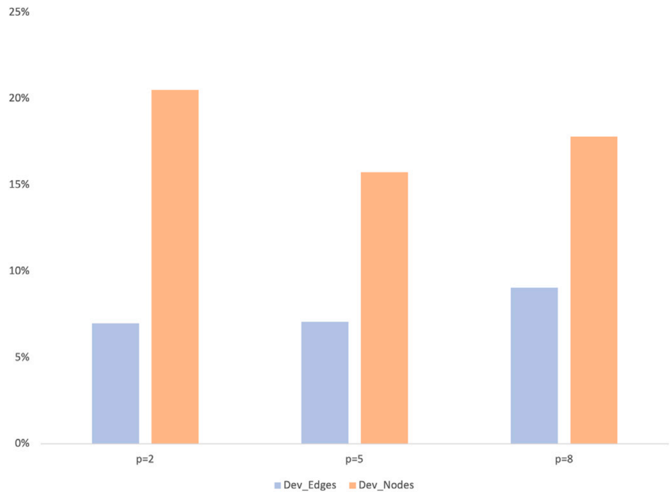


Fig. 5. Average length coverage deviations between the solutions of MNLCLP and the edges/nodes-restricted versions of the problem.

X : Coordinates of the placement of the device.

Y_e^0, Y_e^1 : Intersections points of $\partial\mathbb{B}_R(X)$ with the edge e.

λ_e^0, λ_e^1 : Parameterization values in the segment of intersection points Y_e^0 and Y_e^1, respectively.

The single-device MNLCLP can be formulated as the following Mathematical Programming Model, that we denote as (1-MNLCLP):

$$\max \sum_{e \in E} \omega_e L_e(\lambda_e^1 - \lambda_e^0) \tag{1}$$

$$\text{s.t. } \|X - Y_e^s\| z_e \leq R, \forall e \in E, s \in \{0,1\}, \tag{2}$$

$$Y_e^s = \lambda_e^s o_e + (1 - \lambda_e^s) f_e, \forall e \in E, s \in \{0,1\}, \tag{3}$$

$$\lambda_e^0 \leq \lambda_e^1, \forall e \in E, \tag{4}$$

$$\lambda_e^1 \leq z_e, \forall e \in E, s \in \{0,1\}, \tag{5}$$

$$\lambda_e^0, \lambda_e^1 \geq 0, \forall e \in E, s \in \{0,1\}, \tag{6}$$

$$z_e \in \{0,1\}, \forall e \in E, \tag{7}$$

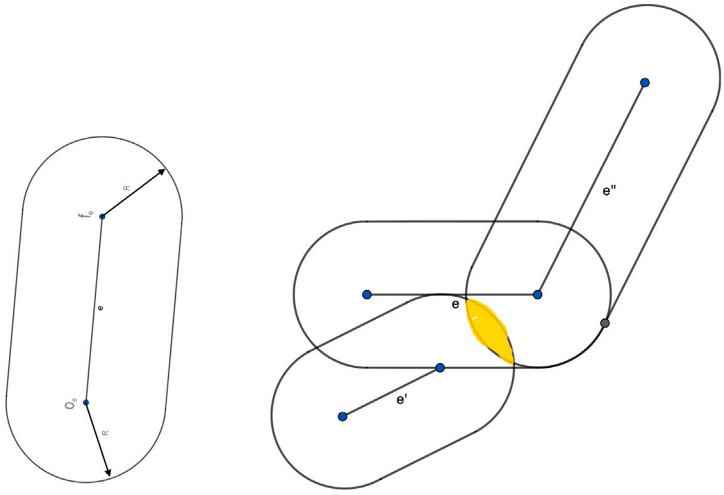


Fig. 6. Shape of extended edges (left) and the intersection of three of these *compatible* shapes (right).

$$X \in \mathbb{R}^d. \tag{8}$$

Constraints (2) enforce that in case the device coverage area intersects the edge, the intersection points must be in the coverage area of X. This constraint can be equivalently rewritten as:

$$\|X - Y_e^s\| \leq R + \Delta(1 - z_e), \forall e \in E, s \in \{0,1\}$$

where Δ a big enough constant with $\Delta > \max\left\{\|z_1 - z_2\| : z_1, z_2 \in \{o_e, f_e : e \in E\}\right\}$. Constraints (3) are the parameterizations of the intersection points. Constraints (4) force that Y_e^0 is closer to o_e than Y_e^1. In case the device does not intersect an edge, by Constraints (5) fix to zero the coefficients of the parameterization, adding a value of zero to the covered lengths in the objective function. (6)–(7) are the domains of the variables.

(1-MNLCLP) is a Mixed Integer Non-Linear Programming problem because of the discrete variables z and the nonlinear Constraints (2). For ℓ_τ or polyhedral norms, these constraints are known to be efficiently rewritten as a set of second-order cone constraints (and in the case of polyhedral norms, as linear constraints) becoming a Mixed Integer Second-Order Cone Optimization (MISOCO) problem that can be solved using the off-the-shelf software (see [32], for further details).

3.1. Generating feasible solutions of MNLCLP

The single-device version of the MNLCLP is already a challenging combinatorial problem since it requires computing a *feasible* group of edges which is able to be covered by the device (in addition to the computation of the covered volume). In what follows, we analyze some geometrical properties and algorithmic strategies for this problem, that will result in an Integer Linear Programming formulation to generate good quality initial feasible solutions to this problem. The same ideas will be extended to generate solutions also for the multi-device problem.

The following result, whose proof is straightforward from Constraints (2) provides a geometrical characterization of the potential position for the device to be located given that the *touched* set of edges is known.

Lemma 4. *Let $\bar{z} \in \{0,1\}^{|E|}$ be a feasible solution for 1-MNLCLP Denote by $C = \{e \in E : \bar{z}_e = 1\}$, the edges (total or partially) covered by the device. Then, we get that*

$$X \in \bigcap_{e \in C} (e \oplus \mathbb{B}_R(0)), \tag{Cov}$$

where $\oplus$ stands for the Minkowski sum in $\mathbb{R}^d$.

The above result states that the position of the device, X, must belong to the intersection of the extended segments induced by the edges in the cluster C. In Fig. 6 (left picture) we illustrate the shape of $e \oplus \mathbb{B}_R(0)$ for a given edge $e \in E$. In Fig. 6 (right picture) we show the intersection of three of these types of sets, where a device covering the three segments is allowed to be located.

As already mentioned, the main combinatorial decision of our models is to determine the sets of edges that are allowed to be *touched* by the same device, i.e., $S \subset E$, such that $\bigcap_{e \in S} (e \oplus \mathbb{B}_R(0)) \neq \emptyset$. Defining the set $\mathcal{C}$ as

$$\mathcal{C} = \left\{ S \subset E : \bigcap_{e \in S} (e \oplus \mathbb{B}_R(0)) \neq \emptyset \right\}$$

each element $S \in \mathcal{C}$ will be called a *compatible subset* for the device. In general, unless the radius is big enough, not all the subsets of E belong to $\mathcal{C}$.

In the following result, we describe a polynomial set (in $|E|$) of valid inequalities for our model that avoid those non-compatible sets in our models.

Lemma 5. *The following inequalities are valid for the 1-MNLCLP:*

$$\sum_{e \in S} z_e \leq d, \forall S \subset E \text{ with } |S| = d+1 \text{ and } \bigcap_{e \in S} (e \oplus \mathbb{B}_R(0)) = \emptyset. \tag{9}$$

Proof. It is straightforward to see, by Constraints (2), that the following condition is verified by any solution of (1)–(8):

$$\sum_{e \in S} z_e \leq |S| - 1, \forall S \subset E : \bigcap_{e \in S} (e \oplus \mathbb{B}_R(0)) = \emptyset,$$

Thus, by Helly's theorem [33], since the sets taking part of the intersections above, $(e \oplus \mathbb{B}_R(0))$, are compact and convex for any $e \in E$, the result follows. □

Corollary 6. *Let $\bar{z} \in \{0,1\}^{|E|}$ be a solution of the system of Diophantine in Eqs. (9). Then, $\bar{z}$ is a feasible solution for the 1-MNLCLP.*

In the classical Maximal Coverage Location Problems, the above observation allows us to replace the non-linear covering constraints – in the shape of (2) – with inequalities in the shape of (9), and the continuous variables which are involved in these constraints can be forgotten (see [23,24, e.g.]). In our model, the latter is no longer possible as in classic MCLP since the λ-values are also needed to compute the covered volume of the network.

Thus, we propose the following Integer Linear Programming formulation to obtain valid compatible subsets for the models.

$$\max \sum_{e\in E} \omega_e L_e z_e \tag{10}$$

$$\text{s.t.} \sum_{e\in S} z_e \leq d, \forall S \subset E (|S| = d+1) : \bigcap_{e\in S} (e \oplus \mathbb{B}_R(0)) = \emptyset, \tag{11}$$

$$z_e \in \{0,1\}, \forall e \in E. \tag{12}$$

The above mathematical programming model is an edge-based version of the classical 1-Maximal Coverage Location Problem, which is known to be NP-hard. The main advantage of this representation is that one can use techniques from Integer Linear Programming to strengthen or solve it, using the available off-the-shelf solvers.

The main bottleneck of this formulation is the computation of the intersections of $d+1$ sets in the form $e \oplus \mathbb{B}_R(0)$ which are empty, in whose case the corresponding inequality is added to the pool of constraints. The general methodology that can be applied for any dimension and any ℓ_τ-based norm, is by applying a relax-and-cut approach based on solving the problems above by removing Constraints (11), separating the violated constraints and incorporating them on-the-fly in an embedded branch-and-cut algorithm.

In what follows we focus on the planar Euclidean case, which is the most useful case in practice, and for which the formulations can be further simplified.

Observe that for $d = 2$, Constraints (9) are equivalent to:

$$z_e + z_{e'} \leq 1, \forall e, e' \in E : (e \oplus \mathbb{B}_R(0)) \cap (e' \oplus \mathbb{B}_R(0)) = \emptyset,$$

$$z_e + z_{e'} + z_{e''} \leq 2, \forall e, e', e'' \in E : (e \oplus \mathbb{B}_R(0)) \cap (e' \oplus \mathbb{B}_R(0)) \cap (e'' \oplus \mathbb{B}_R(0)) = \emptyset,$$

$$z_e \in \{0,1\}, \forall e \in E.$$

Thus, in order to incorporate these types of constraints one needs to check two and three-wise intersections of objects in the form $e \oplus \mathbb{B}_R(0)$. Although these shapes can be difficult to handle in general, the planar Euclidean case can be efficiently handled by analyzing the geometry of these objects as Minkowski sums of segments and disks.

The following results are instrumental for the development of the algorithm that we propose to generate the above sets of constraints. From now on, $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^2$.

Lemma 7. *Let e, e' be two segments in $\mathbb{R}^2$ and $\delta(e, e') = \min\{\|X - X'\| : X \in e, X' \in e'\}$. Then, if $\delta(e, e') > 0$, there exist $X \in e$ and $X' \in e'$ with $\delta(e, e') = \|X - X'\|$ such that either $X \in \{o_e, f_e\}$ or $X' \in \{o_{e'}, f_{e'}\}$.*

Proof. The result follows by observing that the minimum distance between two segments is always achieved by choosing one of the extremes of the segments. □

Lemma 8. *Let e be a segment in $\mathbb{R}^2$ and $Q \in \mathbb{R}^2$. Let S be the intersection point between the line induced by e, r, and its orthogonal line passing through the point Q. Then, $\delta(e, Q) := \min\{\|Q - X\| : X \in e\}$ can be computed as:*

$$\delta(e, Q) = \|Q - (\min\{\max\{0, \mu\}, 1\}(f_e - o_e) + o_e)\|,$$

where μ denote the parameterization value of S in r pointed at o_e.

Proof. The constructive proof is detailed in Appendix. □

Given a set of edges, E, and a radius R, using the above results, we develop algorithms to compute the two and three-wise intersections of sets in the form $e \oplus \mathbb{B}_R(0)$, for $e \in E$. The pseudocodes are shown in Algorithms 1 and 2. The set M, which is initialized to the empty set, will contain, the pairs (e, e') of $E \times E$ with $(e \oplus \mathbb{B}_R(0)) \cap (e' \oplus \mathbb{B}_R(0)) = \emptyset$ by checking the distance between the segments, $\delta(e, e')$. On the one hand, in case, $\delta(e, e') = 0$, both segments intersect so also their Minkowski sums. On the other hand, if $\delta(e, e') \neq 0$, we denote by r_e and $r_{e'}$ the lines containing the segments e and e', respectively, and by Q_0 their intersection point. By Lemma 7 there exist $X, X' \in \mathbb{R}^2$ with $\delta(e, e') = \|X - X'\|$, being $X \in \{o_e, f_e\}$ or $X' \in \{o_{e'}, f_{e'}\}$. Thus, four distances are enough to compute $\delta(e, e')$, namely $\delta_1 = \delta(o_{e'}, e)$, $\delta_2 = \delta(f_{e'}, e)$, $\delta_3 = \delta(o_e, e')$ and $\delta_4 = \delta(f_e, e')$, being $\delta(e, e') = \min\{\delta_1, \delta_2, \delta_3, \delta_4\}$. In case $\delta(e, e') > 2R$, then $(e \oplus \mathbb{B}_R(0)) \cap (e' \oplus \mathbb{B}_R(0)) = \emptyset$, and the tuple (e, e') is added to M.

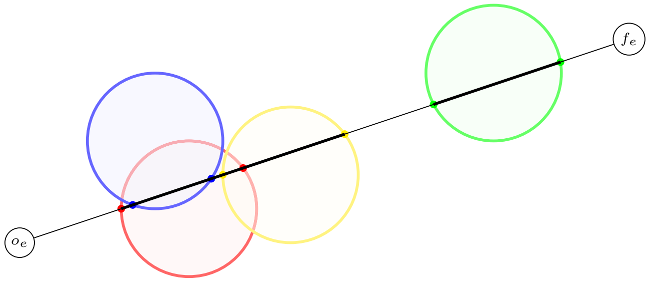


Fig. 7. Example of interaction between the coverages of different devices.

For the three-wise intersections, the set M is again initialized to the empty set. Then, for each triplet (e_1, e_2, e_3) whose pairwise intersection is non-empty (by Algorithm 1), we solve the following mathematical optimization problem:

$$\varepsilon^*(e_1, e_2, e_3) := \min \varepsilon \tag{13}$$

$$\text{s.t. } Y_i = (1 - \lambda_i) o_{e_i} + \lambda_i f_{e_i}, i = 1, 2, 3, \tag{14}$$

$$\|X - Y_i\| \leq R + \varepsilon, i = 1, 2, 3, \tag{15}$$

$$X \in \mathbb{R}^2, \tag{16}$$

$$\lambda_1, \lambda_2, \lambda_3 \in [0, 1], \tag{17}$$

$$\varepsilon \in \mathbb{R}. \tag{18}$$

The above problem is polynomial-time solvable since it can be rewritten as a continuous Second-Order Cone Optimization problem. Furthermore, its objective value provides a way to check for the emptiness of the three-wise intersection problem.

Lemma 9. *Let $e_1, e_2, e_3 \in E$ with $(e_i \oplus \mathbb{B}_R(0)) \cap (e_j \oplus \mathbb{B}_R(0)) \neq \emptyset$ for all $i, j \in \{1, 2, 3\}$. Then, $(e_1 \oplus \mathbb{B}_R(0)) \cap (e_2 \oplus \mathbb{B}_R(0)) \cap (e_3 \oplus \mathbb{B}_R(0)) \neq \emptyset$ if and only if $\varepsilon^*(e_1, e_2, e_3) = 0$.*

4. A general model for (PSNLCLP) and (MNLCLP)

In this section, we provide a general methodology to deal with the optimal location of devices in both the PSNLCLP and the MNLCLP. In the single-device problem analyzed in the previous section, the coverage of an edge can be directly computed by parameterizing the intersection of the boundary of the ball with the edge. Nevertheless, in the multi-device problem, the covered length does not coincide with the sum of the coverages of every single device separately, since the same part of a segment may be covered by two or more devices, but the covered length must be accounted for only once (otherwise the optimal placement for a set of devices is the collocation of all of them in the more weighted edge).

We illustrate the situation in the following toy example.

Example 10. Let us consider a planar network with a single edge e and four devices with Euclidean ball coverage areas as drawn in

Algorithm 1: A complete set of 2-wise incompatible edges.

```
Data: Set of edges, $E$, and radius $R$.
$M = \emptyset$
for $(e, e') \in E \times E$ do
    Set: $\bar{e} = f_e - o_e$.
    Set: $\bar{e'} = f_{e'} - o_{e'}$.
    Compute the intersection point of the lines $o_e + \langle \bar{e} \rangle$ and $o_{e'} + \langle \bar{e'} \rangle$: $Q_0$.
    Calculate $\mu_0$, $\mu'_0$ such that $Q_0 = \mu_0 \bar{e} + o_e$ and $Q_0 = \mu'_0 \bar{e'} + o_{e'}$.
    if $\mu_0$ or $\mu'_0 \notin [0, 1]$ then
        1. Compute the intersection point of the lines $o_e + \langle \bar{e} \rangle$ and $o_{e'} + \langle \bar{e}^{\perp} \rangle$: $Q_1$.
           Calculate $\mu_1$ such that $Q_1 = \mu_1 \bar{e} + o_e$.
           Set: $\delta_1 = \|o_{e'} - (\min\{\max\{0, \mu_1\}, 1\}\bar{e} + o_e)\|$.
        2. Compute the intersection point of the lines $o_e + \langle \bar{e} \rangle$ and $f_{e'} + \langle \bar{e}^{\perp} \rangle$: $Q_2$.
           Calculate $\mu_2$ such that $Q_2 = \mu_2 \bar{e} + o_e$.
           Set: $\delta_2 = \|f_{e'} - (\min\{\max\{0, \mu_2\}, 1\}\bar{e} + o_e)\|$.
        3. Compute the intersection point of the lines $o_e + \langle \bar{e'}^{\perp} \rangle$ and $o_{e'} + \langle \bar{e'} \rangle$: $Q_3$.
           Calculate $\mu_3$ such that $Q_3 = \mu_3 \bar{e'} + o_{e'}$.
           Set: $\delta_3 = \|o_e - (\min\{\max\{0, \mu_3\}, 1\}\bar{e'} + o_{e'})\|$.
        4. Compute the intersection point of the lines $f_e + \langle \bar{e'}^{\perp} \rangle$ and $o_{e'} + \langle \bar{e'} \rangle$: $Q_4$.
           Calculate $\mu_4$ such that $Q_4 = \mu_4 \bar{e'} + o_{e'}$.
           Set: $\delta_4 = \|f_e - (\min\{\max\{0, \mu_4\}, 1\}\bar{e'} + o_{e'})\|$.

        if $\min\{\delta_1, \delta_2, \delta_3, \delta_4\} > 2R$ then
            Add $(e, e')$ to $M$.
Result: $M = \{(e, e') \in E \times E : (e \oplus \mathbb{B}_R(0)) \cap (e' \oplus \mathbb{B}_R(0)) = \emptyset\}$.
```

Algorithm 2: A complete set of 3-wise incompatible edges (which are pair-wise compatible).

```
Data: Set of edges, $E$, and radius $R$.
$L = \{(e_1, e_2, e_3) \in E \times E \times E : (e_1 \oplus \mathbb{B}_R(0)) \cap (e_2 \oplus \mathbb{B}_R(0)) \neq \emptyset, (e \oplus \mathbb{B}_R(0)) \cap (e_3 \oplus \mathbb{B}_R(0)) \neq \emptyset, (e_2 \oplus \mathbb{B}_R(0)) \cap (e_3 \oplus \mathbb{B}_R(0)) \neq \emptyset\}$.
$M_3 = \emptyset$.
for $(e_1, e_2, e_3) \in L$ do
    Compute $\varepsilon^*(e_1, e_2, e_3)$.
    if $\varepsilon^*(e_1, e_2, e_3) > 0$ then
        Add $(e_1, e_2, e_3)$ to $M_3$.
Result: $M = \{(e_1, e_2, e_3) \in E \times E \times E : (e_1 \oplus \mathbb{B}_R(0)) \cap (e_2 \oplus \mathbb{B}_R(0)) \cap (e_3 \oplus \mathbb{B}_R(0)) = \emptyset\}$.
```

Fig. 7. The four devices *touch* the edge. The covered length of the edge is highlighted with thicker segments in the picture. Clearly, this length cannot be computed by adding up separately each of the covered lengths of the devices.

In what follows we derive a mathematical programming model that overcomes this situation.

Observe that the positions of the intersection points of the coverage areas of p devices with an edge (segment) e provide a partition of the edge in at most $2p+1$ subsegments of e. Each of those subsegments is either fully covered or non-covered by the device. Let $\lambda^0_{1e}, \lambda^1_{1e}, \ldots, \lambda^0_{pe}, \lambda^1_{pe}$ the parameterizations of the intersection points of the p devices with respect to e (here λ^0_{je} and λ^1_{je} stands for the parameterizations of the intersection of the coverage area of jth device with segment induced by the edge e).

We assume that the devices not intersecting the edge have both lambda values equal to zero. Sorting the λ^0 and λ^1 values one gets two sorted sequences in the form:

$$\Lambda^0_e := \lambda^0_{(1)e} \leq \cdots \leq \lambda^0_{(p)e}$$

$$\Lambda^1_e := \lambda^1_{(1)e} \leq \cdots \leq \lambda^1_{(p)e}$$

Merging both lists one gets all the partitions of the segment e by the different intersection points:

$$\Lambda_e := \lambda^{i_1}_{(1)e} \leq \cdots \leq \lambda^{i_{2p}}_{(2p)e}$$

where $i_1, \ldots, i_{2p} \in \{0, 1\}$.

For each $l \in \{1, \ldots, 2p-1\}$, the intervals $[\lambda^{i_l}_{(l)e}, \lambda^{i_{l+1}}_{(l+1)e}]$ along with $[o_e, \lambda^{i_1}_{(1)e}]$ and $[\lambda^{i_{2p}}_{(2p)e}, f_e]$ induce a subdivision of the segment e into $2p+1$ pieces (some of them probably singletons). Furthermore, in case any of the extreme points, o_e and f_e, are covered by a device, the corresponding subsegment will be a singleton ($[o_e, o_e] = \{o_e\}$ or $[f_e, f_e] = \{f_e\}$), whose length is zero. Otherwise, in case any extreme subsegment is not covered, still its contribution to the objective function is zero. Thus, in our formulation, it is enough considering the $2p-1$ intermediate subsegments in the subdivision.

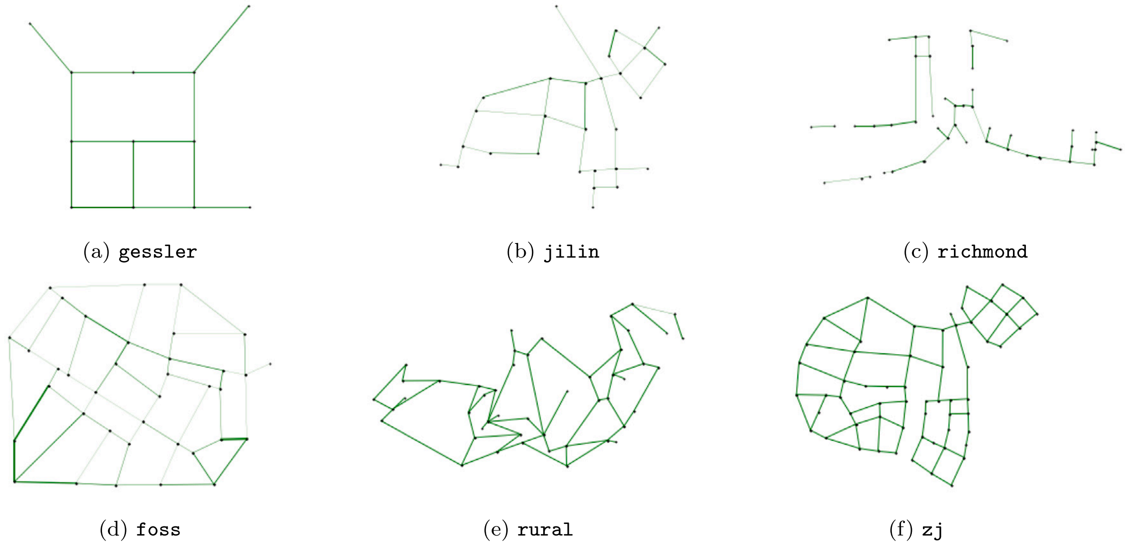
Given the sequence Λ_e for the p given devices located at $X_1, \ldots, X_p$, one can easily determine which of the subsegments in the partitions are covered by the facilities as stated by the following straightforward observation.

Lemma 11. *A subsegment in the form $s = [\lambda^{i_l}_{(l)e}, \lambda^{i_{l+1}}_{(l+1)e}]$ is covered by a set of devices if and only if $s \subseteq [\lambda^0_{je}, \lambda^1_{je}]$ for some $j = 1, \ldots, p$ with $\lambda^0_{je} < \lambda^1_{je}$.*

With the above observations, we derive mathematical programming formulations for the multi-device versions of PSNLCLP and MNLCLP.

We denote by $P = \{1, \ldots, p\}$ the index set for the devices to locate and by $Q = \{1, \ldots, 2p-1\}$ the index sets for the intermediate subsegments in the partition induced by the Λ sequences. We assume that the jth device is endowed with a $\|\cdot\|$-based coverage area with radius R_j.

We use the following decision variables in our models:



(a) gessler (b) jilin (c) richmond

(d) foss (e) rural (f) zj

Fig. 8. Networks used in our computational experiments.

$$z_{je} = \begin{cases} 1 & \text{if edge } e \text{ intersect the } j\text{th device's coverage area,} \\ 0 & \text{otherwise} \end{cases} \qquad \forall j \in P, \forall e \in E.$$

$X_{j1}, \ldots, X_{jd}$: Coordinates of the jth device, $\forall j \in P$.

$\lambda^0_{je}, \lambda^1_{je}$: Parameterization in the segment of the two intersection points of $\partial \mathbb{B}_{R_j}(X_j)$ with segment e, $\forall j \in P, \forall e \in E$.

$$w_{e\ell} = \begin{cases} 1 & \text{if the } \ell\text{-th subsegment of edge } e \text{ is} \\ & \text{covered by some device,} \\ 0 & \text{otherwise,} \end{cases} \qquad \forall \ell \in Q, \forall e \in E.$$

$$\xi^s_{je\ell} = \begin{cases} 1 & \text{if } \lambda^s_{je} \text{ is sorted in } \ell\text{-th} \\ & \text{position in the list of } \Lambda_e, \\ 0 & \text{otherwise} \end{cases} \qquad \forall j \in P, \forall \ell \in Q \cup \{2p\}, \forall e \in E.$$

With the above set of variables, the amount:

$$L_e \left[\sum_{j \in P} \sum_{s=0}^{1} \lambda^s_{je} \xi^s_{je(\ell+1)} - \sum_{j \in P} \sum_{s=0}^{1} \lambda^s_{je} \xi^s_{je\ell} \right]$$

determines the length of the ℓ-th subsegment in case it is covered by any of the devices in P. Note that in case such a subsegment is $[\lambda^s_{je}, \lambda^{s'}_{j'e}]$, the above expression becomes $L_e(\lambda^{s'}_{j'e} - \lambda^s_{je})$ which is the desired amount.

Thus, the overall volume coverage of the network can be computed as:

$$\sum_{e \in E} \sum_{\ell \in Q} \omega_e w_{e\ell} L_e \left[\sum_{j \in P} \sum_{s=0}^{1} \lambda^s_{je} \xi^s_{je(\ell+1)} - \sum_{j \in P} \sum_{s=0}^{1} \lambda^s_{je} \xi^s_{je\ell} \right]$$

In order to adequately represent the decision variables in our model, the following constraints are considered:

1. *Coverage Constraints:*

$$\|(\lambda^s_{je} e + o_e) - X_j\| z_{je} \leq R_j, \forall j \in P, \ \forall e \in E, \ s = 0, 1 \tag{19}$$

These constraints enforce that in case an edge is accounted as *touched* by the jth device ($z_{je} = 1$), the two intersection points $(\lambda^0_{je} e + o_e)$ and $(\lambda^1_{je} e + o_e)$ belong to $\mathbb{B}_{R_j}(X_j) \cap e$. This constraint can be reformulated as:

$$\|(\lambda^s_{je} e + o_e) - X_j\| z_{je} \leq R_j + \Delta(1 - z_{je}), \forall j \in P, \ \forall e \in E, \ s = 0, 1$$

where Δ a big enough constant with $\Delta > \max\left\{ \|z_1 - z_2\| : z_1, z_2 \in \{o_e, f_e : e \in E\} \right\}$.

2. *Directed Parameterization:*

$$\lambda^0_{je} \leq \lambda^1_{je}, \forall j \in P, \ \forall e \in E. \tag{20}$$

In case the coverage area of a device j touches the segment e, the segment is oriented in the parameterization.

3. *Zero parameterizations for untouched edges*

$$\lambda^1_{je} \leq z_{je}, \forall j \in P, \ \forall e \in E. \tag{21}$$

In case the jth device does not touch the segment induced by an edge e, the covered length of such an edge by the device will be zero. By (19), in that case, the device is not restricted to touching the segment, but to assure that no length is accounted for, we fix both λ-values in the fictitious intersection to zero.

4. *Λ-Sorting Constraints:*

$$\sum_{j \in P} (\xi^0_{je\ell} + \xi^1_{je\ell}) = 1, \forall e \in E, \ \forall \ell \in Q \cup \{2p\}, \tag{22}$$

$$\sum_{\ell \in Q \cup \{2p\}} \xi^s_{je\ell} = 1, \forall j \in P, \ \forall e \in E, \ s = 0, 1 \tag{23}$$

$$\sum_{j \in P} (\lambda^0_{je} \xi^0_{je\ell} + \lambda^1_{je} \xi^1_{je\ell}) \leq \sum_{j \in P} (\lambda^0_{je} \xi^0_{je(\ell+1)} + \lambda^1_{je} \xi^1_{je(\ell+1)}), \forall e \in E, \ \forall \ell \in Q. \tag{24}$$

These constraints allow us to adequately define the variables ξ. Constraints (22) and (23) assure that for each e each λ_e-value is sorted in exactly a single position in Q and that each position is assigned to exactly one λ_e value. Constraint (24) enforces that the ξ-variables sort the λ-values in non-decreasing order.

5. *Coverage of subsegments:*

$$w_{e\ell} \leq \sum_{j \in P} \left(\sum_{i \leq \ell} \xi^0_{jei} + \sum_{i > \ell} \xi^1_{jei} - 1 \right), \forall e \in E, \ \forall \ell \in Q, \tag{25}$$

The coverage of a subsegment $\ell \in Q$ is assured by the existence of a device j for which its λ^0_{je} is sorted in a position anterior to ℓ ($\sum_{i \leq \ell} \xi^0_{jei} = 1$) and λ^1_{je} in a posterior position to ℓ ($\sum_{i > \ell} \xi^1_{jei} = 1$). Thus, in case both values are 1, the conditions of Lemma 11 are verified, and the subsegment is covered. Otherwise, one of the above sums is zero, and the constraint is redundant. Indeed, if $\sum_{i \leq \ell} \xi^0_{jei} = 0$, then, by (23), $\sum_{i > \ell} \xi^0_{jei} = 1$. Thus, by (20) and (24), one has that $\sum_{i > \ell} \xi^1_{jei} = 1$. Similarly, if $\sum_{i > \ell} \xi^1_{jei} = 0$, one has that $\sum_{i \leq \ell} \xi^0_{jei} = 1$. In both cases, $\sum_{i \leq \ell} \xi^0_{jei} + \sum_{i > \ell} \xi^1_{jei} - 1$

takes value zero, implying that the jth device does not cover the subsegment.

Apart from the constraints above, we incorporate into our model the following valid inequalities that allow us to strengthen it:

1. *Touched segments and covered subsegments:*

$$\sum_{\ell \in Q} w_{e\ell} \leq 2 \sum_{j \in P} z_{je}, \quad \forall e \in E.$$

In case the whole segment is not touched by any device, non of the subsegments are covered.

2. *Symmetry breaking:*

$$\sum_{k=1}^{d} X_{jk} \leq \sum_{k=1}^{d} X_{(j+1)k}, \quad \forall j \in P, j < p.$$

Since the devices to be located are indistinguishable, any permutation of the j-index will result in an alternative optimal solution, hindering the solution procedure based on a branch-and-bound tree. The above inequality prevents such an amount of alternative optima.

3. *Incompatible edges:*

$$z_{ej} + z_{e'j} \leq 1, \quad \forall j \in P, \forall e, e' \in E \text{ with } \min\{\|x - x'\| : x \in e, x' \in e'\} > 2R_j.$$

Edges that are far enough are not able to be simultaneously touched by the same device.

***Mathematical Programming Model for (MNLCLP)*:**

Using the variables and constraints previously described, the following mathematical programming formulation is valid for the MNLCLP:

$$\max \sum_{e \in E} \sum_{\ell \in Q} \omega_e w_{e\ell} L_e \left[\sum_{j \in P} \sum_{s=0}^{1} \lambda_{je}^s \xi_{je(\ell+1)}^s - \sum_{j \in P} \sum_{s=0}^{1} \lambda_{je}^s \xi_{je\ell}^s \right]$$

s.t. (19)–(25),

$$\begin{aligned}
&\lambda_{je}^s \in [0,1], && \forall j \in P, \forall e \in E, s = 0,1,\\
&X_j \in \mathbb{R}^d, && \forall j \in P,\\
&z_{je} \in \{0,1\}, && \forall j \in P, \forall e \in E,\\
&\xi_{je\ell}^s \in \{0,1\}, && \forall j \in P, \forall e \in E, \forall \ell \in Q \cup \{2p\}, s = 0,1,\\
&w_{e\ell} \in \{0,1\}, && \forall e \in E, \forall \ell \in Q.
\end{aligned}$$

***Mathematical Programming Model for (PSNLCLP)*:**

(PSNLCLP) seeks to minimize the number of devices to cover at least a portion $\gamma \in (0,1]$ of the length of the network. Although the above variables and constraints can be used to derive similarly a model for this problem, the number of devices, p, to locate is unknown in this case. We estimate an upper bound for this parameter and consider the following binary variables to *activate/desactivate* them.

$$y_j = \begin{cases} 1 & \text{if device } j \text{ is activated,} \\ 0 & \text{otherwise.} \end{cases} \quad \forall j \in P.$$

Then, the (PSNLCLP) can be formulated as follows:

$$\min \sum_{j \in P} y_j$$

s.t. (19)–(25), (26)

$$\sum_{e \in E} \sum_{\ell \in Q} \omega_e w_{e\ell} L_e \left[\sum_{j \in P} \sum_{s=0}^{1} \lambda_{je}^s \xi_{je(\ell+1)}^s - \sum_{j \in P} \sum_{s=0}^{1} \lambda_{je}^s \xi_{je\ell}^s \right] \geq \gamma \sum_{e \in E} \omega_e L_e, \tag{27}$$

$$z_{je} \leq y_j, \quad \forall j \in P, \quad \forall e \in E; \tag{28}$$

$$\begin{aligned}
&\lambda_{je}^s \in [0,1], \forall j \in P, \forall e \in E, s = 0,1,\\
&X_j \in \mathbb{R}^d, \forall j \in P,\\
&z_{je} \in \{0,1\}, \forall j \in P, \forall e \in E,\\
&\xi_{je\ell}^s \in \{0,1\}, \forall j \in P, \forall e \in E, \forall \ell \in Q \cup \{2p\}, s = 0,1,\\
&w_{e\ell} \in \{0,1\}, \forall e \in E, \forall \ell \in Q,\\
&y_j \in \{0,1\}, \forall j \in P.
\end{aligned}$$

In this case, the objective function accounts for the number of activated devices, Constraint (27) assures that at least a portion of γ of the coverage volume is attained, and (28) prevents covering edges by devices that are not activated.

Instead of minimizing the number of devices one may also minimize the set-up costs by incorporating individual set-up costs for each of the available devices (f_j for $j \in P$) and replace the above objective function by $\sum_{j \in P} f_j y_j$.

To avoid multiple optimal solutions due to symmetry, we also incorporate into the model the following constraints that avoid activating the jth device in case the $(j-1)$-th device is not activated in the solution.

$$y_{j-1} \geq y_j, \forall j \in P, j > 1.$$

Remark 12. The complexity of the PSNLCLP highly depends on the number of potential devices to locate (p), since the number of constraints and variables are affected by this parameter. We derive a method to compute a reasonable upper bound for that parameter which is based on computing the minimum number of devices necessary to cover each edge in $U_\gamma \subseteq E$ where U_γ is defined as a minimal set verifying that $\sum_{e \in U_\gamma} \omega_e L_e \geq \gamma \sum_{e \in E} \omega_e L_e$.

We initialize $U_\gamma = \emptyset$, and sort the sequence $\{\omega_e L_e\}_{e \in E}$ such that $\omega_{e_1} L_{e_1} \geq \omega_{e_2} L_{e_2} \geq \cdots \geq \omega_{e_i} L_{e_i} \geq \omega_{e_{i+1}} L_{e_{i+1}} \geq \cdots$. Then, we define $U_\gamma = \{e_1, \ldots, e_k\}$ such that:

$$\sum_{i=1}^{k} \omega_{e_i} L_{e_i} \geq \gamma \sum_{e \in E} \omega_e L_e > \sum_{i=1}^{k-1} \omega_{e_i} L_{e_i}.$$

Since the minimum number of devices necessary to cover a single edge e is $\left\lceil \frac{L_e}{2R} \right\rceil$, we can fix $p = \sum_{e \in U_\gamma} \left\lceil \frac{L_e}{2R} \right\rceil$.

The Mixed Integer Non-Linear Programming models that we develop for (MNLCLP) and (PSNLCLP) have $O(p^2|E|)$ variables, $O(p|E|)$ linear constraints, and $O(p|E|f_{\|\cdot\|})$ nonlinear constraints (here, $f_{\|\cdot\|}$ stand for the number of constraints that allow rewriting Constraints (19) as second-order cone constraints (see [32] for upper bounds on this number for ℓ_τ-norms). Thus, it is advisable in these models to design alternative solution strategies for solving them or to provide initial solutions that alleviate the search for optimal solutions by providing lower bounds for our problem. In the following sections, we propose different alternatives taking advantage of the geometric properties of these problems.

4.1. Constructing initial feasible solutions

The geometric properties that we derive in Section 3.1 for the single device problem can be also extended to the p-device case. Specifically, one can construct solutions of MNLCLP by avoiding the computation of covered lengths in the models and assuming that once an edge of the network is touched by the coverage area of a device, the whole is accounted as covered. With these assumptions, we construct initial solutions to our problem by solving the following integer linear programs:

$$\max \sum_{e \in E} \sum_{j \in P} \omega_e L_e z_{je} \tag{29}$$

$$\text{s.t.} \sum_{j \in P} z_{je} \leq 1, \forall e \in E, \tag{30}$$

Table 1
Computational results for the MNLCLP approaches.

Instance	$\lvert V\rvert$	$\lvert E\rvert$	p	R	CPU time (s)			Unsolved		GAP (%)	
					MNLCLP	MNLCLP_1	MNLCLP_2	MNLCLP	MNCLP_1	MNLCLP	MNLCLP_1
gessler	12	14	2	0.1	151.53	13.69	0.89	0%	0%	0%	0%
				0.25	48.97	11.87	1.34	0%	0%	0%	0%
				0.5	26.28	10.59	0.62	0%	0%	0%	0%
			5	0.1	TL	TL	2.26	100%	100%	86%	84%
				0.25	TL	TL	2.92	100%	100%	69%	62%
				0.5	TL	TL	1.61	100%	100%	24%	31%
			8	0.1	TL	TL	3.54	100%	100%	90%	87%
				0.25	TL	TL	5.59	100%	100%	74%	69%
				0.5	TL	TL	2.92	100%	100%	41%	35%
jilin	28	34	2	0.1	167.25	39.10	1.99	0%	0%	0%	0%
				0.25	196.56	144.30	3.37	0%	0%	0%	0%
				0.5	164.83	152.10	2.45	0%	0%	0%	0%
			5	0.1	TL	TL	2.95	100%	100%	86%	85%
				0.25	TL	TL	6.64	100%	100%	72%	64%
				0.5	TL	TL	3.17	100%	100%	40%	42%
			8	0.1	TL	TL	6.07	100%	100%	88%	84%
				0.25	TL	TL	10.34	100%	100%	72%	73%
				0.5	TL	TL	4.67	100%	100%	70%	37%
richmond	48	44	2	0.1	1180.62	133.99	8.75	0%	0%	0%	0%
				0.25	717.09	121.90	7.47	0%	0%	0%	0%
				0.5	184.63	244.25	2.32	0%	0%	0%	0%
			5	0.1	TL	TL	23.22	100%	100%	78%	77%
				0.25	TL	TL	13.79	100%	100%	62%	59%
				0.5	TL	TL	3.70	100%	100%	42%	41%
			8	0.1	TL	TL	33.64	100%	100%	88%	85%
				0.25	TL	TL	23.89	100%	100%	86%	71%
				0.5	TL	TL	5.82	100%	100%	71%	56%
foss	37	58	2	0.1	561.98	39.61	2.77	0%	0%	0%	0%
				0.25	380.54	38.42	1.99	0%	0%	0%	0%
				0.5	196.92	86.40	1.83	0%	0%	0%	0%
			5	0.1	TL	TL	6.49	100%	100%	82%	80%
				0.25	TL	TL	5.46	100%	100%	64%	62%
				0.5	TL	TL	4.31	100%	100%	61%	56%
			8	0.1	TL	TL	9.33	100%	100%	88%	86%
				0.25	TL	TL	7.99	100%	100%	87%	71%
				0.5	TL	TL	9.11	100%	100%	78%	64%
rural	48	60	2	0.1	12263.72	1169.41	16.94	0%	0%	0%	0%
				0.25	TL	559.93	15.69	100%	0%	23%	0%
				0.5	5054.64	1612.73	13.98	0%	0%	0%	0%
			5	0.1	TL	TL	26.46	100%	100%	92%	91%
				0.25	TL	TL	32.19	100%	100%	83%	82%
				0.5	TL	TL	21.99	100%	100%	79%	77%
			8	0.1	TL	TL	40.89	100%	100%	97%	94%
				0.25	TL	TL	49.51	100%	100%	91%	86%
				0.5	TL	TL	40.66	100%	100%	94%	84%
zj	60	85	2	0.1	TL	TL	13.12	100%	100%	49%	65%
				0.25	TL	5235.81	7.29	100%	0%	51%	0%
				0.5	TL	9603.61	9.33	100%	0%	5%	0%
			5	0.1	TL	TL	25.56	100%	100%	96%	95%
				0.25	TL	TL	27.48	100%	100%	90%	89%
				0.5	TL	TL	18.32	100%	100%	87%	86%
			8	0.1	TL	TL	37.85	100%	100%	98%	96%
				0.25	TL	TL	31.05	100%	100%	94%	90%
				0.5	TL	TL	20.45	100%	100%	91%	85%

$$\sum_{e\in S} z_{je} \leq d, \forall S \subset E(|S| = d+1) : \bigcap_{e\in S}(e \oplus \mathbb{B}_{R_j}(0)) = \emptyset, \forall j \in P, \qquad (31)$$

$$z_{je} \in \{0,1\}, \forall e \in E, \forall j \in P. \qquad (32)$$

In the problem above, the overall weighted length of the covered edges is to be maximized by restricting edges to be covered by the same device to those which are feasible for the MNLCLP. The edges are also enforced to be accounted for at most once in the solution.

The strategies for generating and separating the constraints of the above problem are identical to those detailed in Section 3.1.

As can be observed in our computational experience (Section 5), the incorporation of these initial solutions to the original formulation of MNLCLP is advisable to derive exact optimal solutions to the problem in less CPU time.

4.2. Math-heuristic approach

In addition to the exact approaches provided by formulations MNLCLP or PSNLCLP and the generation of initial solutions in the previous

Table 2
Computational results for the PSNLCLP approaches.

Instance	$\vert V\vert$	$\vert E\vert$	γ	R	CPU time (s)		Unsolved	GAP (%)
					PSNLCLP	PSNLCLP_1	PSNLCLP	PSNLCLP
gessler	12	14	0.5	0.1	TL	19.15	100%	96%
				0.25	TL	6.28	100%	89%
				0.5	TL	1.90	100%	75%
			0.75	0.1	TL	30.27	100%	98%
				0.25	TL	10.94	100%	93%
				0.5	TL	3.39	100%	86%
			1	0.1	TL	39.76	100%	97%
				0.25	TL	13.67	100%	93%
				0.5	TL	4.74	100%	89%
jilin	28	34	0.5	0.1	TL	26.40	100%	96%
				0.25	TL	13.29	100%	87%
				0.5	TL	3.44	100%	67%
			0.75	0.1	OoM	54.77	100%	–
				0.25	TL	20.00	100%	95%
				0.5	TL	4.60	100%	88%
			1	0.1	OoM	78.69	100%	–
				0.25	TL	24.36	100%	97%
				0.5	TL	7.05	100%	95%
richmond	48	44	0.5	0.1	TL	57.79	100%	94%
				0.25	TL	14.49	100%	92%
				0.5	TL	3.90	100%	71%
			0.75	0.1	OoM	91.99	100%	–
				0.25	TL	21.33	100%	93%
				0.5	TL	5.62	100%	91%
			1	0.1	OoM	116.10	100%	–
				0.25	TL	25.68	100%	94%
				0.5	TL	7.67	100%	96%
foss	37	58	0.5	0.1	TL	41.95	100%	96%
				0.25	TL	14.21	100%	92%
				0.5	TL	6.83	100%	75%
			0.75	0.1	OoM	111.96	100%	–
				0.25	TL	26.74	100%	97%
				0.5	TL	11.54	100%	94%
			1	0.1	OoM	230.93	100%	–
				0.25	OoM	61.73	100%	–
				0.5	OoM	19.59	100%	–

section, we propose a math-heuristic procedure to obtain good quality approximated solutions for the problems for larger size instances, in less CPU time. The math-heuristic is based on solving, sequentially, the single-device location problem (2)–(8) that was described in Section 3.1.

We show in Algorithm 3 a pseudocode for this procedure. As already mentioned, the approach is based on solving, sequentially, a single-device location device problem until a certain termination criterion (which depends on the problem to solve, MNLCLP or PSNLCLP) is verified. In case the problem is the MNLCLP the algorithm ends when the number of devices in the pool reaches the value of p. Otherwise, for the PSNLCLP the algorithm ends when the covered weighted length reaches the desired value.

At each iteration, a device is located, and the network to be covered in the next iteration is updated from the previous iteration by removing the segments that have been covered.

Algorithm 3: Math-heuristic.

```
Data: Network $G = (V, E; \Omega)$, number of devices $p$ and radius $R$.
$V' = V, E' = E, \Omega' = \Omega$
$X = \emptyset$
while $Termination\_Criterion$ do
    Solve $X', \lambda_e^0, \lambda_e^1, z_e$ = arg (1)–(8) for $e \in E', \omega_e \in \Omega'$ and $R$.
    Update Termination_Criterion Add $X'$ to $X$.
    for $e \in E'$ do
        if $z_e = 1$ then
            if $\lambda_e^0 \in (0,1)$ then
                Add $Y_e^0$ to $V'$.
                Add $\{o_e, Y_e^0\}$ to $E'$.
                Add $\omega_e$ to $\Omega'$.
            if $\lambda_e^1 \in (0,1)$ then
                Add $Y_e^1$ to $V'$.
                Add $\{Y_e^1, f_e\}$ to $E'$.
                Add $\omega_e$ to $\Omega'$.
            Remove $e$ from $E'$
Result: $X \in \mathbb{R}^{d \times p}$: Location of the devices.
```

5. Computational experiments

In this section, we report on the results of a series of computational experiments performed to empirically assess our methodological contribution to the MNLCLP and the PSNLCLP presented in the previous sections. We use six real networks obtained from two different sources: one based on the networks developed by the University of Exeter's (UOE) Centre for Water Systems available in https://emps.exeter.ac.uk/engineering/research/cws/resources/benchmarks/ and other privately provided by Prof. Ormsbee from the University of Kentucky (UKY). These networks, which are called gessler, jilin, richmond, foss, rural and zj, have $14, 34, 44, 58, 60$ and 85 edges, respectively. The networks have been scaled to fit in a disk of radius 5. The networks are drawn in Fig. 8.

We have run the different approaches for the MNLCLP and the PSNLCLP for disk-shaped coverage areas with radii ranging in $\{0.1, 0.25, 0.5\}$. For the MNLCLP the number of devices to locate, p, ranges in $\{2, 5, 8\}$, whereas for the PSNLCLP the values of γ range in $\{0.5, 0.75, 1\}$.

All the experiments have been run on a virtual machine in a physical server equipped with 12 threads from a processor AMD EPYC 7402P 24-Core Processor, 64 Gb of RAM, and running a 64-bit Linux operating system. The models were coded in Python 3.7 and we used Gurobi 9.1 as optimization solver. A time limit of 5 hours was set for all the experiments.

In Tables 1 and 2 we show the average results obtained in our experiments. We report average values of the consumed CPU time (in seconds), and percent of unsolved instances, and the MIP Gap within the time limit. Both tables are similarly organized. In the first block (first three columns), the name of the instance together with its number of nodes and edges is provided. In the second block (next two columns) we write the values of p (for the MNLCLP) or γ (for the PSNLCLP) and the radius. The next three blocks are the results obtained with each of the approaches. For the MNLCLP we run the MISOCO formulation, and also the exact and math-heuristic solution approaches detailed in Section 4.1 (MNLCLP_1, for short) and 4.2 (MNLCLP_2), respectively. We do not report results on the Unsolved instances and MIPGap for the MNLCLP_2 since all the instances were solved within the time limit with that approach. In Table 2 the results are organized similarly for the PSNLCLP, but we do not generate initial solutions since that strategy only applies to the MNLCLP, and only the strategy PSNLCLP_2. The flag `TL` indicates that all the instances averaged in the row reach the time limit without certifying optimality. The flag `OoM` indicates that the solver outputs *Out of Memory* at some point when solving the instance.

The first observation from the results that we obtain is that both problems are computationally challenging since they require large CPU times to solve even the small instances. Actually, the exact MNLCLP was only able to solve up to optimality, small instances with small values of p, and the exact PSNLCLP only solved a few instances, and in many of them, the solver outputs Out of Memory when solving them.

The first alternative (and exact) strategy, MNLCLP_1, based on constructing initial solutions to the problem, had a slightly better performance with respect to those instances that were solved with the initial formulation, both in CPU time and MIPGap. Some of the instances were not able to be solved with MNLCLP but were able to be solved with the initial solutions that we construct.

With respect to the heuristic approach, the consumed CPU times are tiny compared to the times required by the exact approaches and was able to construct feasible solutions for all the instances, even for those that the exact approaches flagged Out of Memory. In terms of the quality of the obtained solutions, in Figs. 9 and 10 we show the average deviations (for each instance) of the alternative approaches with respect to the original one. This measure provides the percent improvement of the alternative method with respect to the best solution obtained by the original formulation of the problem. We observed that the solutions that we obtained with the two strategies are significantly better than those obtained with the original formulation for the MNLCLP within the time limit. Providing initial solutions to the problem allows us to obtain solutions with 20% more coverage than the initial formulation, whereas the heuristic approach gets solutions with more than 25% more coverage. In the case of the PSNLCLP, in most of the instances, the heuristic solutions are even better than those obtained with the exact approach. Nevertheless, in the instance `jilin`, the solutions are almost 50% worse than those obtained with the exact approach.

In Fig. 11 we plot boxplots for the average of the best weighted coverages obtained by each of the approaches within the time limit for each of the six instances. Regarding the distribution of the results, one cannot infer that the results are significantly different. Nevertheless, in terms of the central tendencies, there is a clear improvement in the results obtained with the two alternative methods with respect to the original formulation, especially for the larger instances (see Figs. 9 and 10).

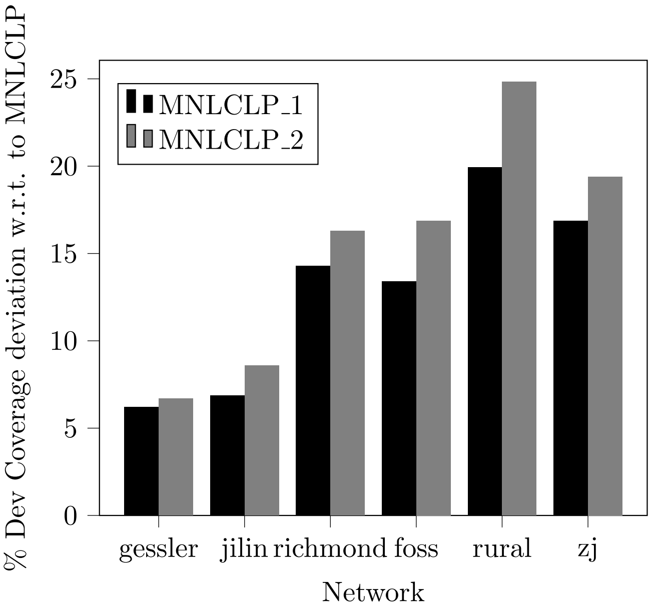


Fig. 9. Average coverage deviations of the MNLCLP_1 and MNLCLP_2 approach with respect to MNLCLP.

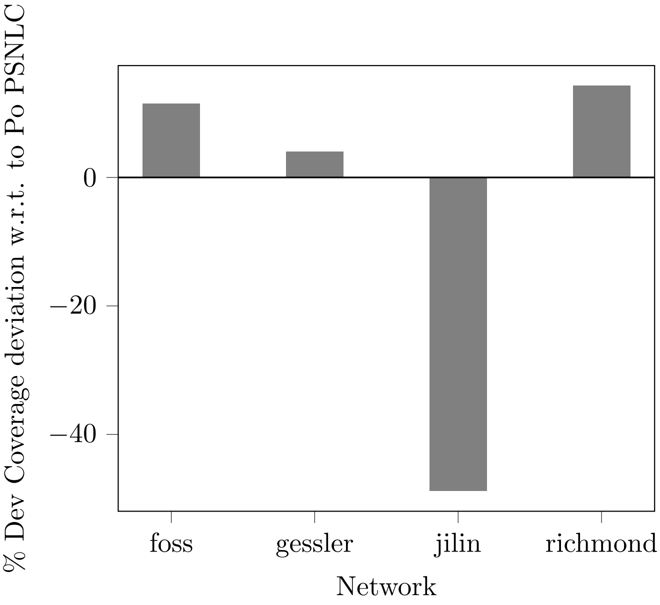


Fig. 10. Average coverage deviations of the PSNLCLP_1 approach with respect to PSNLCLP.

Node and edge-restricted models

Finally, we run some experiments to validate our proposal regarding the coverages obtained with our approaches and those obtained by the node and edge-constrained versions of our model. In those models, the devices are allowed to be located only at the nodes or edges of the network instead of the whole space as in our model. We have run

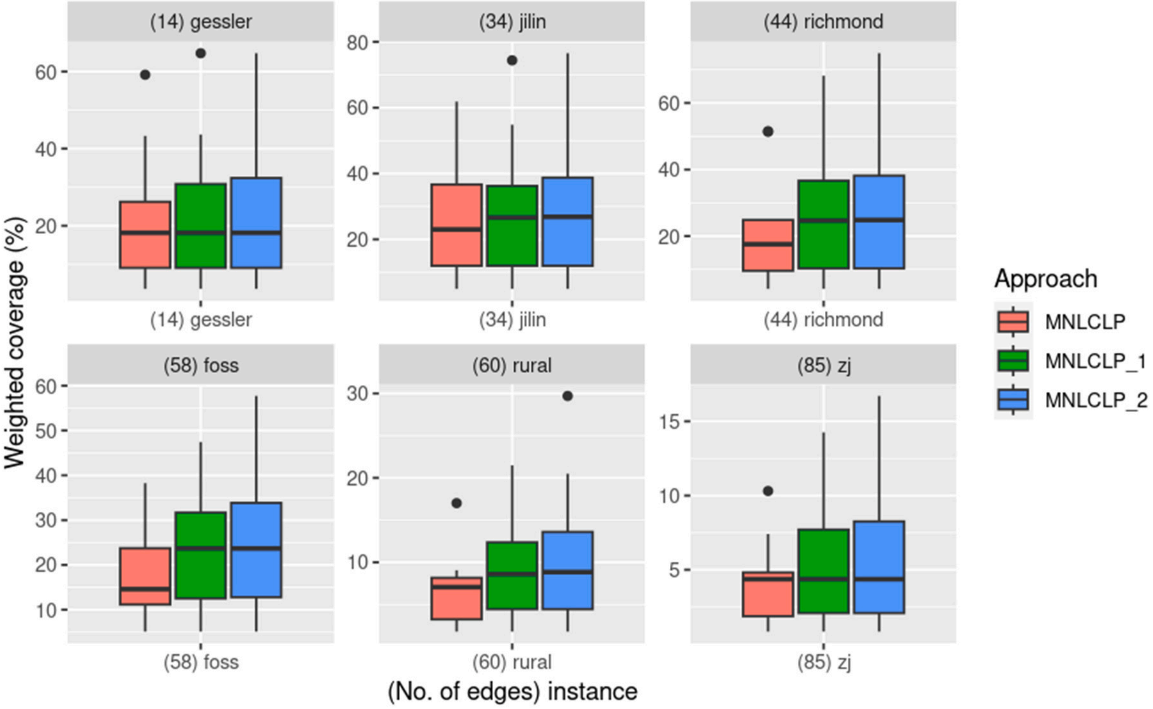


Fig. 11. Weighted coverage comparison between the solutions of the three approaches MNLCLP, MNLCLP_1, and MNLCLP_2.

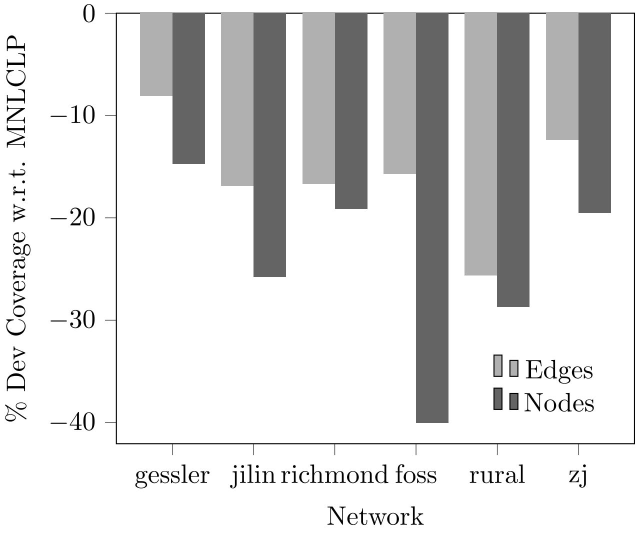


Fig. 12. Average deviations of the coverages obtained with the node and edge-restricted versions of MNLCLP with respect to MNLCLP.

these restricted versions of the Maximal Coverage model, for the same instances and parameters as in the previous section.

First, in Fig. 12 we plot the average deviations of the weighted volume coverages of these approaches with respect to those obtained with our model (the best solution obtained with our solution approaches). One can observe that, in some of the instances, the deviations are close to 40%, that is, our model obtained solutions covering 40% more volume of the network than the restricted versions. As expected, the edge-restricted version covers more network length than the node-restricted model, although, in most instances, the differences are less than 5%.

The detailed average results for the different values of p and R are shown in Table 3. As can be observed, the larger the number of devices to be located, the larger the deviations with respect to the two restricted models. Nevertheless, the deviations do not seem to follow a clear pattern with respect to the radii. For some instances, the deviations are larger for smaller values of R whereas for others the deviations are larger for the larger values.

In Figs. 13 and 14 we show the best solutions obtained with the three models for two of the instances (foss with $p = 5$ and rural with $p = 8$ both of them with $R = 0.5$). In the left plot of both figures, we show the optimal solution of our model. The center and right plots are the solutions obtained with the edge-restricted and node-restricted versions of the MNLCLP, respectively. As can be observed, the optimal location of the devices differs for the different models. Specifically, the MNLCLP takes advantage of locating devices outside the edges of the network to cover edges with a high volume, whereas the restricted versions do not have such flexibility. In Fig. 13, one can observe that, for the foss network, there is a high concentration of weighted volume at the edges in the bottom right corner of the network. Thus, the three models try to locate the devices to cover that demand. In the MNLCLP, a single device (outside the edges and nodes) suffices to cover most of the demand, whereas the restricted models require two or three devices to cover a similar amount of volume. This flexibility directly affects global coverage. For this network, our model was able to cover 10% and 20% more volume than the edge and node-restricted models, respectively. The situation for the rural network (Fig. 14) is even more impressive since our model obtained a solution with more than 30% coverage than the restricted models.

One can conclude that our model is adequate to locate devices that maximize the volume coverage of a network in case they can be located at any place of the space where the network lives, as in the case of leak detection devices, since the obtained solutions significantly outperform (in coverage) the classical node and edge-restricted versions of the problem.

6. Conclusions and future research

In this paper, we study a covering location problem with direct application to the determination of optimal positions of leak detection

Table 3
Average deviations of node and edge-restricted MNLCLP with respect to MNLCLP for the different values of p and R.

Network	p	R	Dev_Nodes	Dev_Edges
gessler	2	0.1	9.94%	1.20%
		0.25	9.84%	1.11%
		0.5	9.82%	1.10%
	5	0.1	11.99%	2.78%
		0.25	13.92%	6.22%
		0.5	11.66%	4.51%
	8	0.1	23.46%	22.10%
		0.25	20.47%	19.13%
		0.5	21.42%	14.82%
jilin	2	0.1	22.72%	0.97%
		0.25	19.88%	1.53%
		0.5	20.38%	10.49%
	5	0.1	24.89%	16.42%
		0.25	18.26%	17.48%
		0.5	21.50%	9.88%
	8	0.1	49.72%	45.95%
		0.25	24.79%	38.00%
		0.5	29.79%	11.23%
richmond	2	0.1	12.62%	2.32%
		0.25	9.53%	3.53%
		0.5	16.67%	9.03%
	5	0.1	13.98%	6.72%
		0.25	19.83%	9.30%
		0.5	14.48%	17.60%
	8	0.1	27.39%	32.84%
		0.25	23.76%	38.96%
		0.5	33.90%	29.89%
foss	2	0.1	32.77%	2.14%
		0.25	32.82%	2.08%
		0.5	26.36%	4.32%
	5	0.1	32.80%	22.35%
		0.25	36.47%	9.38%
		0.5	19.12%	9.55%
	8	0.1	60.73%	40.86%
		0.25	61.87%	17.85%
		0.5	57.66%	32.72%
rural	2	0.1	19.94%	0.67%
		0.25	19.42%	0.43%
		0.5	6.03%	1.14%
	5	0.1	23.77%	22.78%
		0.25	20.44%	45.94%
		0.5	16.37%	39.60%
	8	0.1	40.18%	58.42%
		0.25	80.81%	31.23%
		0.5	31.41%	30.54%
zj	2	0.1	1.32%	0.97%
		0.25	1.31%	0.08%
		0.5	5.86%	2.59%
	5	0.1	14.70%	9.53%
		0.25	5.67%	8.58%
		0.5	12.35%	8.18%
	8	0.1	69.01%	26.19%
		0.25	33.11%	27.69%
		0.5	32.38%	27.59%

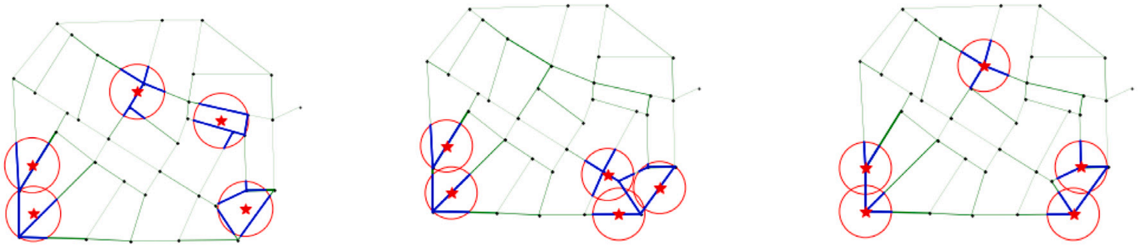


Fig. 13. Solutions of the MNLCLP (left), MNLCLP-edge restricted (center), and MNLCLP-node restricted (right) for instance `foss` ($p = 5$).

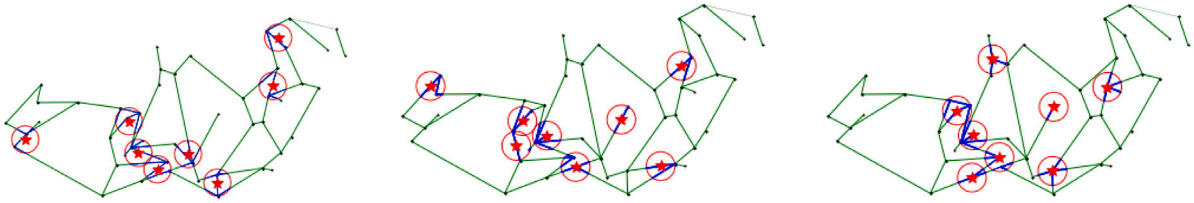


Fig. 14. Solutions of the MNLCLP (left), MNLCLP-edge restricted (center), and MNLCLP-node restricted (right) for instance `rural` ($p = 8$).

devices in urban pipeline networks. We propose a general framework for two different versions of the problem. On the one hand, in case the number of devices is known, we derive the Maximal Network Length Covering Location Problem whose goal is to maximize the length of the network for which the device can detect the leak. On the other hand, in case the number of devices is unknown, the Partial Set Network Length Covering Location Problem aims to minimize the number of devices to locate to be able to detect the leaks in a given percent of the length of the network. We derive a method to construct initial solutions as well as a math-heuristic algorithm. We run our models on different real-world urban water supply pipeline networks and compare the performance of the different proposals.

Future research lines on the topic include incorporating more sophisticated coverage shapes for the devices, as non-convex shapes obtained by the union of different polyhedral and ℓ_τ-norm balls. It would require a further study of τ-order cone constraints, as well as the representation of the union by means of disjunctive constraints, being then a challenge to provide solutions for real-world networks. In this case, it would be advisable to design efficient heuristic approaches capable to scale to large networks adequately.

CRediT authorship contribution statement

Víctor Blanco: Conceptualization, Data curation, Formal analysis, Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Roles/Writing – original draft, Writing – review & editing. **Miguel Martínez-Antón:** Conceptualization, Data curation, Formal analysis,

Funding acquisition, Investigation, Methodology, Project administration, Resources, Software, Supervision, Validation, Visualization, Roles/Writing – original draft, Writing – review & editing.

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.

Data availability

Data will be made available on request.

Acknowledgments

The authors of this research acknowledge financial support by the Spanish Ministerio de Ciencia y Tecnologia, Agencia Estatal de Investigacion and Fondos Europeos de Desarrollo Regional (FEDER) via project PID2020-114594GB-C21 and AEI grant number RED2022-134149-T (Thematic Network: Location Science and Related Problems). The authors also acknowledge partial support from projects FEDER-US-1256951, Junta de Andalucía P18-FR-1422, P18-FR-2369, B-FQM-322-UGR20, NetmeetData: Ayudas Fundación BBVA a equipos de investigación científica 2019, and the IMAG-Maria de Maeztu grant CEX2020-001105-M /AEI /10.13039/501100011033. The first author also acknowledges the financial support of the European Union-Next GenerationEU through the program "Ayudas para la Recualificación del Sistema Universitario Español 2021–2023".

Appendix. Proof of Lemma 8

Proof. Let S be the intersection point between the line induced by e, r, and its orthogonal line passing through the point Q. We denote by μ the parameterization of S in the ray induced by the segment pointed at o_e. Thus, $\|Q-S\| = \min\{\|Q-T\| : T \in r\}$. Since $S \in r$, one can parameterize S as $S = (1-\mu)o_e + \mu f_e$ for some $\mu \in \mathbb{R}$. Let us analyze the different possible values for μ:

- If $\mu \in [0,1]$, one gets that:

$$\begin{aligned}\|Q-(\mu(f_e-o_e)+o_e)\| = \|Q-S\| &= \min\{\|Q-T\| : T \in r\} \\ &\leq \min\{\|Q-T\| : T \in e\} = \delta(e,Q).\end{aligned}$$

- If $\mu < 0$, will show that $\delta(e,Q) = \|Q-o_e\|$. Let $\lambda \in [0,1]$ and $X = (1-\lambda)o_e + \lambda f_e \in e$. Then:

$$\begin{aligned}\|Q-o_e\|^2 &= \|Q-S\|^2 + \|S-o_e\|^2 = \|Q-S\|^2 + \|\mu(f_e-o_e)+o_e-o_e\|^2 \\ &= \|Q-S\|^2 + |\mu|^2\|(f_e-o_e)\|^2 \leq \|Q-S\|^2 + |(\mu-\lambda)|^2\|(f_e-o_e)\|^2 \\ &= \|Q-S\|^2 + \|\mu(f_e-o_e)+o_e-(\lambda(f_e-o_e)+o_e)\|^2 \\ &= \|Q-S\|^2 + \|S-X\|^2 \\ &= \|Q-X\|^2.\end{aligned}$$

- In case $\mu > 1$, let us see that $\delta(e,Q) = \|Q-f_e\|$. Let $\lambda \in [0,1]$ and $X = \lambda(f_e-o_e)+o_e$ be in e:

$$\begin{aligned}\|Q-f_e\|^2 &= \|Q-S\|^2 + \|S-f_e\|^2 = \|Q-S\|^2 + \|\mu(f_e-o_e)+o_e-f_e\|^2 \\ &= \|Q-S\|^2 + \|\mu(f_e-o_e)+o_e-f_e+o_e-o_e\|^2 \\ &= \|Q-S\|^2 + |\mu-1|^2\|(f_e-o_e)\|^2 \\ &\leq \|Q-S\|^2 + |(\mu-\lambda)|^2\|(f_e-o_e)\|^2 \\ &= \|Q-S\|^2 + \|\mu(f_e-o_e)+o_e-(\lambda(f_e-o_e)+o_e)\|^2 \\ &= \|Q-S\|^2 + \|S-X\|^2 \\ &= \|Q-X\|^2.\end{aligned}$$

Summarizing, we get that the point in e closest to Q is in the form $(1-\lambda)o_e + \lambda f_e$ with

$$\lambda = \begin{cases} 0 & \text{if } \mu < 0, \\ \mu & \text{if } 0 \leq \mu \leq 1, \\ 1 & \text{if } \mu > 1. \end{cases}$$

that is, $\lambda = \min\{\max\{0,\mu\},1\}$, being then $\delta(e,Q) = \|Q-(\min\{\max\{0,\mu\},1\}(f_e-o_e)+o_e)\|$. □

References

[1] El-Zahab S, Zayed T. Leak detection in water distribution networks: an introductory overview. Smart Water 2019;4:1–23.

[2] Walski TM. Water supply system rehabilitation. In: Task committee on water supply rehabilitation systems of the water supply and resources management committee of the environmental engineering division of ASCE. American Society of Civil Engineers (ASCE); 1987, p. 336.

[3] El-Abbasy MS, Senouci A, Zayed T, Mirahadi F, Parvizsedghy L. Condition prediction models for oil and gas pipelines using regression analysis. J Constr Eng Manag 2014;140:04014013.

[4] Seckler DW, et al. World Water Demand and Supply, 1990 To 2025: Scenarios and Issues, Vol. 19. Iwmi; 1998.

[5] Hamilton S. Alc in low pressure areas—it can be done. In: Proceedings of 5th IWA water loss reduction specialist conference. 2009, p. 131–7.

[6] El-Abbasy MS, Mosleh F, Senouci A, Zayed T, Al-Derham H. Locating leaks in water mains using noise loggers. J Infrastruct Syst 2016;22:04016012.

[7] Li W, Ling W, Liu S, Zhao J, Liu R, Chen Q, et al. Development of systems for detection, early warning, and control of pipeline leakage in drinking water distribution: A case study. J Environ Sci 2011;23:1816–22.

[8] Fantozzi M, Calza F, Lambert A. Experience and results achieved in introducing district metered areas (dma) and pressure management areas (pma) at Enia utility (Italy). In: Proceedings of the 5th IWA water loss reduction specialist conference. 2009, p. 153–60.

[9] Royal AC, Atkins PR, Brennan MJ, Chapman DN, Chen H, Cohn AG, et al. Site assessment of multiple-sensor approaches for buried utility detection. Int J Geophys 2011;2011.

[10] Cody RA, Dey P, Narasimhan S. Linear prediction for leak detection in water distribution networks. J Pipeline Syst Eng Pract 2020;11:04019043.

[11] Cody RA, Tolson BA, Orchard J. Detecting leaks in water distribution pipes using a deep autoencoder and hydroacoustic spectrograms. J Comput Civ Eng 2020b;34:04020001.

[12] Khulief Y, Khalifa A, Mansour RB, Habib M. Acoustic detection of leaks in water pipelines using measurements inside pipe. J Pipeline Syst Eng Pract 2012;3:47–54.

[13] Puust R, Kapelan Z, Savic D, Koppel T. A review of methods for leakage management in pipe networks. Urban Water J 2010;7:25–45.

[14] Li R, Huang H, Xin K, Tao T. A review of methods for burst/leakage detection and location in water distribution systems. Water Sci Technol Water Supply 2015;15:429–41.

[15] Mohamed IN, El-Hamawi M, Al Hassan RA, Aslanyan A, Filenev M, Aslanyan I, et al. Leak detection by temperature and noise logging. In: Abu Dhabi international petroleum conference and exhibition. OnePetro; 2012.

[16] Tijani I, Abdelmageed S, Fares A, Fan K, Hu Z, Zayed T. Improving the leak detection efficiency in water distribution networks using noise loggers. Sci Total Environ 2022;821:153530.

[17] Venkateswaran P, Han Q, Eguchi RT, Venkatasubramanian N. Impact driven sensor placement for leak detection in community water networks. In: 2018 ACM/IEEE 9th international conference on cyber-physical systems. IEEE; 2018, p. 77–87.

[18] Casillas MV, Puig V, Garza-Castañón LE, Rosich A. Optimal sensor placement for leak location in water distribution networks using genetic algorithms. Sensors 2013;13:14984–5005.

[19] S. García, Marín A. Covering location problems. Locat Sci 2015;9:3–114.

[20] Murray AT, Tong D. Coverage optimization in continuous space facility siting. Int J Geogr Inf Sci 2007;21:757–76.

[21] Murray AT. Contemporary optimization application through geographic information systems. Omega 2021;99:102176.

[22] Pelegrín M, Xu L. Continuous covering on networks: Improved mixed integer programming formulations. Omega 2023;117:102835.

[23] Blanco V, Gázquez R. Continuous maximal covering location problems with interconnected facilities. Comput Oper Res 2021;132:105310.

[24] Blanco V, Gázquez R, Saldanha-da Gama F. Multitype maximal covering location problems: Hybridizing discrete and continuous problems. European J Oper Res 2022. http://dx.doi.org/10.1016/j.ejor.2022.10.037.

[25] Hosseininezhad SJ, Jabalameli MS, Naini SGJ. A continuous covering location model with risk consideration. Appl Math Model 2013;37:9665–76.

[26] Blanquero R, Carrizosa E, Boglárka G, et al. Maximal covering location problems on networks with regional demand. Omega 2016;64:77–85.

[27] Tedeschi D, Andretta M. New exact algorithms for planar maximum covering location by ellipses problems. European J Oper Res 2021;291:114–27.

[28] Blanco V, González G, Hinojosa Y, Ponce D, Pozo MA, Puerto J. Network flow based approaches for the pipelines routing problem in naval design. Omega 2022;111:102659.

[29] Blanco V, González G, Hinojosa Y, Ponce D, Pozo MA, Puerto J. The pipelines and cable trays location problem in naval design. Ocean Eng 2023;286:115525.

[30] Berman O, Wang J. The minmax regret gradual covering location problem on a network with incomplete information of demand weights. European J Oper Res 2011;208:233–8.

[31] Berman O, Kalcsics J, Krass D. On covering location problems on networks with edge demand. Comput Oper Res 2016;74:214–27.

[32] Blanco V, Puerto J, El Haj Ben Ali S. Revisiting several problems and algorithms in continuous location with ℓ_τ norms. Comput Optim Appl 2014;58:563–95.

[33] Helly E. Über mengen konvexer körper mit gemeinschaftlichen punkte. Jahresber Dtsch Math-Ver 1923;32:175–6 [in German].