Maximal Covering Location Problems on networks with regional demand *

Rafael Blanquero,
Universidad de Sevilla, Spain,
rblanquero@us.es
Emilio Carrizosa,
Universidad de Sevilla, Spain,
ecarrizosa@us.es
Boglárka G.-Tóth,
Budapest University of Technology and Economics,
Hungary,
bog@math.bme.hu

Abstract

Covering problems are well studied in the Operations Research literature under the assumption that both the set of users and the set of potential facilities are finite. In this paper, we address the following variant, which leads to a Mixed Integer Nonlinear Program (MINLP): locations of p facilities are sought along the edges of a network so that the expected demand covered is maximized, where demand is continuously distributed along the edges. This MINLP has a combinatorial part (which edges of the network are chosen to contain facilities) and a continuous global optimization part (once the edges are chosen, which are the optimal locations within such edges).

A branch-and-bound algorithm is proposed, which exploits the structure of the problem: specialized data structures are introduced to successfully cope with the combinatorial part, inserted in a geometric branch-and-bound algorithm.

*Research partially supported by research grants and projects Hungarian National Research, Development and Innovation Office – NKFIH, OTKA grant PD115554, ICT COST Action TD1207 (EU), MTM2012-36163 (Ministerio de Ciencia e Innovación, Spain), P11-FQM-7603, FQM329 (Junta de Andalucía, Spain), all with EU ERDF funds.

Computational results are presented, showing the appropriateness of our procedure to solve covering problems for small (but non-trivial) values of p.

Key words: Maximal Covering Location Problem. Location on networks. Regional demand. Global optimization. Branch and bound.

1 Introduction

The Maximal Covering Location Problem, (MCLP), [3, 14, 15, 22], is a classic problem in locational analysis with applications in a good number of fields, such as health care, emergency planning, ecology, statistical classification, homeland security, see e.g. [1, 8, 13, 18, 39, 40] and the references therein. Given a finite set of users A, each $a \in A$ with demand $\omega_a \geq 0$, a set of p facilities in a set F is sought in order to maximize the demand covered. A point is said to be covered by a set $F^* \subset F$ of p facilities if there is at least one $f \in F^*$ at distance from a not greater than R, where $R > 0$ is a fixed number, called the *covering radius.*
(MCLP) is easily expressed as an Integer Program. Indeed, defining binary variables y_f and z_a to indicate respectively whether a facility at f is open, and whether a is covered, (MCLP) amounts to solving the following program:

$$\begin{aligned} \max \quad & \sum_{a \in A} \omega_a z_a \\ \text{s.t.} \quad & z_a \leq \sum_{f \in F:\, d(a,f) \leq R} y_f && \forall a \in A \\ & \sum_{f \in F} y_f = p \\ & y_f \in \{0,1\} && \forall f \in F \\ & z_a \in \{0,1\} && \forall a \in A. \end{aligned} \tag{1}$$

(MCLP) is known to be NP-hard, [27], but formulated as (1) is, in words of [37], integer-friendly, in the sense that its continuous relaxation is often all-integer, and thus no much branching is usually needed in a branch-and-bound algorithm. See [23, 29, 36, 38] and the references therein for heuristic approaches to handle problems of larger size.
Extensions and closely related models to the (MCLP) abound in the Operations Research literature. First, (MCLP) has been studied assuming that the space is not a discrete set but a network: the set A of users is the set of nodes of a network N, and facilities are allowed to be located not only at

the nodes, but anywhere on N. It is shown, however, that one only needs to consider a finite and relatively small set of candidate locations, [14, 27], and thus the problem can be written in the form of (MCLP) above. Nontrivial extensions include, for instance, replacing the basic yes/no covering function to more general decreasing functions in the distance separating the user and the facility, [3, 4, 2, 5]; another variant is found when the set A of users is finite, but the feasible locations are assumed to be a subset of the plane, yielding planar covering models, as reviewed in [33].
Much less literature exists on covering models with *regional demand*, [21, 26, 31], in which, by the very nature of the problem, assuming the demand to be concentrated at a finite set (e.g. centroids of neighbourhoods, towns, administrative units or census boundaries, [31]) is a crude approximation. The consequences of inaccuracies due to such discretization are well studied, [16, 28, 31], and thus demand is advocated to be modeled as following a continuous distribution on a given region. See also [9, 10, 11] for other location models with continuously distributed demand.

The following version of the classic (MCLP) with regional demand is addressed in this paper: demand is assumed to be continuously distributed along the edges of a network and p points along the set of edges of the network are sought in order to maximize the expected covering of the demand. Hence, the model differs from the classic (MCLP) in two main issues: first, the set of feasible locations is not a discrete set, but (a set of) the edges of a network; moreover, demand is assumed here to be distributed along the edges of the network, making it a realistic model, for instance, for covering problems in an urban context, in which users are located along streets (the edges), or for the location of emergency services to attend accidents, which take place along the roads (edges of the transportation network).

Let us now introduce formally the problem under consideration. We are given a network $N = (V, E)$; each edge $e \in E$ has associated its length l_e, which allows us to talk about points in an edge: edge e, with endpoints u, v, is identified with the interval $[0, l_e]$, and we thus identify any $x \in [0, l_e]$ as the point in the edge e at distance x of u and distance $l_e - x$ of v. With this identification, the shortest-path distance between the nodes in V is readily extended to a metric d on the points in the edges. Moreover, each edge e has a weight $\omega_e \geq 0$ and a probability density function (pdf) f_e, which models the demand along edge e. We assume that a radius $R > 0$ is given, and a point x along an edge $e \in E$ is covered by the set of facilities at $t_1, \ldots, t_p$ if

$$\min_{1 \leq i \leq p} d(t_i, x) \leq R. \tag{2}$$

The expected demand of edge e covered by facilities at $\mathbf{t} = (t_1, \ldots, t_p)$ is given by

$$\omega_e \int_0^{l_e} \delta_e(x;\mathbf{t}) f_e(x)\, dx,$$

where $\delta_e(x;\mathbf{t})$ takes the value 1 when $x \in e$ is covered by facilities at $\mathbf{t} = (t_1, \ldots, t_p)$, i.e., when (2) is fulfilled, and takes the value 0 otherwise.
With this, the optimization problem at hand can be written as

$$\max_{\mathbf{t} \in E^p} C(\mathbf{t}) := \sum_{e \in E} \omega_e \int_0^{l_e} \delta_e(x;\mathbf{t}) f_e(x)\, dx. \tag{3}$$

The remainder of this note is structured as follows. In Section 2, structural properties of the MINLP (3) are studied. A branch-and-bound method is designed in Section 3. Exploiting the structure of the problem, data structures and bounding procedures are proposed, and they are tested on a set of instances in Section 4. The paper ends with some concluding remarks and possible extensions in Section 5.

2 Structural properties

Property 2.1. *For any p-tuple of edges $(e_1, \ldots, e_p) \in E^p$, the function C : $\mathbf{t} = (t_1, \ldots, t_p) \in [0, l_{e_1}] \times \ldots \times [0, l_{e_p}] \longrightarrow C(\mathbf{t})$ is continuous in $[0, l_{e_1}] \times \ldots \times [0, l_{e_p}]$.*

Proof. Using the inclusion-exclusion principle, we can re-write $C(\mathbf{t})$ as

$$C(\mathbf{t}) = \sum_{e \in E} \omega_e \int_0^{l_e} \sum_{I \subset \{1,\ldots,p\}} (-1)^{1+|I|} \prod_{i \in I} \delta_e(x;t_i) f_e(x)\, dx.$$

Hence, it suffices to show that, for any $e = (u, v) \in E$ and any nonempty I, the function $\int_0^{l_e} \prod_{i \in I} \delta_e(x;t_i) f_e(x)\, dx$ is continuous in $\mathbf{t}$. Split the index set I in those indices corresponding to facilities in e and not in e respectively:

$$\begin{aligned} I_+ &:= \{i \in I : e_i = e\} \\ I_- &:= \{i \in I : e_i \neq e\}. \end{aligned}$$

Observe that, for $i \in I_+$, one has

$$\begin{aligned} \delta_e(x;t_i) = 1 \quad &\text{iff} \quad d(x, t_i) \leq R \\ &\text{iff} \quad x \in [t_i - R, t_i + R], \end{aligned}$$

while for $i \in I_-$,

$$\begin{aligned}\delta_e(x;t_i) = 1 \quad & \text{iff} \quad \min\{x + d(u,t_i), l_e - x + d(v,t_i)\} \leq R \\ & \text{iff} \quad x \in [0, R - d(u,t_i)] \cup [d(v,t_i) + l_e - R, l_e]\end{aligned}$$

Hence

$$\begin{aligned}
\prod_{i\in I_+} \delta_e(x;t_i) = 1 \quad & \text{iff} \quad x \in [\max_{i\in I_+} t_i - R, \min_{i\in I_+} t_i + R] \\
\prod_{i\in I_-} \delta_e(x;t_i) = 1 \quad & \text{iff} \quad x \in [0, R - \max_{i\in I_-} d(u,t_i)] \cup [\max_{i\in I_-} d(v,t_i) + l_e - R, l_e] \\
\prod_{i\in I} \delta_e(x;t_i) = 1 \quad & \text{iff} \\
& x \in [\max\{\max_{i\in I_+} t_i - R, 0\}, \min\{\min_{i\in I_+} t_i + R, R - \max_{i\in I_-} d(u,t_i)\}] \\
& \cup [\max\{\max_{i\in I_+} t_i - R, \max_{i\in I_-} d(v,t_i) + l_e - R\}, \min\{\min_{i\in I_+} t_i + R, l_e\}] \\
& = [a_1(\mathbf{t}), b_1(\mathbf{t})] \cup [a_2(\mathbf{t}), b_2(\mathbf{t})].
\end{aligned}$$

Hence,

$$\begin{aligned}
\int_0^{l_e} \prod_{i\in I} \quad & \delta_e(x;t_i) f_e(x)dx = \int_{[a_1(\mathbf{t}),b_1(\mathbf{t})]\cup[a_2(\mathbf{t}),b_2(\mathbf{t})]} f_e(x)dx \\
& = \int_{a_1(\mathbf{t})}^{b_1(\mathbf{t})} f_e(x)dx + \int_{a_2(\mathbf{t})}^{b_2(\mathbf{t})} f_e(x)dx - \int_{\max\{a_1(\mathbf{t}),a_2(\mathbf{t})\}}^{\min\{b_1(\mathbf{t}),b_2(\mathbf{t})\}} f_e(x)dx \\
& = \max\{F_e(b_1(\mathbf{t})) - F_e(a_1(\mathbf{t})), 0\} + \max\{F_e(b_2(\mathbf{t})) - F_e(a_2(\mathbf{t})), 0\} \\
& - \max\{F_e(\min\{b_1(\mathbf{t}), b_2(\mathbf{t})\}) - F_e(\max\{a_1(\mathbf{t}), a_2(\mathbf{t})\}), 0\},
\end{aligned}$$

where F_e is the cumulative distribution function associated with the pdf f_e. Since F_e is continuous, $C(\mathbf{t})$ is continuous as well. □

Once the p-tuple of edges $(e_1, \ldots, e_p)$ is chosen, the function C is continuous on the compact set $[0, l_{e_1}] \times \ldots \times [0, l_{e_p}]$, and attains its maximum. Since the possible choices of p-tuple of edges is also finite, the maximum of C on E^p is attained. Finding such maximum may be hard because, for arbitrary pdfs f_e defining the demand along the edges, the function C may not be convex, and thus Global Optimization techniques are to be used; in its full generality, C may lack important structural properties, such as Lipschitz-continuity. This is shown in the following example.

Example 2.1. *Consider a graph $N = (V, E)$ with two nodes, v_1, v_2, connected by an edge e of length 2, so that we can identify the edge with the segment*

$[-1,1]$ and the nodes with the endpoints of the segment. The density f_e of the demand is given by

$$f_e(x) = \frac{1}{4\sqrt{|x|}}, \qquad x \in [-1,1].$$

Consider the problem of locating one facility ($p = 1$) on e, and a coverage radius $R = 1/4$. Let us study the behavior of the function C on the interval $[-1,1]$. First, the cumulative distribution F_e is easily shown to be given by

$$F_e(x) = \begin{cases} 0, & \text{if } x < -1 \\ \frac{1-\sqrt{-x}}{2}, & \text{if } -1 \leq x < 0 \\ \frac{1+\sqrt{x}}{2}, & \text{if } 0 \leq x < 1 \\ 1, & \text{if } x \geq 1. \end{cases} \tag{4}$$

On the other hand, C is given by

$$C(x) = F_e\left(x + \frac{1}{4}\right) - F_e\left(x - \frac{1}{4}\right) \tag{5}$$

Joining (4) and (5) one obtains after some algebra the following expression of C :

$$C(x) = \begin{cases} \frac{1-\sqrt{-x-1/4}}{2}, & \text{if } -1 \leq x < -\frac{3}{4} \\ \frac{\sqrt{-x+1/4}-\sqrt{-x-1/4}}{2}, & \text{if } -\frac{3}{4} \leq x < -\frac{1}{4} \\ \frac{\sqrt{x+1/4}+\sqrt{-x+1/4}}{2}, & \text{if } -\frac{1}{4} \leq x < \frac{1}{4} \\ \frac{\sqrt{x+1/4}-\sqrt{x-1/4}}{2}, & \text{if } \frac{1}{4} \leq x < \frac{3}{4} \\ \frac{1-\sqrt{x-1/4}}{2}, & \text{if } \frac{3}{4} \leq x < 1 \end{cases}$$

Observe that the function C has infinite directional derivatives at points $x = \pm\frac{1}{4}$, which are interior to the interval $[-1,1]$. Hence C cannot be Lipschitz-continuous in the interval $[-1,1]$.

Under some reasonable assumptions on the pdfs involved, the function C is Lipschitz-continuous:

Property 2.2. *Suppose that, for each $e \in E$, the pdf f_e is bounded above by some constant M. Then, for any p-tuple of edges $(e_1,\ldots,e_p) \in E^p$, the function $C : \mathbf{t} = (t_1,\ldots,t_p) \in [0,l_{e_1}] \times \ldots \times [0,l_{e_p}] \longrightarrow C(\mathbf{t})$ is Lipschitz-continuous in $[0,l_{e_1}] \times \ldots \times [0,l_{e_p}]$.*

Proof. Let $\mathbf{t} = (t_1, \ldots, t_p), \mathbf{s} = (s_1, \ldots, s_p) \in [0, l_{e_1}] \times \ldots \times [0, l_{e_p}]$. One has

$$|C(\mathbf{t}) - C(\mathbf{s})| \leq \sum_{e \in E} \omega_e \int_0^{l_e} |\delta_e(x; \mathbf{t}) - \delta_e(x; \mathbf{s})| M \, dx. \tag{6}$$

Now, for $x \in e := (u, v)$, $|\delta_e(x; \mathbf{t}) - \delta_e(x; \mathbf{s})| > 0$ iff one of the two following conditions holds:

$$\delta_e(x; \mathbf{t}) = 1, \qquad \delta_e(x; \mathbf{s}) = 0, \tag{7}$$
$$\delta_e(x; \mathbf{t}) = 0, \qquad \delta_e(x; \mathbf{s}) = 1. \tag{8}$$

Let us study separately the two cases, by identifying necessary conditions which must hold and are more manageable. If (7) holds, then, there exists some $i \in \{1, \ldots, p\}$, $e_i = (a_i, b_i)$ such that one of the following conditions holds:

$$\begin{array}{ll}
e_i \neq e, & t_i + d(a_i, u) + x \leq R < d(s_i, x) \\
e_i \neq e, & l_{e_i} - t_i + d(b_i, u) + x \leq R < d(s_i, x) \\
e_i \neq e, & t_i + d(a_i, v) + l_e - x \leq R < d(s_i, x) \\
e_i \neq e, & l_{e_i} - t_i + d(b_i, v) + l_e - x \leq R < d(s_i, x) \\
e_i = e, & |x - t_i| \leq R < x - s_i \\
e_i = e, & |x - t_i| \leq R < -x + s_i,
\end{array}$$

which imply respectively the following:

$$\begin{array}{ll}
e_i \neq e, & t_i + d(a_i, u) + x \leq R < s_i + d(a_i, u) + x \\
e_i \neq e, & l_{e_i} - t_i + d(b_i, u) + x \leq R < l_{e_i} - s_i + d(b_i, u) + x \\
e_i \neq e, & t_i + d(a_i, v) + l_e - x \leq R < s_i + d(a_i, v) + l_e - x \\
e_i \neq e, & l_{e_i} - t_i + d(b_i, v) + l_e - x \leq R < l_{e_i} - s_i + d(b_i, v) + l_e - x \\
e_i = e, & x - t_i \leq R < x - s_i \\
e_i = e, & -x + t_i \leq R < -x + s_i,
\end{array}$$

i.e.,

$$e_i \neq e, \qquad x \in (-s_i - d(a_i, u) + R, -t_i - d(a_i, u) + R] \tag{9}$$
$$e_i \neq e, \qquad x \in (-l_{e_i} + s_i - d(b_i, u) + R, -l_{e_i} + t_i - d(b_i, u) + R] \tag{10}$$
$$e_i \neq e, \qquad x \in [t_i + d(a_i, v) + l_e - R, s_i + d(a_i, v) + l_e - R) \tag{11}$$
$$e_i \neq e, \qquad x \in [l_{e_i} - t_i + d(b_i, v) + l_e - R, l_{e_i} - s_i + d(b_i, v) + l_e - R) \tag{12}$$
$$e_i = e, \qquad x \in (s_i + R, t_i + R] \tag{13}$$
$$e_i = e, \qquad x \in [t_i - R, s_i - R). \tag{14}$$

If, instead of (7), (8) holds, then conditions analogous to (9)-(14) are obtained, but exchanging the roles of s_i and t_i.
Hence, by (6) one has

$$\begin{aligned}
|C(\mathbf{t}) - C(\mathbf{s})| &\leq \sum_{e \in E} \omega_e \int_0^{l_e} |\delta_e(x;\mathbf{t}) - \delta_e(x;\mathbf{s})| M \, dx \\
&\leq \sum_{e \in E} \omega_e 2 \left(\sum_{i:\, e_i \neq e} 4|t_i - s_i| + 2 \sum_{i: e_i = e} |t_i - s_i| \right) M \\
&\leq \sum_{e \in E} \omega_e 8pM \|\mathbf{t} - \mathbf{s}\|_\infty,
\end{aligned}$$

and thus C is Lipschitz-continuous. □

3 A global optimization approach

A branch-and-bound algorithm is proposed to cope with this MINLP. As in any branch-and-bound procedure, the two key elements are the branching and the bounding strategies, which are discussed in Sections 3.1 and 3.2, respectively. Firstly, we define the splitting rules, which take advantage of the structure of the problem, by taking into account that the variables indicating the number of facilities per edge should be strongly correlated: if facilities are located at a given edge, it is unlikely that more facilities are located in neighboring edges, leaving big clusters of edges uncovered. Bounding strategies for such subdivision elements will then be built. Other important algorithmic issues of our proposal, such as the selection, elimination and termination rules, are outlined in Section 3.3.

3.1 Division rule

One first and naive approach is to decide first how many facilities are located within each edge, and then, once these variables are fixed, one solves, by means of a standard branch-and-bound algorithm on networks, e.g. [6, 7], the nonlinear optimization problem of deciding where to locate them. However, full inspection of all p-tuples of edges will be doable only for very small networks. For this reason, our approach is to facilitate branching on the combinatorial and the continuous part at the same time.
In order to avoid the enumeration of every possible combination of p edges, we propose to construct clusters of (sub)edges. Instead of associating with each edge an integer variable indicating the number of facilities to be located

in such edge, the integer variables will be associated with the clusters of (sub)edges, called hereafter *edgesets*, and the tuple of edgesets will be called *superset*.
To be precise, an edgeset is a finite collection of (sub)edges of E; a superset S is any tuple of the form $(E_1, p_1; E_2, p_2; \ldots, E_k, p_k)$, where $E_1, E_2, \ldots, E_k$ are disjoint edgesets and $p_1, \ldots, p_k$ are strictly positive integer numbers with

$$\sum_{j=1}^{k} p_j = p,$$

indicating, for each $j = 1, \ldots, k$, that exactly p_j facilities are to be located within the points in E_j.

Example 3.1. *Consider the network depicted in Figure 1, with all lengths equal to* 1*, uniform demand on each edge, weights ω_e given by*

$$\begin{array}{lcl} \omega_{12} & = & 2 \\ \omega_{14} & = & 1 \\ \omega_{23} & = & 1 \\ \omega_{34} & = & 1 \\ \omega_{45} & = & 2 \\ \omega_{46} & = & 1 \\ \omega_{56} & = & 1 \\ \omega_{67} & = & 1, \end{array} \tag{15}$$

and suppose $p = 3$ facilities are to be located for a covering radius $R = 1/4$. The partition of E in the three edgesets E_1, E_2, E_3,

$$\begin{array}{lcl} E_1 & = & \{(1,2), (1,4), (2,3), (3,4), (4,6)\} \\ E_2 & = & \{(6,7)\} \\ E_3 & = & \{(4,5), (5,6)\} \end{array} \tag{16}$$

induces, among others, the superset S

$$S = (E_1, 2; E_2, 1), \tag{17}$$

which corresponds to the decision of locating two facilities in the edges of E_1 and one facility in the edges of E_2.

Supersets will correspond to nodes in the branch-and-bound tree. We discuss in what follows our proposal to build the starting nodes, and the way to sequentially subdivide the supersets.

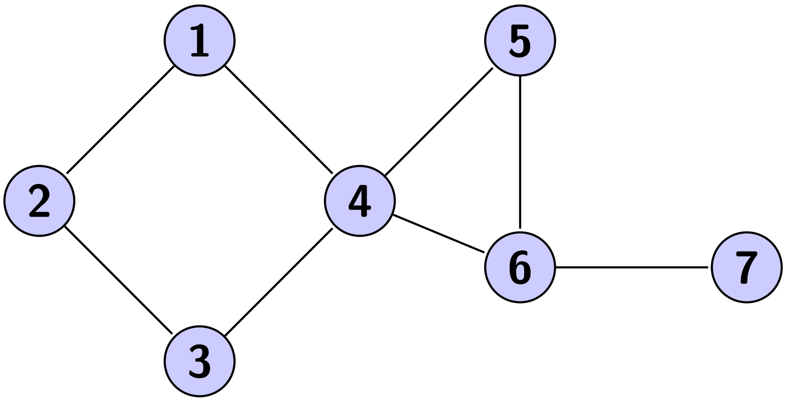

Figure 1: Example of a network

3.1.1 Initial supersets

The root node of our branch-and-bound tree is the superset $S_0 = (E, p)$. S_0 is subdivided by using a given partition $E_1, E_2, \ldots, E_k$ of E : we add to the branch-and-bound tree list $\binom{p+k-1}{p}$ supersets of the form $(E_{i_1}, p_1; \ldots; E_{i_l}, p_l)$, where $\{i_1, ..., i_l\} \subseteq \{1, ..., k\}$ and $p_1 + \ldots + p_l = p$. Observe that, although such starting list will have a cardinality exponentially increasing in p, the difficulty of the MINLP under study only allows us to handle problems with a low value of p. Hence, the cardinality of the starting list will not grow much.

A critical issue is how the edges of the network, conforming the initial superset S_0, are split into edgesets in such a way that the so-obtained subdivision fits with the actual distribution of facilities at the optimal solution of the problem. To do this, we build from the network a discrete (MCLP) as follows: we consider a discrete covering problem in which we have, as possible locations, the edges of the network, we have as users also the edges e of the network, with demand ω_e, and we define the distance $d^*(e, f)$ between user e and edge f as the smallest distance between the points in e and f. Then, we consider a user e covered if $d^*(e, f) \leq R$ for some edge f. Hence, we count an edge e as fully covered (and thus, the weight ω_e is taken) as soon as some point in some f is at distance not greater than R from some point in e. Once this discrete (MCLP) is solved, and $f_1^*, \ldots, f_p^*$ is an optimal solution, we build the edgesets $E_1, \ldots, E_p$ so that E_j contains the edges e for which f_j^* is the closest facility.

Let us illustrate this procedure with the data of Example 3.1 for $p = 2$. The distance matrix d^* is then given by

	$(1,2)$	$(1,4)$	$(2,3)$	$(3,4)$	$(4,5)$	$(4,6)$	$(5,6)$	$(6,7)$
$(1,2)$	0	0	0	1	1	1	2	2
$(1,4)$	0	0	1	0	0	0	1	1
$(2,3)$	0	1	0	0	1	1	2	2
$(3,4)$	1	0	0	0	0	0	1	1
$(4,5)$	1	0	1	0	0	0	0	1
$(4,6)$	1	0	1	0	0	0	0	0
$(5,6)$	2	1	2	1	0	0	0	0
$(6,7)$	2	1	2	1	1	0	0	0

Solving such (MCLP) yields as an optimal solution the edges $f_1^* = (1,2)$ and $f_2^* = (4,6)$, and, starting from them, the edgesets

$$\begin{aligned} E_1 &= \{(1,2),(1,4),(2,3)\} \\ E_2 &= \{(3,4),(4,5),(4,6),(5,6),(6,7)\}, \end{aligned}$$

where, in case of ties in d^*, edges have been allocated randomly. With such definition of E_1, E_2, three supersets are obtained as split of the starting superset S_0, namely $(E_1,2),(E_1,1;E_2,1),(E_2,2)$, represented in Figure 2.

3.1.2 Subdivision of a superset

In order to guarantee convergence of the branch-and-bound algorithm, elements in the list should become arbitrarily small. Let us define the *diameter* $\lambda(E^*)$ of an edgeset E^* as the sum of the lengths of the (sub)edges in E^*, and define the diameter $\lambda(S)$ of a superset S as the highest length of its edgesets with assigned facilities,

$$\lambda(E_1,p_1;E_2,p_2;\ldots;E_k,p_k) = \max_j \lambda(E_j).$$

Reduction of the diameters of the supersets in the list guides our subdivision strategy. Superset $S = (E_1,p_1;E_2,p_2;\ldots;E_k,p_k)$ is subdivided as follows: first, the edgeset E_{j^*} with highest diameter is found,

$$\lambda(E_1,p_1;E_2,p_2;\ldots;E_k,p_k) = \lambda(E_{j^*}).$$

Then, the edgeset E_{j^*} is split into two subsets by identifying two "central" edges, and then clustering the edges around such edges. The process, similar to the one described in Section 3.1.1 for splitting the initial set, is based on the construction of an auxiliary (MCLP): a 2-facility discrete covering

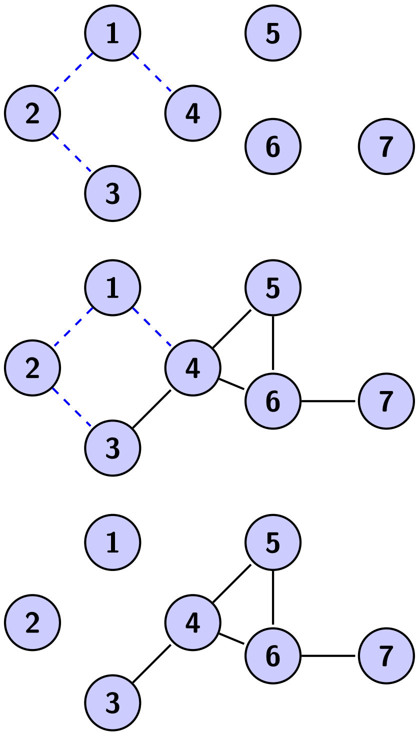

Figure 2: Splitting the starting superset

problem is considered, in which we have, as possible locations, the edges of the edgeset E_{j^*}, we have as users the edges e of the network, with demand ω_e, and we define the distance $d^*(e, f)$ between user e and edge f as the smallest distance between the points in e and f. Then, we consider a user e covered if $d^*(e, f) \leq R$ for some edge f. Once this discrete (MCLP) is solved and an optimal solution f^+, f^- is obtained, we build the sets $E_{j^*}^+$ and $E_{j^*}^-$ so that $E_{j^*}^+$ contains the edges $e \in E_{j^*}$ for which f^+ is the closest facility.
Given the splitting of E_{j^*} into $E_{j^*}^+$ and $E_{j^*}^-$, the superset S is subdivided into $p_{j^*} + 1$ supersets, by assigning respectively i and $p_{j^*} - i$ facilities to $E_{j^*}^+$ and $E_{j^*}^-$, $i = 0, 1, \ldots, p_{j^*}$.
By construction, one immediately has

Property 3.1. *The given subdivision of the supersets is exhaustive, that is, for an infinite nested series of supersets* $\{S_q\}_{q=0}^{\infty}$, $\lambda(S_q) \to 0$ *as* $q \to \infty$.

3.2 Bounding Rules

As in any branch-and-bound algorithm, procedures for giving lower and upper bounds are needed here. Lower bounds on the objective C of (3) are obtained by evaluating C at heuristic solutions, built as the midpoints of p (sub)edges in the superset under evaluation. Different strategies for obtaining upper bounds are described in Sections 3.2.1–3.2.3.

3.2.1 Shadow Bound

An easy way to obtain an upper bound for C on the superset S is to consider as covered all points in S as well as those at distance at most R of some point in S. In other words, a bound is obtained if one considers as covered the points both in S and the "shadow" of S, i.e., those points at distance R from points in S. Formally, the Shadow Bound, $C_{SB}(S)$, for C on the superset $S = (E_1, p_1; \ldots, E_k, p_k)$ is calculated as

$$\overline{C}_{SB}(S) := \sum_{e \in E} \omega_e \int_0^{l_e} \delta_e^{SB}(x; S) f_e(x)\, dx, \tag{18}$$

where $\delta_e^{SB}(x; S)$ takes the value 1 when x is at distance at most R of some $y \in E_j$ and takes the value 0 otherwise.
For instance, for the data of Example 3.1 and the superset S in (17), we have

$$\begin{aligned}
\delta_e^{SB}(x; S) &= 1 \quad \forall x \in [0,1],\ \forall e \in E_1 \cup E_2 \\
\delta_{(4,5)}^{SB}(x; S) &= \begin{cases} 1, & \text{if } x \in [0, 1/4] \\ 0, & \text{else} \end{cases} \\
\delta_{(5,6)}^{SB}(x; S) &= \begin{cases} 1, & \text{if } x \in [3/4, 1] \\ 0, & \text{else} \end{cases}
\end{aligned}$$

Then, given the weights in (15), one obtains

$$\overline{C}_{SB}(S) = 6 + 2\frac{1}{4} + \frac{1}{4} = \frac{27}{4}.$$

By construction, the Shadow Bound has the important property of monotonicity, in the sense that, if $S = (E_1, p_1; \ldots, E_k, p_k)$ and $S' = (E_1', p_1; \ldots, E_k', p_k)$ are supersets satisfying $E_i \supseteq E_i'$ for all i, then

$$\overline{C}_{SB}(S) \geq \overline{C}_{SB}(S'). \tag{19}$$

Moreover, using the same arguments than in the proof of Property 2.1, if $\{(s_1^q, 1; \ldots, s_p^q, 1)\}_q$ is a sequence of supersets, where each s_j is a subedge of an edge e_j converging to some point t_j, then $\overline{C}_{SB}((s_1, 1; \ldots, s_p, 1)) = C(t_1, \ldots, t_p)$. Hence, the bounds go arbitrarily sharp when the length of the supersets goes to zero. Consequently, having an exhaustive division rule and a convergent bounding rule, a branch-and-bound method using this bound is convergent.

3.2.2 MCLP Bound

The upper bound $\overline{C}_{MCLP}$ is obtained by solving a variant of a discrete (MCLP) as (1): we consider a discrete covering problem in which we have, as possible locations, the (sub)edges of the edgesets of the superset $S = (E_1, p_1; \ldots; E_k, p_k)$, we have as users the edges e of the network, with demand ω_e, and we define the distance $d^*(e, f)$ between user e and (sub)edge f as the smallest distance between the points in e and f. Then, we consider a user e covered if $d^*(e, f) \leq R$ for some (sub)edge f of some edgeset E_j.
Hence, we count an edge e as fully covered (and thus, the weight ω_e is taken) as soon as some point in some f is at distance not greater than R from some point in e. Moreover, since the number p_j of facilities within each edgeset E_j is given, we impose at most p_j different edges in E_j are to be chosen.
By construction, the optimal value of such discrete covering problem is a valid upper bound of C on S :

$$\begin{array}{lll} \max & \displaystyle\sum_{e \in E} \omega_e z_e & \\ \text{s.t.} & z_e \leq \displaystyle\sum_{f \in \cup_j E_j} a_{ef} y_f & \forall e \in E \\ & \displaystyle\sum_{f \in E_i} y_f \leq p_i, & i = 1, 2, \ldots, k \\ & y_f \in \{0, 1\} & \forall f \in \cup_j E_j \\ & z_e \in \{0, 1\} & \forall e \in E, \end{array} \tag{20}$$

where a_{ef} is the scalar taking the value 1 if $f \in E_j$ for some j with $d^*(e, f) \leq R$, and taking the value 0 otherwise.
Contrary to what happens with the Shadow Bound $\overline{C}_{SB}$, this bound may not be sharp enough in small supersets, since, if any point of an edge is covered, then all the demand of that edge is considered as covered. For this reason, the bounding approach is not convergent.
This bound can easily be sharpened by observing that, by construction, for an edge e, if at least one point in some f in some E_j is at distance not greater

than R, we are considering in (20) all the demand of the edge e covered, whilst a smaller amount, ω_e^*,

$$\omega_e^* = \omega_e \int_0^{l_e} \delta_e^{SB}(x,S) f_e(x)\, dx \tag{21}$$

can be captured. Here, $\delta_e^{SB}(x,S)$, as defined in the Shadow Bound (18), takes the value 1 when x is at distance at most R of some $x \in E_j$ and takes the value 0 otherwise.

In this paper we call MCLP bound $\overline{C}_{MCLP}$ as the optimal value of problem (20) after replacing in the objective the weights ω_e by the weights ω_e^* in (21). Observe that the MCLP bound is, by construction, monotonic. Moreover, when each edgeset is part of one edge, the bound obtained is exactly the Shadow Bound, and thus it will enjoy the same convergence properties as the Shadow Bound. Note also that, since an upper bound is needed, a (more crude but less expensive) upper bound is obtained if, instead of the IP (20), its LP relaxation is solved.

3.2.3 Mixed Bound

The upper bounds $\overline{C}_{SB}$ and $\overline{C}_{MCLP}$ above described usually work well if the covering areas have big overlapping parts. When, on the contrary, the areas covered are almost disjoint, the problem could be split into a series of (almost) independent single-facility problems, successfully yielding sharp bounds.

More precisely, for $S = (E_1, p_1; \dots; E_k, p_k)$, we can combine the Shadow Bound $\overline{C}_{SB}(E_j, 1)$ on E_j with any upper bound $\overline{C}_1(E_j)$ for the problem of locating one facility at some point in E_j. This way, the so-called Mixed Bound $\overline{C}_{MB}(S)$ is defined as

$$\overline{C}_{MB}(S) = \sum_{j=1}^{k} \min \left\{ p_j \overline{C}_1(E_j), \overline{C}_{SB}(E_j, 1) \right\},$$

where $\overline{C}_{SB}(E_j, 1)$ is the Shadow Bound on E_j. So the problem is reduced to obtaining an upper bound for the single-facility problem with the edgeset E_j as set of candidate points. If $\mathcal{F}_j$ is a collection of small subedges of the network with

$$E_j \subseteq \bigcup_{f \in \mathcal{F}_j} f,$$

then one can take as upper bound $\overline{C}_1(E_j)$ the maximum of the Shadow

Bounds for locating one facility on f, when f varies in the class $\mathcal{F}_j$,

$$\overline{C}_1(E_j) = \max_{f \in \mathcal{F}_j} \overline{C}_{SB}((f,1)),$$

yielding

$$\overline{C}_{MB}(S) = \sum_{j=1}^{k} \min\left\{ p_j \max_{f \in \mathcal{F}_j} \overline{C}_{SB}((f,1)), \overline{C}_{SB}(E_j,1) \right\}.$$

As an illustration consider the network in Example 3.1 and the superset S in (17). If, for each edgeset E_j, we define the split $\mathcal{F}_j$ as the edges of the network in E_j, we have:

$$\begin{array}{lcl}
\overline{C}_1(E_1) & = & \max\{\overline{C}_{SB}((1,2),1), \overline{C}_{SB}((1,4),1), \overline{C}_{SB}((2,3),1), \overline{C}_{SB}((3,4),1)\} \\
 & = & \max\{10/4, 9/4, 7/4, 8/4\} = 10/4 \\
\overline{C}_{SB}(E_1,1) & = & 7 \\
\overline{C}_1(E_2) & = & 5/4 \\
\overline{C}_{SB}(E_2,1) & = & 3/2 \\
\overline{C}_{MB}(S) & = & 2 \cdot 10/4 + 5/4 = 25/4.
\end{array}$$

Note that, by construction, the Mixed Bound $\overline{C}_{MB}$ is monotonic. However, since it calculates separately the covering of each edgeset E_j, in case of overlapping in the areas covered, such points are counted more than once. Hence, the bound is not necessarily convergent.

3.3 Further algorithmic issues

In order to have a functional method, some other rules are necessary, although these are some of the usual rules.

Selection Rule: The next superset to be evaluated is the one with the highest upper bound on the list.

Elimination Rule: Whenever a superset S is such that $\overline{C}(S) < LB$, any possible location of the facilities in the edgesets of S would lead to a worse covering that the best solution we have so far, therefore the set S can be omitted from further consideration.

Termination Rule: When the relative error of the largest upper bound and the best found solution is less then the tolerance ε, the algorithm stops. The supersets remaining on the list contain the global optimum, and the best solution found so far is reported.

4 Computational Results

Our branch and bound was implemented in Fortran 90 (Intel©Fortran Compiler XE 12.0), using the integration tools of the IMSL Fortran Numerical Library and calling the MIP solver of Cplex 12.5. Executions were carried out on an Intel Core i7 computer with 8.00 Gb of RAM memory at 2.8 GHz, running Windows 7.

Two types of experiments were performed. First, a series of networks of medium size, obtained e.g. from [6, 19], were solved for a small number p of facilities: $p = 2, 3, 4$. In order to analyze the impact of p on the running times, we have tested our method on a small network, the Sioux-Falls, taken from [24].

Let us describe now the first experiment class. Problems on 7 test networks obtained are solved. The number of nodes of these networks ranges from 150 to 225, and the number of edges from 296 to 386. Demand parameters are randomly generated: the overall demand ω_e of an edge e is assumed to follow a uniform distribution in $[0, l_e]$, and the demand along each edge is assumed to follow a beta distribution with parameters randomly generated in the interval [0.1, 5], which provides a wide range of density functions with very different shapes. We stress that we have chosen the beta distribution just because the beta class is versatile enough and it requires numerical integration routines for evaluation, so the usefulness of the method is demonstrated in a difficult case. However, arbitrary densities could have been used instead.

On each network, the problem is solved for p facilities, $p = 2, 3, 4$, and three different radii R, a small, a medium and a large one with respect to the diameter of the networks.

The stopping criterion is set to the relative gap of 10^{-3} for all problems.

In order to see the efficiency of the bounding rules, different settings, using the different bounding schemes proposed in the paper, were compared. In all cases, the Shadow Bound $\overline{C}_{SB}$ was calculated to guarantee convergence of the branch-and-bound algorithm, and, if needed, to compute the coefficients ω_e^* in the MCLP bound $\overline{C}_{MCLP}$. The following strategies were tested:

SB: Just the Shadow Bound is calculated.

MCLP: In addition to the Shadow Bound (needed to calculate ω_e^*), the MCLP bound is also calculated.

MB: Both the Shadow Bound and the Mixed Bound are calculated.

ALL: All three bounds, namely the Shadow Bound, the MCLP and the Mixed Bound, are calculated.

Smart: Heuristic bound rule, where, for every third level in the division tree at each superset, all the bounding rules are calculated. The most efficient rule is stored for each superset, where efficiency is measured by means of a merit function which combines sharpness of the bounds and computational time: i is the most efficient bound if for any bound j it holds that $2^{R_{UB}-1}R_T > 1$, where $R_{UB} = \frac{\overline{C}_j-\tilde{f}}{\overline{C}_i-\tilde{f}}$ is the ratio of overestimations, and $R_T = \frac{T_j}{T_i}$ is the ratio of computational time for bounds j and i; otherwise the second best bound is chosen.

Once the most efficient bound is identified, only such bound is calculated for their descendants in the next two levels.

In Tables 1-3, running times in seconds of the different bounding approaches are presented for the different values of p and R. In the tables results are grouped by the radius, and average values are also shown. For the instances which did not terminate in 5 hours (18000 sec), the achieved relative gap is reported. The best approach for each problem is highlighted.

In Table 1 the results for $p = 2$ are shown. One can see clearly the not surprising differences from one approach to the other with respect to the radius. Namely, while for the SB and MCLP approaches running time is decreasing as R is increasing, for MB is just the opposite. The balance of forces is already clear: although SB and MCLP are good for large radius, MB is necessary for small and medium R. Our Smart rule is shown to be the best for small and medium radii, while for large R almost always SB was the most efficient.
In Table 2 the running times and achieved gaps are shown for $p = 3$. For the SB and MCLP approaches, most problems with small radius are intractable, since the gap after 5 hours of running time is still over 15-25% on average. With the exponential growth of possibilities for the solution, the MB approach gets more useful. This happens because the evaluation of the Mixed Bound is expensive rather at the beginning of the algorithm, when the maximal bound for each edge is calculated, but it takes almost no time until bounds on small segments have to be evaluated. While from the pure bounding rules MB is almost always the best, the Smart approach still has a slightly better performance.
In Table 3 results for $p = 4$ are shown for only the MB, ALL, and Smart approaches, since SB and MCLP can solve only the PR152G problem with large R. Although the Smart approach is still the best one on average, we can see that the average time is very similar for the different approaches. This is due to the fact that many problems were stopped after 5 hours, making averages similar (and high).

Table 1: Running times ($p = 2$).

Graph	R	SB	MCLP	MB	ALL	Smart
KROA150G	small	286.4	271.2	34.8	37.2	30.7
KROA200G	small	413.8	373.7	36.1	38.5	33.4
KROB150G	small	833.4	847.3	67.3	69.5	54.2
KROB200G	small	789.0	770.5	53.2	56.7	45.7
PR152G	small	171.5	182.6	20.8	21.6	19.3
RAT195G	small	2021.5	2000.9	37.8	40.0	31.9
TS225G	small	301.4	293.3	14.6	16.9	14.1
Average		688.1	677.1	37.8	40.1	32.8
KROA150G	medium	384.9	378.5	45.7	47.1	38.6
KROA200G	medium	269.2	258.9	92.2	98.7	86.3
KROB150G	medium	287.0	282.0	34.2	37.8	31.2
KROB200G	medium	538.5	544.7	190.2	202.2	181.3
PR152G	medium	12.3	16.8	6.1	6.5	6.0
RAT195G	medium	716.6	696.4	112.8	116.5	91.2
TS225G	medium	242.8	178.4	29.3	32.7	25.6
Average		350.2	336.5	72.9	77.4	65.7
KROA150G	large	2.3	3.4	21.0	22.0	21.7
KROA200G	large	607.0	622.2	669.2	694.1	677.9
KROB150G	large	32.2	36.6	55.0	61.0	55.2
KROB200G	large	2.2	3.8	25.1	26.8	25.0
PR152G	large	15.7	20.3	9.8	12.1	11.0
RAT195G	large	2.8	6.4	22.7	26.6	24.4
TS225G	large	44.9	50.1	65.1	71.3	62.7
Average		101.0	106.1	124.0	130.6	125.4
Average	–	379.8	373.2	78.2	82.7	74.6

Let us discuss now the second experiment. In order to see how the results change as p grows, the Smart bounding rule was used for a very small (24 nodes and 39 edges) network, namely, the Sioux-Falls network, [24].
In Table 4 computational times are given for $p = 2, \ldots, 7$, and, as in the first type of experiments, three different radii. For the large radius, when $p = 6, 7$ more than $100,000$ supersets needed to be stored in the list; this was the maximum allowed in the program, so the reached gap was also reported in

Table 2: Running times and gaps ($p = 3$).

Graph name	R	SB T(s)	SB Gap	MCLP T(s)	MCLP Gap	MB T(s)	ALL T(s)	Smart T(s)
KROA150G	small	–	0.254	–	0.218	337.8	355.0	285.0
KROA200G	small	–	0.149	–	0.098	243.3	252.3	181.7
KROB150G	small	–	0.276	–	0.206	156.1	164.1	124.2
KROB200G	small	–	0.209	–	0.142	453.1	446.7	363.0
PR152G	small	15770.3	–	16863.6	–	37.2	43.9	31.3
RAT195G	small	–	0.633	–	0.451	93.1	112.2	72.6
TS225G	small	–	0.103	–	0.086	167.5	183.4	121.7
Average		17681.5	0.232	17837.7	0.172	212.6	222.5	168.5
KROA150G	medium	12146.4	–	11096.7	–	269.8	298.9	238.5
KROA200G	medium	4410.2	–	3992.2	–	111.8	120.4	99.0
KROB150G	medium	–	0.001	16591.4	–	632.8	652.6	477.1
KROB200G	medium	6332.9	–	4678.2	–	198.6	196.0	144.7
PR152G	medium	1009.3	–	1103.3	–	25.1	27.8	24.3
RAT195G	medium	–	0.101	–	0.071	3804.5	3794.3	3329.6
TS225G	medium	–	0.038	–	0.005	210.6	241.5	182.6
Average		11128.4	0.021	10494.5	0.012	750.5	761.6	642.3
KROA150G	large	3072.0	–	3178.4	–	2987.0	3035.1	2978.9
KROA200G	large	4481.7	–	4675.3	–	3155.2	3277.2	3158.3
KROB150G	large	1993.0	–	1960.2	–	752.8	770.4	700.6
KROB200G	large	270.7	–	284.9	–	282.8	304.7	277.7
PR152G	large	77.2	–	106.2	–	17.0	22.0	18.2
RAT195G	large	150.4	–	169.3	–	181.9	201.6	182.7
TS225G	large	2686.5	–	1951.6	–	1872.5	1525.7	1574.3
Average		1818.8	0.001	1760.8	0.001	1321.3	1305.2	1270.1
Average	–	10209.6	0.085	10031.0	0.061	761.5	763.1	693.6

these cases.

Observe that for the small and large radii, an explosion in running times happens from $p = 5$ to $p = 6$, whereas for the medium radius it is rather from $p = 4$ to $p = 5$. It is also interesting to see that the difficulty can be very different from problem to problem, as for small radius and $p = 7$ facilities,

Table 3: Running times and gaps ($p = 4$).

Graph name	R	MB T(s)	MB Gap	ALL T(s)	ALL Gap	Smart T(s)	Smart Gap
KROA150G	small	–	0.003	–	0.003	–	0.003
KROA200G	small	2612.6	–	2792.8	–	2068.5	–
KROB150G	small	4367.5	–	4867.2	–	3658.3	–
KROB200G	small	–	0.019	–	0.020	–	0.014
PR152G	small	225.5	–	261.5	–	171.8	–
RAT195G	small	1862.2	–	2105.0	–	1423.3	–
TS225G	small	435.7	–	535.6	–	357.2	–
Average		6500.5	0.004	6651.7	0.004	6239.9	0.003
KROA150G	medium	2428.6	–	2573.0	–	1875.6	–
KROA200G	medium	846.8	–	881.9	–	736.5	–
KROB150G	medium	5619.4	–	5771.9	–	4694.6	–
KROB200G	medium	4403.2	–	4581.3	–	3406.7	–
PR152G	medium	9976.9	–	10681.3	–	8892.0	–
RAT195G	medium	–	0.049	–	0.049	–	0.047
TS225G	medium	432.0	–	749.1	–	405.9	–
Average		5958.1	0.008	6176.9	0.008	5430.2	0.008
KROA150G	large	–	0.044	–	0.041	–	0.042
KROA200G	large	–	0.013	–	0.013	–	0.013
KROB150G	large	–	0.004	–	0.005	–	0.004
KROB200G	large	–	0.048	–	0.042	–	0.042
PR152G	large	16.1	–	21.5	–	17.9	–
RAT195G	large	–	0.049	–	0.044	–	0.045
TS225G	large	–	0.002	10603.9	–	10914.5	–
Average		15430.9	0.023	14375.1	0.021	14418.9	0.021
Average	–	9296.5	0.012	9067.9	0.011	8696.3	0.011

it can be solved faster than the same problem with 6 facilities. This may be due to the number of local optima which are close to the global optima, or due to the flatness of the objective function near the global optimizer. Even though more extensive testing needs to be performed to fully understand the dependence of running times of the covering problems with respect to all the parameters involved, it is clear from our tests that the running times increase

Table 4: Running times and gaps for the Sioux-Falls network.

R	$p = 2$	$p = 3$	$p = 4$	$p = 5$	$p = 6$		$p = 7$	
	T(s)	T(s)	T(s)	T(s)	T(s)	Gap	T(s)	Gap
small	3.0	5.4	20.7	91.4	10652.8	–	6487.5	–
medium	10.2	29.2	257.7	6123.1	38222.6	–	58451.4	–
large	9.8	147.7	1256.5	1493.6	49633.6	0.0012	46297.7	0.14

exponentially when p increases.

5 Conclusions

In this paper we have studied a covering location problem on networks which, contrary to those already in the literature, assumes the demand distributed along the edges of the network, which is a more realistic assumption for problems with networks representing high-density regions, such as cities. The problem is a challenging MINLP, in which combinatorial decisions (which edges of the network are to be selected to contain facilities) are coupled with continuous decisions (where to locate the facilities once the edges are chosen). A branch-and-bound algorithm has been developed for this MINLP. While some ingredients of such branch and bound are standard, the branching procedure is rather specific, since it successfully exploits the fact that the locational decisions are taken on a network. Different bounding rules are proposed and tested on different networks; it is shown that the so-called Smart strategy, which through a learning process, is identifying for each branch-and-bound node the most promising branching strategy, is the most promising in terms of running times.
For the resolution of the problem, we have also considered a special type of superset, where no information about the number of facilities in each edgeset is stored. For these supersets similar bounding rules can be derived, although in some cases giving looser bounds. The advantage of this data structure is that it reduces the exponential growth of the number of supersets as p increases, but for the number of facilities in the experiments we have performed the results were very similar. However it may give better results for higher number of facilities, and thus we believe this alternative approach deserves further analysis and testing.

Several extensions of the problem are possible, and in most cases the bound-

ing strategies proposed in this paper could be adapted to such extensions. To mention a few, the most straightforward extension would be the addition of capacity constraints to the covering model, as proposed e.g. in [32]. On the other hand, we have assumed the demand along each edge to follow an absolutely continuous random variable. The more general case in which the demand is expressed as a mixture of an absolutely continuous random variable and a discrete variable with finite support can be handled in the same way, by splitting the edges at the preprocessing step into subedges in such a way that the cover of points with positive mass is constant along each subedge.
A third line of extensions would consist of including congestion effects, as proposed for (standard) covering models e.g. in [12, 25]. This calls also for the re-definition of the objective, since, in this case, the potential users causing the congestion are not identified by a finite set. The fourth and most challenging extension consists of incorporating in the covering problem competition issues, [17, 20, 34, 35]: in a leader-follower problem, the location of the follower is a covering problem, similar to the one described here; solving the leader problem is a much harder problem than the one addressed here, since one has to solve a bilevel problem in which the follower strategy is the one described in this paper. This, as well as the other extensions, deserve further study, not only by its implications in location analysis (more realistic models for dense demand are considered) but also from the Global Optimization viewpoint, since new, challenging MINLPs are addressed with new branch-and-bound procedures.

References

[1] Adenso-Díaz, B. and F. Rodríguez, "A simple search heuristic for the MCLP: Application to the location of ambulance bases in a rural region", *Omega* **25** (1997) 181–187.

[2] Berman, O. and D. Krass, "The Generalized Maximal Covering Location Problem," *Computers & Operations Research* **29** (2002) 563–581.

[3] Berman, O., Z. Drezner and D. Krass, "Generalized coverage: New developments in covering location models", *Computers & Operations Research* **37** (2010) 1675–1687.

[4] Berman, O., Z. Drezner and D.Krass, "Continuous covering and cooperative covering problems with a general decay function on net-

works", *Journal of the Operational Research Society* **64** (2013) 1644–1653.

[5] Berman, O., D. Krass and Z. Drezner, "The Gradual Covering Decay Location Problem on a Network", *European Journal of Operational Research* **151** (2003) 474–480.

[6] Blanquero, R. and E. Carrizosa, "Solving the median problem with continuous demand on a network", *Computational Optimization and Applications* **56** (2013) 723–734.

[7] Blanquero, R., E. Carrizosa, A. Nogales-Gómez and F. Plastria, "Single-facility huff location problems on networks", *Annals of Operations Research* **222** (2014) 175–195.

[8] Bonn, A., A.S.L. Rodrigues and K.J. Gaston, "Threatened and endemic species: are they good indicators of patterns of biodiversity on a national scale?", *Ecology Letters* **5** (2002) 733–741.

[9] Carrizosa, E. and B. G.-Tóth, "Anti-Covering Problems". In *Location Science* (G. Laporte, S. Nickel, F. Saldanha da Gama, Eds.), 115–132. Springer Springer International Publishing, 2015.

[10] Carrizosa, E. and F. Plastria, "Optimal expected-distance separating halfspace", *Mathematics of Operations Research* **33** (2008) 662–677.

[11] Carrizosa, E., M. Munoz-Márquez and J. Puerto, "Optimal location and design of a competitive facility", *Operations Research* **46** (1998) 155–156.

[12] Choo, S.-H., H. Jang, T. Lee and J. Turner, "Simultaneous location of trauma centers and helicopters for emergency medical service planning", *Operations Research* **62** (2014) 751–777.

[13] Chung, C.-H., "Recent Applications of the Maximal Covering Location Planning (M.C.L.P.) Model", *Journal of the Operational Research Society* **37** (1986) 735–746.

[14] Church, R.L. and M. E. Meadows, "Location modeling using maximum service distance criteria," *Geographical Analysis* **11** (1979) 358–373.

[15] Church, R.L., and C.S. ReVelle, "The maximal covering location problem", *Papers of the Regional Science Association* **32** (1974) 101–118.

[16] Current, J.R. and D.A. Schilling, "Analysis of errors due to demand data aggregation in the set covering and maximal covering location problems", *Geographical Analysis* (1990) **22** 116–126.

[17] Dasci, A. and G. Laporte, "A continuous model for multistore competitive location", *Operations Research* **53** (2005) 263–280.

[18] Daskin, M.S. and L.K. Dean, "Location of Health Care Facilities", *Operations Research and Health Care* **70** (2004) 43–76.

[19] Data instances for arc routing problems. Available at www.uv.es/corberan/instancias.htm

[20] Dobson, G. and U.S. Karmarkar, "Competitive location on a network", *Operations Research* **35** (1987) 565–574.

[21] Drezner, Z., and A. Suzuki, "Covering continuous demand in the plane", *Journal of the Operational Research Society* **61** (2010) 878–881.

[22] Galvao, R., and C.S. ReVelle, "Lagrangean heuristic for the maximal covering location problem", *European Journal of Operational Research* **88** (1996) 114–123.

[23] Galvao, R. D., L.G.A. Espejo and B. Boffey, "A comparison of Lagrangian and surrogate relaxations for the maximal covering location problem", *European Journal of Operational Research* **124** (2000) 377–389.

[24] LeBlanc, L.J., E.K. Morlok, and W.P. Pierskalla, "An efficient approach to solving the road network equilibrium traffic assignment problem", *Transportation Research*, **9** (1975) 309–318.

[25] Marianov V., and C. ReVelle, "The queueing maximal availability location problem: A model for the siting of emergency vehicles", *European Journal of Operational Research* **93** (1996) 110–120.

[26] Matisziw, T. C., and A.T. Murray, "Siting a facility in continuous space to maximize coverage of continuously distributed demand", *Socio-Economic Planning Sciences* **43** (2009) 131–139.

[27] Megiddo, N., E. Zemel and S.L. Hakimi, "The maximum coverage location problem," *SIAM Journal on Algebraic and Discrete Methods*, **4** (1983) 253–261.

[28] Miller, H.J., "GIS and geometric representation in facility location problems", *International Journal of Geographical Information Systems* **10** (1996) 791-–816.

[29] Murray, A.T, and R.L. Church, "Applying simulated annealing to location-planning models", *Journal of Heuristics* **2** (1996) 31–53.

[30] Murray, A.T. and M.E. O'Kelly, "Assessing representation error in point-based coverage modeling", *Journal of Geographical Systems* (2002) **4** 171–191.

[31] Murray, A.T., M.E. O'Kelly and R.L. Church, "Regional service coverage modeling", *Computers & Operations Research* **35** (2008) 339–355.

[32] Pirkul, H. and D.A. Schilling, "The Maximal Covering Location Problem with Capacities on Total Workload", *Management Science* **37** (1991) 233–248.

[33] Plastria, F., "Continuous covering location problems". In *Facility location: Applications and theory* (Z. Drezner, H. W. Hamacher, Eds.), 37–79. Springer Verlag, 2002, Berlin.

[34] Plastria, F. and E. Carrizosa, "Optimal location and design of a competitive facility", *Mathematical Programming* **100** (2004) 247–265.

[35] Plastria, F. and L. Vanhaverbeke, "Discrete models for competitive location with foresight", *Computers & Operations Research* **35** (2008) 683–700.

[36] Resende, M.G.C., "Computing Approximate Solutions of the Maximum Covering Problem with GRASP", *Journal of Heuristics* **4** (1998) 161–177.

[37] ReVelle, C., "Facility siting and integer-friendly programming", *European Journal of Operational Research* **65** (1993) 147–159.

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

[39] Schilling, D. A., V. Jayaraman and R. Barkhi, "A review of covering problem in facility location", *Location Science* **1** (1993) 25–55.

[40] Wright, P.D., M.J. Liberatore and R.L. Nydick, "A Survey of Operations Research Models and Applications in Homeland Security", *Interfaces* **36** (2006) 514–529.