The Ordered Gradual Covering Location Problem on a Network

Oded Berman a, Jörg Kalcsics b,∗, Dmitry Krass a, Stefan Nickel b,c

a Rotman School of Management, University of Toronto, 105 St. George Street, Toronto, ONT M5S 3E6, Canada
b Institute of Operations Research, Karlsruhe Institute of Technology, Kaiserstr. 12, 76131 Karlsruhe, Germany
c Fraunhofer ITWM, Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany

ARTICLE INFO

Article history:
Received 15 February 2008
Received in revised form 11 March 2009
Accepted 7 August 2009
Available online 13 October 2009

Keywords:
Gradual Covering
Ordered Median Function
Network location

ABSTRACT

In this paper we develop a network location model that combines the characteristics of ordered median and gradual cover models resulting in the Ordered Gradual Covering Location Problem (OGCLP). The Gradual Cover Location Problem (GCLP) was specifically designed to extend the basic cover objective to capture sensitivity with respect to absolute travel distance. The Ordered Median Location problem is a generalization of most of the classical locations problems like p-median or p-center problems. The OGCLP model provides a unifying structure for the standard location models and allows us to develop objectives sensitive to both relative and absolute customer-to-facility distances. We derive Finite Dominating Sets (FDS) for the one facility case of the OGCLP. Moreover, we present efficient algorithms for determining the FDS and also discuss the conditional case where a certain number of facilities is already assumed to exist and one new facility is to be added. For the multi-facility case we are able to identify a finite set of potential facility locations a priori, which essentially converts the network location model into its discrete counterpart. For the multi-facility discrete OGCLP we discuss several Integer Programming formulations and give computational results.

© 2009 Elsevier B.V. All rights reserved.

1. Introduction

In this paper we develop a network location model that combines the characteristics of ordered median and gradual cover models. As discussed below, this provides a "unifying structure" for the standard location models, and allows us to consider combined objectives that are sensitive to both relative and absolute customer-to-facility travel distances.

The three "classical" location problems are the median problem for which the objective function is minimizing the total travel distance from customers to facilities, the center problem, for which the objective function is minimizing the maximum travel distance (i.e., the travel distance for the customer who has to travel the furthest to get to a facility), and the max cover problem, for which the objective function is maximizing the number of customers covered by the facilities, where customers are considered to be covered if they are within a certain coverage radius of a facility. A good recent overview of location models on networks can be found in [4].

Since each of these objectives covers some important aspects of the underlying location problem, there has also been a considerable interest in combinations of these objectives (e.g., the "cent–dian" objective, which is a convex combination of the median and center objectives, [14] and references therein). An important step in this direction has been the recent development of the Ordered Median Location Problem (OMP) that provides a unifying framework for the location models with median and center objectives, as well as the objectives that combine aspects of the two. (The OMP on networks was first introduced in [13] and the discrete version in [12]; see [14] for a comprehensive treatment.)
This unifying framework is accomplished as follows. Assuming there are \( n \) customers located at the nodes of the network, for given facility locations the median objective can be thought of as a two-step process: (1) compute the distance from each customer to the closest facility, and (2) add up the components of this vector to obtain the total travel distance. The facility locations are then chosen so as to minimize this total travel distance. The OMP interjects two additional steps into this process: (1A) “sorting”, where customer travel distances are sorted from smallest to largest, and (1B) “weighting” where the \( i \)-th smallest customer travel distance is multiplied by the weight \( \lambda_i \in \mathbb{R} \) (the weights do not have to be sorted); the weighted costs are now added up as in step (2) above. This allows us to represent the standard median objective (by choosing weights \( \lambda_i = 1 \) for all \( i \)), the center objective (by choosing weights \( \lambda_i = 0 \) for the first \( n - 1 \) components and \( \lambda_n = 1 \) for the last component, we obtain the largest travel distance), and the cent-dian objective (to obtain a convex combination of the median and center objectives with weight \( \alpha \in (0, 1) \), set \( \lambda_i = \alpha \) for all the first \( n - 1 \) components and \( \lambda_n = 1 \) for the last component). Thus, the median and the center models, as well as their combinations, are special cases of the OMP. In addition, by using different weight vectors, many other objectives can be represented [14].

Note however, that one shortcoming of the OMP model is that it can only represent objectives based on relative travel distances – i.e., the travel distance of one customer relative to the other customers (e.g., in the center problem we try to find a location for the new facility such that the largest distance from a client to the facility relative to the others is a small as possible). In many settings, the absolute travel distances may be more relevant (e.g., a customer located over 10 km from a supermarket is unlikely to patronize it, even if the store happens to be the closest to the customer). In fact, retailers typically define their trading areas in terms of the number of potential customers within a certain distance from the store. Unfortunately, it is not possible to capture sensitivity with respect to absolute travel distances (e.g., assign a higher weight to customers within 2 km from a facility) within the standard OMP framework.

The Gradual Cover Location Problem (GCLP), described in [1,3], was specifically designed to extend the basic cover objective to capture sensitivity with respect to absolute travel distance. This model replaces the fixed coverage radius of the cover objective with a “coverage decay function” which assigns a coverage weight (a value between 0 and 1) to each customer based on the customer’s distance from the closest facility. The objective is to maximize the weighted sum of covered customers. For example, the coverage decay function may specify two coverage levels \( l < u \) with the stipulation that customers that are further than \( u \) from the closest facility are not covered at all (have coverage weight of 0), customers with travel distance between \( l \) and \( u \) are partially covered (weight of \( 1/2 \)) and customers closer than \( l \) from the closest facility are fully covered (weight of 1). Other forms of the coverage decay function may include linear decay, exponential decay, step function (representing multiple coverage radii instead of a single one in the cover objective), etc. In fact, by using a linear cover decay function with lower radius \( l = 0 \) and the upper radius \( u \) equal to the maximum distance between any two nodes, we obtain the median objective (details are provided below). Thus, the gradual cover framework allows us to represent the median objective, the cover objective and the intermediate objectives with various degrees of sensitivity with respect to the absolute travel distance. It can be seen as the counterpart of the OMP where the sorting and weighting steps (1A) and (1B) above are replaced with the “coverage weight” step (1C): for each customer determine the coverage weight by applying the coverage decay function to the travel distance to the closest facility, followed by step (2'): maximize the sum of the coverage weights (instead of minimizing the total weighted distance as in step (2) above).

However, the gradual cover framework is not capable of representing objectives that depend on relative distances since it is missing the sorting step (1A). Thus, it cannot capture the center objective, the cent-dian objective or other objectives related to relative distances that are easily representable within the OMP framework.

The goal of the current paper is to define and analyze a new model, the Ordered Gradual Covering Location Problem (OGCLP), that combines the features of the OMP and GCLP models. This new model is defined by performing step (1) above, followed by steps (1C), (1A), (1B) and (2') – i.e., the sorting and weighting steps are inserted into the gradual cover framework. The resulting model provides a unifying structure with respect to a wide range of classical location objectives (including the ones described earlier) and is capable of capturing sensitivity with respect to both, the absolute and relative travel distances. This has practical implications since certain aspects of the underlying real-life problems – e.g., equity – are best represented in terms of relative travel distances, while others – e.g., definition of primary and secondary trading areas – are most naturally captured in terms of absolute distances. Hence, the focus and aim of the new framework is on the modeling aspect and not on enhanced solvability and decreased running times for the OMP or the GCLP. Apart from already existing problems, it easily allows us to also model completely new objectives which might be more suitable for a certain application than the current ones. Moreover, given a solution approach for the unified model we can solve all these specific problems using the same algorithm. By contrast, currently, with ever-so-slight a change in one of the known problems, one has to think of a new way to solve the modified problem. Hence, the power of the framework lies in its flexibility to easily adapt objectives to model an underlying problem just by changing some parameters and to solve it without the necessity for any additional (research or implementation) work. In addition, OGCLP is of theoretical importance since any results established for this model are directly applicable to the standard location objectives. In the following, we restrict our attention to network and discrete versions of the OGCLP, leaving the study of planar version to future research.

The plan for the paper is as follows. Necessary notations are introduced and gradual decay functions are discussed in Section 1.1. The OGCLP is formally defined in Section 1.2. The single-facility version of the OGCLP is considered in detail in Section 2. In particular, we derive Finite Dominating Sets (FDS's) for models with non-negative weights, special classes of non-negative weights that yield simpler FDS's (Section 2.1), and general weights (Section 2.2). Moreover, we present efficient algorithms for determining the FDS. We also discuss the “conditional case” where a certain number of facilities is already
assumed to exist and a new facility is to be added (Section 2.3). In Section 3, the FDS results are extended to multi-facility models. These results allow us to discretize the network model by determining a finite set of potential facility locations \textit{a priori}, which essentially converts the network location model into its discrete counterpart. The multi-facility discrete location problem is addressed in Section 4 where we discuss several Integer Programming formulations for the OGCLP. The computational experiments analyzing the performance of these formulations are presented in Section 4.1. Some concluding remarks are presented in Section 5.

1.1. Preliminaries

Let $N = (G, \ell)$ be a network with an underlying undirected graph $G = (V, E)$ and edge length $\ell, V = \{v_1, \ldots, v_n\}$ and $E = \{e_1, \ldots, e_m\}$ denote the set of nodes and edges, respectively, of the graph. An edge $e \in E$ is denoted $e = [v_i, v_j]$ with $i < j$. A point $x \in G$ on an edge $e = [v_i, v_j]$ of the network is denoted $x = (e, t), 0 \leq t \leq 1$, where $t$ is the relative distance of $x$ from node $v_i$. Let $w_i$ be the demand associated with node $v_i \in V$ (which can be interpreted as the number of customers at node $v_i$), and $d_i(x) = d(v_i, x)$ be the shortest distance between node $v_i$ and $x \in G$. For short, we often write $i \in V$ instead of $v_i \in V$.

Assume that we wish to locate $p \geq 1$ facilities and that the facilities can be located at nodes or along the edges of the network. Suppose, for the moment, that the locations have already been chosen and let $S = \{x_1, \ldots, x_p\} \subseteq G$, $|S| = p$ be the location set. Define $d_i(S) = \min_{x \in S} d_i(x)$.

Let $(l_i, u_i)$ be a pair of radii associated with node $v_i \in V$. Node $v_i$ is fully covered (not covered) if $d_i(S) \leq l_i$ ($d_i(S) > u_i$).

For $l_i < d_i(S) \leq u_i$, node $v_i$ is partially covered.

Let $f_i(t)$ be a non-increasing function for $t \in [l_i, u_i]$ with $f_i(l_i) = 1$ and $f_i(u_i) = 0$. The function $f_i$ is called the coverage decay function. The demand of node $v_i$ that is covered by $S$ is defined as

$$c_i(d_i(S)) = \begin{cases} w_i & \text{if } d_i(S) \leq l_i \\ u_i f_i(d_i(S)) & \text{if } l_i < d_i(S) \leq u_i \\ 0 & \text{if } u_i < d_i(S). \end{cases}$$

We call $c_i(d_i(S))$ the coverage function. For short, we will write $c_i(S)$ instead of $c_i(d_i(S))$.

\textbf{Remark.} Note that $c_i(S) = \max_{x \in S} c_i(x)$

The Gradual Cover Location Problems (GCLP) can now be stated as

$$\max_{S \subseteq G, |S| = p} \sum_{i \in V} c_i(S).$$

Some examples of GCLP models with different coverage decay functions are provided below:

(1) Linear decay function

$$f_i(t) = 1 - \frac{t}{\alpha}, \quad i \in V$$

where $\alpha = \max_{i \in V} d_i(f)$ is a constant. If $u_i = \alpha$ and $l_i = 0$ for all $i \in V$, we have

$$\sum_{i \in V} c_i(S) = \sum_{i \in V} w_i f_i(d_i(S)) = \sum_{i \in V} w_i - \frac{1}{\alpha} \sum_{i \in V} d_i(S) w_i.$$

Since the first term is a constant and the second term is a (constant multiple of the) standard $p$-median objective, the GCLP with this coverage function is equivalent to the $p$-median problem.

(2) Stepwise decay function

$$f_i(t) = \alpha_i^k \quad \text{if } t \in (r_i^{k-1}, r_i^k], \text{ } k = 1, \ldots, K_i$$

where $1 > \alpha_i^1 > \cdots > \alpha_i^{K_i} > 0$ and $l_i = r_i^0 < r_i^1 < \cdots < r_i^{K_i} = u_i$. For this type of coverage decay function, the values $\alpha_i^k$ are known as coverage levels and $r_i^k$ as coverage radii. Clearly, in case of a single coverage level and a single coverage radius, the problem becomes equivalent to the standard maximum cover problem.

(3) Piecewise linear decay function

$$f_i(t) = \beta_i^k - \alpha_i^k \cdot t \quad \text{if } t \in (r_i^{k-1}, r_i^k], \text{ } k = 1, \ldots, K_i$$

where $\beta_i^1, \ldots, \beta_i^{K_i}, \alpha_i^1, \ldots, \alpha_i^{K_i} > 0$ and $l_i = r_i^0 < r_i^1 < \cdots < r_i^{K_i} = u_i.$
1.2. OGCLP: Formulation and relationship to the ordered median problem

Let \(\sigma \in \mathcal{P}(1 \ldots n)\) be a permutation of the index set \(\{1 \ldots n\}\) that sorts the values of \(c_i(S)\) in non-decreasing order:

\[
c_{\sigma(1)}(S) \leq c_{\sigma(2)}(S) \leq \cdots \leq c_{\sigma(n)}(S).
\]

Let \(\lambda \in \mathbb{R}^n\) be a real-valued vector; \(\lambda\) is called the modeling vector and its entries are the modeling weights.

The Ordered Gradual Covering Location Problem (OGCLP) is defined as follows:

\[
\max \left\{ \sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S) \mid S \subseteq G, |S| = p \right\}.
\]

For \(S \subseteq G, |S| = p\), we call

\[
g(S) = \sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S)
\]

the Ordered Gradual Covering Function (OGCF).

Note that if \(c_i(S) = -w_{d_i}(S)\) for all \(i \in V\), the above formulation is exactly equivalent to the OMP on the network \(G\) [13]. However, this would require that the coverage decay function be \(f_i(t) = -t\), which does not satisfy the requirements of a valid coverage decay function (recall that \(f_i(t)\) has to be non-negative, equal to 1 at \(u_i\) and 0 at \(l_i\)). However, as the following result shows, with a suitable alteration of the network and the coverage decay function, the OMP can indeed be represented as a special case of OGCLP.

Let network \(G' = (V, E)\) be identical to \(G\) (i.e., have the same nodes, edges and the distance function), but equipped with a unitary node demand vector \(w' = (1, \ldots, 1)\) instead of the original vector \(w\). We have the following result:

**Theorem 1.1.** For any modeling vector of \(\lambda \in \mathbb{R}^n\) and any integer number of facilities \(p \geq 1\), the OMP on network \(G\) is equivalent to the OGCLP on network \(G'\).

**Proof.** For \(i \in V\), let \(j(i) = \arg \max_{j \in V} d_i(j)\) and let \(i^* = \arg \max_{k \in V} w_k d_k(j(k))\). Define the coverage decay function

\[
f_i(t) = 1 - \frac{w_i}{\alpha'} t, \quad i \in V
\]

where, \(\alpha' = w_{i^*} d_{\sigma(i^*)} - w_i\) and \(l_{j(i)} = 0\) for all \(i \in V\). Intuitively, the original node weight vector has been incorporated into the coverage decay function. First note that \(f_i(t)\) defined above is a proper coverage decay function since \(f_i(u_i) = 0, f_i(l_i) = f_i(0) = 1\) and for any \(t \in [l_i, u_i]\), \(f_i(t) \geq 0\) must hold since \(t \leq u_i\) implies that \(t w_i < \alpha'\).

Let \(S \subseteq G, |S| = p\) be a location set. For any \(i \in V, d_i(S) \leq d_i(j(i)) \leq (1/w_i) \max_{k \in V} w_k d_k(j(k)) = u_i\). Thus \(d_i(S) \in [l_i, u_i]\) and

\[
c_i(S) = 1 - \frac{w_i d_i(S)}{\alpha'}, \quad i \in V.
\]

Therefore for \(i, j \in V\),

\[
c_i(S) \leq c_j(S) \Leftrightarrow w_j d_j(S) \geq w_i d_i(S).
\]

Let \(\sigma'\) be a permutation defined by (1) with respect to the network \(G'\) and \(\sigma\) be the OMP-defining permutation on the network \(G\) (i.e., \(w_{\sigma(1)} d_{\sigma(1)}(S) \leq \cdots \leq w_{\sigma(n)} d_{\sigma(n)}(S)\)). By (3), \(\sigma'(k) = \sigma(n-k+1)\) must hold for all \(k \in \{1, \ldots, n\}\). Define \(\lambda_k' := \lambda_{n-k+1}\). Thus

\[
\lambda_k' c_{\sigma'(k)}(S) = \sum_{k=1}^{n} \lambda_{n-k+1} c_{\sigma(n-k+1)}(S)
\]

\[
= \sum_{k=1}^{n} \lambda_k c_{\sigma(k)}(S) = \sum_{k=1}^{n} \lambda_k - \frac{1}{\alpha'} \sum_{k=1}^{n} \lambda_k w_{\sigma(k)} d_{\sigma(k)}(S).
\]

Since the first term above is constant and the second term is a scalar multiple of the OMP objective function, maximizing the OGCLP objective is equivalent to minimizing the OMP objective. \(\square\)

The preceding result shows that any problem that can be represented as an OMP (i.e., \(p\)-median, \(p\)-center, cent-dian, etc.) can be solved via the OGCLP, proving that OGCLP indeed provides the unifying framework for all classical objectives in location models. OGCLP with different modeling weight vectors may also be interesting in its own right, as discussed in the following examples.
Examples of OGCLP’s with different modeling vectors \( \lambda \)

**Median:** \( \lambda = (1, \ldots, 1) \)

In this case we have

\[
\sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S) = \sum_{i=1}^{n} c_i(S)
\]

that is, the problem reduces to the standard GCLP. As discussed earlier, both the median and the maximum cover location problems are special cases of the GCLP.

**k-Centra:** \( \lambda^k = (0, \frac{n-k}{k}, 0, 1, 0, \ldots, 0) \)

We have

\[
\sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S) = \lambda_{n-k+1} c_{\sigma(n-k+1)}(S).
\]

This can be viewed as an extension of the center objective to the gradual cover setting. The center objective calls for maximizing the coverage of the worst-covered customer node, which can be achieved by setting \( \lambda^n = (1, 0, \ldots, 0) \). However, in a “true” cover setting, we would normally not expect to be able to extend the coverage to all customer nodes—thus the coverage level of some nodes would be 0; rendering the modeling vector \( \lambda^n \) not very useful.

One of the standard motivations for maximum cover problems is that the coverage of the worst-covered nodes will be sub-contracted to another service provider. The \( k \)-centra objective calls for maximizing the coverage of the worst-covered node among the \( k \) nodes receiving the best coverage. Thus, if the intention is to sub-contract the \( n - k \geq 0 \) worst-covered nodes to another provider, then the objective above maximizes the coverage for the worst-covered node which will be served from the facility set \( S \).

**k-Cover:** \( \lambda = (0, \frac{n-k}{k}, 0, 1, \ldots, 1) \)

In this case we have

\[
\sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S) = \max_{V'^{\subseteq} V} \sum_{i \in V'} c_i(S),
\]

i.e., we concentrate on providing the best possible coverage to the \( k \) best-covered nodes (again, under the assumption that the \( n - k \) worst-covered nodes will be sub-contracted to another service provider).

**k-Centdiain-cover:** \( \lambda = (0, \frac{n-k}{k}, 0, 1, \alpha, \frac{k-\alpha}{k}, \alpha), \alpha \in (0, 1) \)

In this case we have

\[
\sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S) = \alpha \max_{V'^{\subseteq} V} \sum_{i \in V'} c_i(S) + (1 - \alpha) c_{\sigma(k)}(S),
\]

i.e., a convex combination of the \( k \)-cover and the \( k \)-centra objectives, allowing us to put extra weight on the \( k \)-th worst-covered customer nodes in the gradual \( k \)-cover setting.

**Trimmed-Cover:** \( \lambda = (0, \frac{k_1}{k_1}, 0, 1, \ldots, 1, 0, \frac{k_2}{k_2}, 0) \)

In this case we have

\[
\sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S) = \sum_{i=k_1+1}^{n-k_2} c_{\sigma(i)}(S)
\]

which is the so-called \((k_1, k_2)\)-trimmed mean, leaving aside the \( k_1 \) worst and \( k_2 \) best covered nodes. This is an alternative to the \( k \)-centra objective which may be useful when it has been decided that the service of the \( k_1 \) worst-covered nodes will be subcontracted to another provider, and the \( k_2 \) best-covered nodes are excluded from the calculation on the grounds that they will automatically receive adequate coverage.

**Anti-Cover:** \( \lambda = (-1, \ldots, -1) \)

In this case we have

\[
\sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S) = - \sum_{i=1}^{n} c_i(S)
\]

that is, we minimize the coverage of the clients. The possible applications include the location of obnoxious facilities, e.g., waste disposals. All clients within a given distance \( l \) of a facility experience the full impact of the
obnoxious effect of the facility, the clients further than \( u \) away experience a negligible effect, with the impact of the facility gradually decreasing between these two radii.

**Equity Objective:** \( \lambda = (1, \tfrac{k}{5}, 1, 0, \ldots, 0, -1, \tfrac{k}{5}, -1) \)

In this case we have

\[
\sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S) := - \left( \max_{V' \subseteq V} \left\{ \sum_{i \in V'} c_i(S) - \min_{V' \subseteq V} \left\{ \sum_{i \in V'} c_i(S) \right\} \right\} \right)
\]

that is, we are trying to minimize the difference in total coverage received by the \( k \) best-covered and \( k \) worst-covered clients. This objective may be useful in location of non-emergency public service facilities, such as schools, where equity considerations are important.

2. **Single facility**

For the ease of understanding we will first discuss the single facility case, i.e., \( S = \{x\} \). Then

\[
c_i(x) = \begin{cases} 
  w_i & \text{if } d_i(x) \leq l_i \\
  w_i f_i(d_i(x)) & \text{if } l_i < d_i(x) \leq u_i \\
  0 & \text{if } u_i < d_i(x).
\end{cases}
\]

The Ordered Gradual Decay Function

\[
g(x) = \sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(x)
\]

is defined point-wise and globally is neither linear nor convex nor concave. Moreover, as the new facility can be located everywhere on the network, there are infinitely many points to consider. Therefore, it is not possible to solve the problem exactly with a general purpose solver. To that end, in the following we will determine parts of the network where the objective is linear or at least convex/concave. This allows us to identify points which are candidates for an optimal solution and to discard others, which can never be optimal. This analysis leads to a finite set of points, called the finite dominating set (FDS), where we can always find an optimal solution. Afterward, we will present an efficient algorithm which actually determines an optimal solution.

Whenever two coverage functions \( c_i \) and \( c_j \) intersect, the permutation of coverage functions changes and therefore so does the representation of the objective function.

Analogously to [3], we define the following sets. For all \( i \in V \), let

\[
L_i = \{ x \in G \mid d_i(x) = l_i \} \text{ and } U_i = \{ x \in G \mid d_i(x) = u_i \}.
\]

and

\[
L = \bigcup_{i \in V} L_i \text{ and } U = \bigcup_{i \in V} U_i.
\]

Note that \( L \) and \( U \) have \( O(mn) \) elements each, as the distance function \( d_i(x) \) can attain the values \( l_i \) and \( u_i \) at most twice on each edge. (To see that, consider a star graph with one central node \( v_1 \) and \( n-1 \) outlying nodes \( v_2, \ldots, v_n \), identical node weights, and edge lengths of one; thus, \( m = n-1 \); for \( l_i = 1.5, i = 2, \ldots, n \), we have \( |L_i| = m-1 \) as \( d_i([v_1, v_j], 0.5)) = 1.5 \) for \( j = 2, \ldots, n, i \neq j \); for \( l_i = 0.5, |L_i| = m \); hence \( |L_I| = (n-1)(m-1) + m = O(mn) \).

Next, we will show that if \( f_i \) is convex and continuous for all \( i \in V \), the intersection points of coverage functions, together with the node set \( V \) and the set \( L \) comprise a finite dominating set (FDS) for the single-facility problem.

**Definition 2.1 (Equilibrium and Bottleneck Points).** Let \( v_i, v_j \in V, v_i \neq v_j \), and \( w_i \cdot w_j \neq 0 \). Define

\[
EQ^i_j := \{ x \in G : c_i(x) = c_j(x) \}.
\]

For \( v_i = v_j \) or \( w_i \cdot w_j = 0 \), we set \( EQ^i_j := \emptyset \). Let \( EQ_{ij} \) be the boundary of \( EQ^i_j \) and let \( EQ := \bigcup_{i,j} EQ_{ij} \). The points in \( EQ \) are called equilibrium points.

A point \( x = (e, t) \) on an edge \( e = [v_i, v_j] \in E \) is called a bottleneck point if there exists some node \( v_k \) with \( w_k \neq 0 \), such that \( d(x, v_k) = d(x, v_i) + d(v_i, v_k) = d(x, v_j) + d(v_j, v_k) \). Denote by \( BN \) the set of bottleneck points on \( N \).

Note that the set of bottleneck points \( BN \) has \( O(mn) \) elements. The number of equilibrium points on an edge \( e \in E \) depends on the characteristics of the functions \( f_i \), as we will show in the following example.
**Example 2.1.** (1) Assume the coverage decay functions \( f_i(t) \) are linear for all \( i \in V \).

Let \( v_i \in V \) and \( e \in E \). If \( d_i(x) \) does not have a bottleneck point on \( e \), then \( d_i(x) \) is linear on \( e \). Therefore, \( c_j(x) \) has at most two breakpoints on \( e \) (for \( d_i(x) = l_i \) and \( d_i(x) = u_i \)). If \( d_i(x) \) has a bottleneck point \( y \in e \), then \( d_i(x) \) is linear left and right of \( y \) and hence \( c_j(x) \) has at most four breakpoints on \( e \). See left-hand side picture in Fig. 1. (Note that \( c_j(x) \) has breakpoints either at \( d_i(x) = u_i \) or at \( y \) but not both.)

As the two coverage functions \( c_j(x) \) and \( c_j(x) \) are linear between consecutive breakpoints of \( c_i \) and \( c_j \) on \( e \), they intersect in at most six points. (We can ignore situations where both are constant.) Hence, there are \( O(n^2) \) equilibria on an edge and thus \( EQ \) has \( O(mn^2) \) elements.

Consider the undirected network and the lower and upper bounds \( l_i \) and \( u_i \) depicted in Fig. 2. Fig. 3 shows the distance functions \( d_i(x) \) (left hand side) and the coverage functions (right hand side) on the edge \( e = [v_4, v_5] \). The point \( x = ([v_4, v_5], 0.33) \) is an equilibrium of the coverage functions \( c_3(x) \) and \( c_5(x) \), i.e., \( x = EQ_{35} \).

(2) Assume the coverage decay functions \( f_i(t) \) are piecewise linear with \( K_i \) breakpoints for all \( i \in V \).

If \( d_i(x) \) is linear (has a breakpoint) on \( e \), \( c_j(x) \) has at most \( K_i + 1(2K_i + 1) \) breakpoints on \( e \). Therefore, \( c_j(x) \) and \( c_j(x) \) intersect in at most \( O(K_i + K_j) \) points. Hence, \( |EQ| = O(Kmn^2) \), where \( K = \max_{i \in V} K_i \).

Observe that if \( f_i \) is convex, then also

\[
\bar{f}_i(t) = \begin{cases} f_i(t) & \text{if } l_i < t \leq u_i \\ 0 & \text{if } t > u_i \end{cases}
\]

is convex for \( t > l_i \), since \( f_i(u_i) = 0 \). This leads to the following result.

**Lemma 2.1.** Let \( f_i \) be convex and continuous for all \( v_i \in V \), \( S \subseteq G \), and \( z_1, z_2 \in e \), \( e \in E \), be two consecutive elements of \( V \cup L \) on edge \( e \), i.e., \( [z_1, z_2] \cap (V \cup L) = \{z_1, z_2\} \). Then, \( c(S \cup \{x\}) \) is convex for \( x \in [z_1, z_2] \).
Proof. We have that \( d_i(S \cup \{x\}) \) is concave for \( x \in [z_1, z_2] \), since \( d_i(x) \) is concave on an edge, \( d_i(S) \) is constant with respect to \( x \), and \( d_i(S \cup \{x\}) = \min\{d_i(S), d_i(x)\} \). Moreover, we have that \( c_i(S \cup \{x\}) = \max\{c_i(S), c_i(x)\} \) with \( c_i(S) \) being constant with respect to \( x \).

By assumption on \( z_1 \) and \( z_2 \), either \( d_i(x) \leq l_i \) or \( d_i(x) > l_i \) for all \( x \in [z_1, z_2] \). In the former case, \( c_i(x) = w_i \) is constant.

In the latter case, \( c_i(x) = w_i f_i(d_i(x)) \) is a composition of a concave and a convex non-increasing function and thus convex. Hence, \( c_i(S \cup \{x\}) \) is convex as a maximum of convex functions. \( \square \)

If all \( f_i \) are continuous, the permutation of the coverage functions can only change at equilibrium points. This observation leads to the following theorem.

**Theorem 2.1.** Let \( N \) be an undirected network, \( w, \lambda \geq 0 \), and \( f_i \) be convex and continuous for all \( i \in V \). Then, \( V \cup EQ \cup L \) is a finite dominating set.

Proof. Augment \( G \) by inserting the elements of \( EQ \cup L \) as new nodes with weight zero. Denote the augmented graph \( G' = (V', E') \).

Let \( e \in E' \) be an edge of the augmented graph. As we added the equilibrium points to the finite dominating set and all coverage decay functions are continuous, the order of the functions \( \{c_i(x)\}_{i=1}^n \) will not change on the edge. Thus, the objective function reduces to a weighted sum of the coverage functions

\[
g(x) = \sum_{i=1}^n \lambda_i c_{i_{\sigma_e(i)}}(x)
\]

where \( \sigma_e \) denotes the corresponding permutation of the coverage functions on \( e \). As the coverage functions are convex (Lemma 2.1) and \( \lambda \geq 0 \), \( g(x) \) is also convex as a weighted sum of convex functions. Therefore, the objective function attains its maximum at one of end nodes of \( e \) and the result follows. \( \square \)

As \( |L| = O(mn) \), the size of the finite dominating set is of order \( O(mn + |EQ|) \), where \( |EQ| \) depends on the actual representations of the decay functions, see Example 2.1.

If the decay functions \( f_i \) are not convex, the above result no longer holds. However, we can extend it to stepwise as well as piecewise linear decay functions as follows. Let \( r_i^k, k = 0, \ldots, K_i \), with \( l_i = r_i^0 < r_i^1 < \cdots < r_i^K_i = u_i \) denote the breakpoints of the decay functions \( f_i \), see also Section 1.1. Define \( R_i = \{r_i^k \mid k = 0, \ldots, K_i\} \). Each breakpoint of \( f_i \) induces one or more breakpoints of the coverage functions \( c_i \). Denote by

\[
BP_i = \{x \in G \mid d_i(x) = r \in R_i\}
\]

the set of breakpoints of the coverage functions \( c_i \) induced by the breakpoints of the decay functions. Define \( BP = \bigcup_{i \in V} BP_i \) as the set of all these breakpoints. Note that \( L, U \subset BP \).

**Theorem 2.2.** Let \( N \) be an undirected network and \( w, \lambda \geq 0 \).

1. For piecewise linear decay functions, \( V \cup BP \cup EQ \) is a finite dominating set.
2. For stepwise decay functions, \( V \cup BP \) is a finite dominating set.

Proof. We use the same ideas as in the proof of Theorem 2.1.

1. For piecewise linear decay functions, \( V \cup BP \cup EQ \) as new nodes with weight zero. As a result, the decay functions \( f_i \) reduce to linear functions on each edge of the augmented graph. Hence, the coverage functions \( c_i \) are convex on each edge as a composition of a concave and a linear non-increasing function.

Moreover, as the order of the functions \( \{c_i(x)\}_{i=1}^n \) will not change on an edge, the objective function reduces to a weighted sum of convex functions. Therefore, \( g(x) \) attains its maximum at one of the end nodes of the edges of the augmented graph.

2. For stepwise functions, augment \( G \) by inserting the elements of \( BP \) as new nodes with weight zero. Then, the coverage functions \( c_i \) are constant on the interior of each edge of the augmented graph. As the order of the functions \( \{c_i(x)\}_{i=1}^n \) will not change on the interior, the objective function will also be constant. As \( w, \lambda \geq 0 \), \( g(x) \) will always attain its maximum at least at one of the end nodes of the edges of the augmented graph.

Hence, the result follows. \( \square \)

Concerning the size of the finite dominating sets, observe that \( |BP| = O(mK_i) \) as each breakpoint \( r \in R_i \) may induce a breakpoint of a coverage function on each edge. Hence, \( |BP| = O(mnK) \), where \( K = \max_{i \in V} K_i \). Moreover, we have \( |EQ| = O(mn^2K) \). Thus, the size of the finite dominating sets is \( O(mn^2K) \) and \( O(mnK) \), respectively.

Unfortunately, it does not appear possible to characterize FDS for the case of general coverage decay functions and unrestricted modeling vectors \( \lambda \). However, we will see in the following sections that FDS can be characterized for many important special cases.
Efficient algorithms

After identifying finite dominating sets, we now discuss how to efficiently compute them and, subsequently, solve the corresponding problems. For the sake of brevity, we restrict ourselves to stepwise and piecewise linear decay functions.

We start by considering the case of piecewise linear decay functions. To solve the problem, we have to determine the sets \( BP \) and \( EQ \). \( BP \) can be computed in \( O(mnK) \) time, as a distance function \( d_i \) consists of at most two linear pieces on an edge, and we can solve the equations \( d_i(x) = r^i_k, k = 0, \ldots, K_i, \) in \( O(K) \) time. Now we turn to \( EQ \). For \( i, j \in V \) and \( k \in E \) we can compute the intersection points of the coverage functions \( c_i \) and \( c_j \) on edge \( e_k \) in \( O(K_i + K_j) \) time, since \( c_i(c_j) \) has at most \( O(K_i) \) \((O(K_j)) \) breakpoints on \( e_k \). To determine the set \( EQ \) we have to intersect pairwise all coverage functions on all edges, which requires in total \( O(mn^2K) \) time.

To find the optimal solution, we evaluate the objective function for all elements of the finite dominating set. For a point \( x \in G \), we can compute the objective function value \( g(x) \) in \( O(n \log n) \) time. Therefore, the overall complexity for solving the problem is

\[
O(mnK + mn^2K + (n + mnK + mn^2K)n \log n) = O(mn^3K \log n).
\]

For stepwise decay functions this reduces to \( O(mn^2K \log n) \).

However, for piecewise linear decay functions that are continuous we can adapt the efficient algorithm for the single-facility Ordered Median Problem of [10] to obtain a lower complexity algorithm. Observe, that the only breakpoints of the coverage functions \( c_i(x) \) occur at elements of \( BP_i \) or at bottleneck points of the distance functions \( d_i(x) \).

For an edge \( e \in E \), we first compute the set of bottleneck points, equilibria, and elements of \( BP \) on \( e \). Afterward, we sort them in nondecreasing distance from one of the end nodes. Denote the elements of the sorted sequence by \( \{x_1, \ldots, x_q\} \). Then, for any \( x \) in the interior of the subedge connecting two consecutive elements \( x_q \) and \( x_{q+1} \), the sorting of the coverage functions \( c_i \) does not change; moreover, the \( c_i \) are linear on the subedge \( x_q = [x_q, x_{q+1}] \) (as we included the bottleneck and breakpoints) and hence also the objective function. Therefore, if we know the objective value at \( x_q \) and the slope of the objective function on \( x_q \), we can obtain \( g(x_{q+1}) \) in constant time. Moreover, we can update the slope of \( g \) at each \( x_q \) in constant time (as the \( f_i \) are continuous, see [8] for more details). Only for the first element, \( x_1 \), do we have to explicitly compute the objective function value \( g(x_1) \) and the slope of \( g(x) \) on \( s_1 \) in \( O(n \log n) \) time. Therefore, the overall complexity of the algorithm is \( O(mn^2K \log(nK)) \), as we have \( O(n^2K) \) bottleneck points, equilibria, and elements of \( BP \) on an edge.

2.1. Special cases

For certain modeling vectors \( \lambda \) we can obtain smaller finite dominating sets, as the following result shows.

**Corollary 2.1.** Let \( N \) be an undirected network, \( w, \lambda \geq 0 \), and \( f_i \) be convex for all \( i \in V \). Moreover, let the modeling weights be non-decreasing, i.e., \( \lambda_1 \leq \cdots \leq \lambda_n \). Then, \( V \cup L \) is a finite dominating set.

**Proof.** Augment \( G \) by inserting the elements of \( L \) as new nodes with weight zero. Denote the augmented graph \( G^* = (V', E') \). From the definition of \( L \) it follows that all coverage functions \( c_i(x), i \in V \), are convex on each edge of the augmented graph.

For \( \lambda_1 \leq \cdots \leq \lambda_n \), we have

\[
g(x) = \sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(x) = \max_{\pi \in P(1 \ldots n)} \sum_{i=1}^{n} \lambda_i c_{\pi(i)}(x)
\]

as the permutation \( \sigma \in P(1 \ldots n) \) that sorts the coverage functions \( c_i \) in nondecreasing order for a given \( x \in e', e' \in E' \), is identical to the permutation \( \pi^* \) for which the maximum on the right-hand side is obtained [7]. As the coverage functions \( c_{\pi(i)}(x) \) are convex on each edge of \( G^* \), so is the right-hand side expression as a maximum of a weighted sum of convex functions. Therefore, \( g(x) \) obtains its maximum at an end node of \( e' \) and the result follows.

For this special case, the size of the finite dominating set reduces to \( O(mn) \).

2.2. Real-valued modeling weights: The case of (semi-) obnoxious facilities

First, we consider problems where the modeling weights are strictly non-positive, i.e., \( \lambda \leq 0 \) (alternatively, we could assume the node weights to be non-positive). Intuitively, this corresponds to the case where customer benefit is maximized when their coverage is as low as possible — which occurs in case of facilities like garbage dumps or nuclear waste sites, i.e., facilities one would rather not be covered by. Such facilities are typically referred to as “obnoxious facilities” in the location literature (e.g., [6,5]). Since \( \lambda \leq 0 \), we have

\[
\max_{x \in G} \sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(x) = -\min_{x \in G} \sum_{i=1}^{n} |\lambda_i| c_{\sigma(i)}(x).
\]
Observe that if \( f_i \) is concave, then the function

\[
\tilde{f}_i(t) = \begin{cases} 
  u_i & \text{if } d_i(S) \leq l_i \\
  f_i(t) & \text{if } l_i < d_i(S) \leq u_i
\end{cases}
\]

is concave for \( t \leq u_i \), since \( f(l_i) = w_i \).

**Lemma 2.2.** Let \( f_i \) be concave for all \( i \in V \) and \( z_1, z_2 \) be two consecutive elements of the set \( V \cup BN \cup U \) on an edge, i.e., \( [z_1, z_2] \cap (V \cup BN \cup U) = [z_1, z_2] \). Then, \( c_i(x) \) is concave for \( x \in [z_1, z_2] \).

**Proof.** As we included the bottleneck points, \( d_i(x) \) is linear for \( x \in [z_1, z_2] \). By assumption on \( z_1 \) and \( z_2 \), either \( d_i(x) \leq u_i \) or \( d_i(x) > u_i \) for all \( x \in [z_1, z_2] \). In the latter case, \( c_i(x) \) is constant. In the former case, \( c_i(x) \) is a composition of a linear and a concave non-increasing function and therefore concave. \( \Box \)

Now we can state a result analogous to **Theorem 2.1**.

**Theorem 2.3.** Let \( N \) be an undirected network, \( \lambda \leq 0 \leq w \), and \( f_i \) be concave and continuous for all \( i \in V \). Then, \( V \cup BN \cup EQ \cup U \) is a finite dominating set.

**Proof.** Augment \( G \) by inserting the elements of \( BN \cup EQ \cup U \) as new nodes with weight zero. Denote the augmented graph \( G' = (V', E') \).

Let \( e \in E' \) be an edge of the augmented graph. As we added the equilibrium points to the finite dominating set and all coverage decay functions are continuous, the order of the functions \( \{c_i(x)\}_{i=1}^n \) will not change on the edge. Thus, the objective function reduces to a weighted sum of the coverage functions

\[
g(x) = \sum_{i=1}^n \lambda_i c_{\sigma_e(i)}(x)
\]

where \( \sigma_e \) denotes the corresponding permutation of the coverage functions on \( e \). As the coverage functions \( c_{\sigma_e(i)} \) are concave (Lemma 2.2) and \( \lambda \leq 0 \), \( g(x) \) is convex as a negatively weighted sum of concave functions. Therefore, the objective function attains its maximum at one of the end nodes of the edge and the result follows. \( \Box \)

As \( |U|, |BN| = O(nm) \), the size of the finite dominating set is of order \( O(nm + |EQ|) \), where \( |EQ| \) depends on the actual representations of the decay functions.

Now we turn to the problem with semi-obnoxious facilities where we allow the modeling weights to be real-valued. Unfortunately, the result of **Theorem 2.3** does not carry over to this case. The problem is that for a mixture of negative and positive \( \lambda_i \), the functions \( \lambda_i c_{\sigma_e(i)}(x) \) we are summing up for the objective function are convex for negative lambdas and concave for \( \lambda_i > 0 \). Hence, their sum will not necessarily be concave or convex.

However, we can extend **Theorem 2.2** for stepwise and piecewise linear decay functions to real-valued node and modeling weights. But before, we need the following definition. For two adjacent elements \( x, y \) of the set \( V \cup BP \) on an edge, we denote by \( mp(x, y) \) the midpoint between the two elements and by \( MP \) the set of all midpoints between adjacent elements.

**Theorem 2.4.** Let \( N \) be an undirected network and \( w, \lambda \in \mathbb{R}^n \).

1. For piecewise linear decay functions, \( V \cup BN \cup BP \cup EQ \) is a finite dominating set.
2. For stepwise decay functions, \( V \cup BP \cup MP \) is a finite dominating set.

**Proof.** The proof is similar to the one of **Theorem 2.2**.

1. By inserting the bottleneck points in addition to the elements of \( EQ \cup BP \), the distance functions are linear on each edge of the augmented graph. Hence, the coverage functions are also linear on each edge as a composition of two linear functions \( d_i \) and \( f_i \). Therefore, \( g(x) \) again attains its maximum at one of the end nodes of the edges of the augmented graph.
2. As for the nonnegative case, the objective function is constant in the interior of an edge. However, now a point in the interior can have a strictly larger objective value than one of the end nodes.

Hence, the result follows. \( \Box \)

The size of the finite dominating sets is again \( O(mn^2K) \) and \( O(mnK) \), respectively, as \( |BN| = O(mn) \) and \( |MP| = O(mnK) \). Moreover, also the complexity for solving the problems is the same as in the nonnegative case.
2.3. The conditional location case

Assume that there are already some pre-existing facilities in operation. Denote \( C \in G \) the set of points where these facilities are sited. The task is to optimally locate an additional facility on the network.

Define the conditional distance function \( d_i(x) \)

\[
d_i^*(x) = d_i(C \cup \{x\}).
\]

The functions \( d_i^* \) are piecewise linear and concave with at most two breakpoints. Therefore, the results for the unconditional problem with nonnegative modeling vectors carry over to the conditional location case.

Now consider the conditional obnoxious case, i.e., \( \lambda \leq 0 \). For the obnoxious unconditional problem we had to add the bottleneck points to the FDS in order to have the distance functions being linear between consecutive elements of the FDS. Now, the conditional distance functions have breakpoints at bottleneck points or at points where

\[
d_i(x) = d_i(C).
\]

Formally, these points are defined as follows.

**Definition 2.2 (Conditional Eextreme Points).** Let \( v_i \in V \) and \( e \in E \). A point \( x \in G \) is called a conditional extreme point of node \( v_i \), if \( d_i(x) = d_i(C) \). Let \( CEP \) denote the set of all conditional extreme points of node \( v_i \) and let \( CEP = \bigcup_{i=1}^{n} CEP \) be the set of all conditional extreme points of nodes.

Note that \( CEP \) has \( O(mn) \) elements. Now we can prove analogous results to the unconditional obnoxious problem. As the proofs are nearly identical to the previous ones, we omit them.

**Lemma 2.3.** Let \( f_i \) be concave for all \( i \in V \) and \( z_1, z_2 \) be two consecutive elements of the set \( V \cup BN \cup CEP \cup U \) on an edge, i.e., \( [z_1, z_2] \cap (V \cup BN \cup CEP \cup U) = \{z_1, z_2\} \). Then, \( c_i(x) \) is concave for \( x \in [z_1, z_2] \).

**Theorem 2.5.** Let \( N \) be an undirected network, \( \lambda \leq 0 \leq w \), and \( f_i \) be concave and continuous for all \( i \in V \). Then, \( V \cup BN \cup CEP \cup EQ \cup U \) is a finite dominating set.

**Theorem 2.6.** Let \( N \) be an undirected network, \( w, \lambda \in \mathbb{R}^n \), and \( f_i \) be continuous piecewise linear (stepwise) functions. Then, \( V \cup BN \cup CEP \cup EQ \cup BP(V \cup CEP \cup BP \cup MP) \) is a finite dominating set.

The finite dominating sets and the algorithms have the same cardinality and complexity, respectively, as the ones for the unconditional problem.

3. Multi-Facility

Let now \(|S| = p > 1\) and \( w, \lambda \geq 0 \). First, we note that this problem is NP-hard (since it reduces to the gradual covering decay problem of Berman et al. [3] for \( \lambda = (1, \ldots, 1) \)).

Before turning to the general problem, we first discuss a special case, namely, the problem with non-decreasing modeling weights, i.e., \( \lambda_1 \leq \cdots \leq \lambda_p \). Here, we can prove a result analogous to the one in Berman et al. [3].

**Theorem 3.1.** Let \( N \) be an undirected network, \( \lambda \geq 0 \), and \( f_i \) be convex for all \( i \in V \). Moreover, let the modeling weights be non-decreasing, i.e., \( \lambda_1 \leq \cdots \leq \lambda_p \). Then, \( V \cup L \) is a finite dominating set.

**Proof.** Augment \( G \) by inserting the elements of \( L \) as new nodes. Denote the augmented graph \( G' = (V', E') \). From Lemma 2.1 we know that \( c_i(S \cup \{x\}), S \subseteq G, \) is convex for \( x \in e', e' \in E', \) i.e., on each edge of the multi-augmented graph \( G' \). Moreover, for \( \lambda_1 \leq \cdots \leq \lambda_p \), we have

\[
g(S \cup \{x\}) = \sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S \cup \{x\}) = \max_{\sigma \in P(1 \ldots n)} \sum_{i=1}^{n} \lambda_i c_{\sigma(i)}(S \cup \{x\})
\]

as the permutation \( \sigma \in P(1 \ldots n) \) that sorts the coverage functions \( c_i \) in non-decreasing order for a given \( x \in e' \) is identical to the permutation \( \pi^* \) for which the maximum on the right-hand side is obtained (e.g., Hardy et al. [7]). As the coverage functions \( c_{\pi(i)}(S \cup \{x\}) \) are convex on each edge of the augmented graph, so is the right-hand side expression as a maximum of a weighted sum of convex functions.

Suppose \( S^* \) is an optimal set of locations, and there exists \( s^* \in S^* \setminus \{V \cup L\} \). Let \( s^* \in e^*, e^* \in E' \). Replacing \( s^* \) by one of the end nodes of \( e^* \) will not decrease the objective function value. Thus, the result follows. \( \square \)

Now, we turn to the general case. Unfortunately, the finite dominating set for the single facility problem, \( V \cup L \cup EQ \), is no longer valid for \( p > 1 \), as the following counter example shows.
Example 3.1. Consider the tree network in Fig. 4 with \( l_i = 0, i \in \{1, \ldots, 4, 6\} \), \( l_5 = 1 \), and \( u_i = 3, i = 1, 4, 6, u_2 = 6, u_3 = 5, u_5 = 7 \). Let \( p = 2, \lambda = (1, 0.2, \ldots, 0.2) \) and the coverage decay functions be linear. Fig. 5 shows the coverage functions on the edges \([v_1, v_2]\) and \([v_4, v_6]\). As in an optimal solution nodes \( v_1, v_2, \) and \( v_3 \) will always be covered by a solution point on the “left-hand side” of the tree and the other nodes by a point on the “right-hand side”, the respective coverage functions are omitted.

We will show that \( V \cup L \cup EQ \) is no longer a finite dominating set for the multifacility problem. If we restrict \( X_2 \) to be a subset of \( V \cup L \cup EQ \), the optimal solution is given by

\[
S = \left\{ EQ_{13} = ([v_1, v_2], \frac{1}{2}), EQ_{46} = ([v_4, v_6], \frac{1}{2}) \right\},
\]

with objective value \( g(S) = 0.883 \). However, if we drop this restriction we obtain a slightly better solution, namely

\[
S^* = \left\{ x^* = ([v_1, v_2], \frac{2}{3}), EQ_{46} = ([v_4, v_6], \frac{1}{2}) \right\},
\]

with an optimal objective function value of 0.886. Note that \( x^* \) is not a node, or an element of the set \( L \), or an equilibrium point.

Fortunately, for special classes of modeling vectors we can identify finite dominating sets.

3.1. A finite dominating set for a special class of modeling vectors

For the ordered median location problem, for modeling vectors \( \lambda \) with

\[
\lambda_1 = \cdots = \lambda_b < \lambda_{b+1} = \cdots = \lambda_n, \quad b \in \{1, \ldots, n-1\}.
\]

Kalcsic et al. [9] provide a finite dominating set. This result was later extended by Kalcsics [8] to modeling vectors \( \lambda \) with

\[
\lambda_1 \geq \cdots \geq \lambda_b, \quad \lambda_{b+1} \geq \cdots \geq \lambda_n \quad \text{and} \quad \lambda_b < \lambda_{b+1}.
\]  \hfill (4)

As the only requirement for the proofs in [9] as well as [8] was that the distance functions are concave and continuous, we can prove an analogous result for the gradual covering problem with convex and continuous coverage functions. Here, however, with modeling vectors \( \lambda \) of the form

\[
\lambda_1 \leq \cdots \leq \lambda_b, \quad \lambda_{b+1} \leq \cdots \leq \lambda_n \quad \text{and} \quad \lambda_b > \lambda_{b+1}.
\]  \hfill (5)

We denote by \( \Lambda^b \) the set of all modeling vectors that fulfill (5). Note that the modeling vectors \( \lambda \) for the median, \( k \)-Cent-cover, \( k \)-cover, and trimmed-cover problems belong to \( \Lambda^b \).
In Theorem 3.1 we could observe that the problem is easy to solve if all modeling weights are non-decreasing. The modeling vectors $\lambda \in \Lambda^b$ have a similar representation. The only difference is, that there exist two consecutive modeling weights $\lambda_b$ and $\lambda_{b+1}$ where this property does not hold. The approach to identify a finite dominating set is to consider situations where the order of the coverage functions changes at the positions $b$ and $b+1$, i.e., where $c_{\tau(b)}(S) = c_{\tau(b+1)}(S)$. Analyzing what happens in these situations will lead to the desired FDS.

From now on, we assume that $N = (G, \ell)$ is an undirected network, $p \geq 2$, $w, \lambda \geq 0$, and $\lambda \in \Lambda^b$. Moreover, let $f_i$ be continuous and convex for all $i \in V$. The following discretization result is equivalent to the one of Kalcsics et al. [9] for the $p$-facility Ordered Median Problem (pOMP) with lambda vectors fulfilling (4). The only difference is that the set $V$ has to be replaced by $V \cup L$. We briefly describe the analogy. The Ordered Gradual Covering Location Problem

$$\text{max} \sum_{i \in G} \sum_{j=1}^{n} \lambda_i c_{\tau(i)}(S)$$

can be equivalently formulated as

$$\text{min} \sum_{i \in G} \sum_{j=1}^{n} \tilde{\lambda}_i \tilde{c}_{\tau(i)}(S),$$

where $\tilde{\lambda}_i := \lambda_{n-i+1}$ and $\tilde{c}_i(S) := -c_i(S)$. If $\lambda \in \Lambda^b$, then $\tilde{\lambda}$ fulfills (4). As the function $c_i(S \cup \{x\})$ is continuous and convex between consecutive elements of $V \cup L$ on an edge, $\tilde{c}_i(S \cup \{x\})$ is continuous and concave.

The $p$-facility Ordered Median Problem was defined as

$$\text{min} \sum_{i \in G} \sum_{j=1}^{n} \tilde{\lambda}_i d_\tau^{w_i}(S),$$

where $d_\tau^{w}(S) = w_i d_i(S)$ and $\tilde{\lambda}$ fulfills (4). The results for the pOMP in [9] only depend on the fact that the functions $d_\tau^{w}(S \cup \{x\})$ are continuous and concave. Therefore, if we restrict ourselves to subedges between consecutive elements of $V \cup L$ on an edge, we can simply replace $d_\tau^{w}(S)$ by $\tilde{c}_i(S)$ and use the same techniques as in [9] to prove the results.

Before we present the discretization result, we need the following two definitions.

**Definition 3.1 (Ranges).** Let $S \subset G$. Define the set of ranges (canonical set of distances) by

$$R := \{r \in \mathbb{R} \mid \exists x \in EQ_0 : c_i(x) = r = c_i(x) \text{ or } \exists v_i \in V, x \in V \cup L : r = c_i(x)\},$$

and the set of ranges with respect to $S$ by

$$R(S) := \{r \in R \mid \exists x \in EQ_0 \cap S : c_i(x) = r = c_i(x) \text{ or } \exists v_i \in V, \exists x \in (V \cup L) \cap S : r = c_i(x)\}.$$  

The ranges correspond to coverage function values of equilibria or node to node distances.

**Definition 3.2 (Extreme Points).** Let $r \in \mathbb{R}$. A point $x \in G$ is called an $r$-extreme point, if there exists a node $v_i \in V$ with $r = c_i(x)$. By

$$EP(r) = \{x \in G \mid \exists i \in \{1, \ldots, n\} : r = c_i(x)\}$$

we denote the set of all $r$-extreme points on the network and by $EP(Q) = \bigcup_{r \in Q} EP(r)$ the set of $r$-extreme points with respect to a set $Q \subset \mathbb{R}$ of $r$-values.

Now, we can state the FDS.

**Theorem 3.2.** There always exists an optimal solution $S^* \subset EP(R)$ such that $S^* \cap (V \cup L \cup EQ) \neq \emptyset$ and $S^* \subset EP(R(S^*))$.

That is, there always exists an optimal solution $S^*$ which contains a node, an element of $L$, or an equilibrium, and all other solution points are extreme points with respect to the set of distances induced be the nodes, the elements of $L$, and the equilibria of $S^*$. (Observe that each node, element of $L$, or equilibrium is an extreme point with respect to itself.)

Unfortunately, this result does not hold for arbitrary modeling vectors, as the following counterexample shows.

**Example 3.2.** Consider the path graph in Fig. 6 with $l_0 = 0, i = 1, \ldots, 5, u_1 = 10, i = 1, 2, 4, 5,$ and $u_5 = 7$. Let $p = 2$ and $\lambda = (1, 1, 0, 1, 0)$. Fig. 7 shows the linear coverage functions on the edges $[v_1, v_2]$ and $[v_3, v_4]$. As in an optimal solution nodes $v_1$ and $v_2$ will always be covered by a solution point on $[v_1, v_2]$ and the other nodes by a point on $[v_3, v_4]$ or $[v_4, v_5]$, the respective coverage functions are omitted. For this example, $X_2 = (x_1, x_2)$ with $x_1 = ([v_1, v_2], 0.3)$ and $x_2 = ([v_3, v_4], 0.7)$ is an optimal solution. Note that neither solution point is a node, an element of the set $L$, an equilibrium or an extreme point. Moreover, $c_1(x_1) = c_4(x_2)$ and $c_2(x_1) = c_5(x_2)$, i.e., we have “equality” at two positions.
4. Solving the discrete multi-facility OGCP

In this section we discuss exact solution approaches for a discrete multi-facility OGCP, i.e., we assume that the set of potential locations for the new facilities is discrete. In view of the results in the previous section, a finite dominating set can, under some conditions, be identified a priori in network location problems, which allows us to treat them as discrete location problems; alternatively, a discrete problem may arise in its own right in cases where potential facility locations have been pre-selected.

Without any loss of generality we assume that \( V \) is the set of potential facility sites (since any non-nodal site can be added to the set of nodes). The following formulation directly extends the standard UFLP formulation [2] and is similar to formulation (OMP) presented in [14].

Let:

\[
\begin{align*}
c_{ij} &= \begin{cases} 
w_i & \text{if } d_i(j) \leq l_i \\
w_i f_i(d_i(j)) & \text{if } l_i \leq d_i(j) \leq u_i \quad \text{for } i, j \in V \\
0 & \text{if } d_i(j) \geq u_i
\end{cases} \\
Q_{ij} &= \{ l \in \{1, \ldots, n\} | c_{il} \geq c_{ij} \}.
\end{align*}
\]

The decision variables and the integer programming formulation can now be stated as follows:

\[
\begin{align*}
y_j &= \begin{cases} 1 & \text{if a new facility is established at } v_j \in V \\
0 & \text{otherwise}
\end{cases} \\
x_{ijk} &= \begin{cases} 1 & \text{if node } v_j \text{ is covered by a facility at } v_j \text{ and the corresponding coverage level is at position } k \text{ in the sorted vector} \\
0 & \text{otherwise}
\end{cases}
\end{align*}
\]

for \( i, j, k = 1, \ldots, n \).

\[
\begin{align*}
\max & \sum_{k=1}^{n} \sum_{i=1}^{n} \sum_{j=1}^{n} \lambda_k c_{ij} x_{ijk} \\
\text{s.t.} & \sum_{j=1}^{n} y_j = p \tag{6} \\
x_{ijk} \leq y_j & \quad \forall i, j, k = 1, \ldots, n \tag{7} \\
\sum_{j=1}^{n} \sum_{k=1}^{n} x_{ijk} = 1 & \quad \forall i = 1, \ldots, n \tag{8} \\
\sum_{i=1}^{n} \sum_{j=1}^{n} x_{ijk} = 1 & \quad \forall k = 1, \ldots, n \tag{9} \\
\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ijk} \leq \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij(k+1)} & \quad \forall k = 1, \ldots, n - 1 \tag{10}
\end{align*}
\]
The improved IP formulation is:

\[
\sum_{k=1}^{n} \sum_{i \in Q_j} x_{ik} \geq y_j \quad \forall i, j = 1, \ldots, n
\]

\[
y_j, x_{ik} \in \{0, 1\} \quad \forall i, j, k = 1, \ldots, n.
\]

Constraints (6) ensure that \( p \) facilities are located. Constraints (7) guarantee that node \( v_j \) can only be covered from \( v_j \) if a facility has been established there. Constraints (8) and (9) make sure that node \( v_i \) is covered exactly once (recall that \( c_{ij} = 0 \) is possible) and, that one coverage value has to be assigned to each position. Constraints (10) ensure that the coverage levels of the nodes are sorted in non-decreasing order for the objective function. Finally, constraints (11) guarantee that any node \( v_i \) is covered at its maximal level. The latter constraints are only required in case of negative modeling vectors \( \lambda \)-otherwise they will hold automatically due to the objective function.

This IP formulation requires \( n^3 + n \) decision variables and \( n^3 + n^2 + 3n \) constraints and thus the dimensionality grows very quickly with \( n \). Note that we could replace Constraints (7) for each \( i \) and \( j \) by an aggregate constraint; however, despite the increased number of constraints, using the disaggregated form usually yields a tighter LP-Relaxation. We now present an alternative IP formulation that takes advantage of the structure of the coverage functions and can yield significantly more compact and solvable formulations.

This formulation is based on the one in Berman and Krass [2] as well as Marin et al. [11]. The main idea is that in order to compute the contribution of customer node \( v_i \in V \) to the objective function, it is not necessary to know where the customer is covered from—we only need to know the coverage level the customer receives. For example, in the traditional coverage context, node \( v_i \) is either covered or not, and thus the coverage function can only take on two values: \( w_i \) or 0. In the gradual coverage framework, the number of possible values of the coverage function can be larger (theoretically, as large as \( n \)—if every possible facility location results in a different coverage), but may also be small in many applications. The formulation below exploits this feature by focusing on the coverage level received by each customer node.

Recall the definition of \( c_{ij} \). For each \( v_i \in V \) the distinct coverage values in the set \( \{c_{ij} \mid v_j \in V\} \) are sorted as \( c_{i0}^{(0)} < \cdots < c_{iG_i}^{(G_i)} \). We will call these values the coverage levels of node \( v_i \). Note that the number of coverage levels \( G_i \leq n \). We also define the following sets and decision variables (the meaning of decision variables \( y_j \) is the same as in the previous formulation):

\[
J_i(r) = \{v_j \in V\mid c_{ij} = c_{i}^{(r)}\} \quad \text{for} \quad v_i \in V \quad \text{and} \quad r \in [0, \ldots, G_i] \quad \text{—the set of nodes from which \( v_i \) can be covered at level} \ r.
\]

\[
x_{ik}^{(r)} = 1 \quad \text{if \ the node with \( k \) smallest coverage level in the solution receives coverage level} \ c_{i}^{(r)}; \ x_{ik}^{(r)} = 0 \quad \text{otherwise. Here} \ k \in \{1, \ldots, n\}, \ r \in \{1, \ldots, G_i\}.
\]

The “improved IP” formulation is:

\[
\text{max} \sum_{k=1}^{n} \sum_{i=1}^{n} \sum_{r=1}^{G_i} \lambda_k c_{i}^{(r)} x_{ik}^{(r)}
\]

s.t.

\[
\sum_{j=1}^{n} y_j = p
\]

\[
x_{ik}^{(r)} \leq \sum_{j \in J_i(r)} y_j, \quad \text{for} \quad i, k = 1, \ldots, n, \ r = 1, \ldots, G_i
\]

\[
\sum_{i=1}^{n} \sum_{r=1}^{G_i} x_{ik}^{(r)} \leq 1, \quad \text{for} \quad k = 1, \ldots, n
\]

\[
\sum_{i=1}^{n} \sum_{r=1}^{G_i} x_{ik}^{(r)} \leq 1, \quad \text{for} \quad i = 1, \ldots, n
\]

\[
\sum_{i=1}^{n} \sum_{r=1}^{G_i} c_{i}^{(r)} x_{ik}^{(r)} \leq \sum_{i=1}^{n} \sum_{r=1}^{G_i} c_{i}^{(r)} x_{i(k+1)}^{(r)} , \quad \text{for} \quad k = 1, \ldots, n - 1
\]

\[
p \sum_{k=1}^{n} \sum_{r=\rho}^{G_i} c_{i}^{(r)} x_{ik}^{(r)} \geq \sum_{j \in J_i(\rho)} y_j , \quad \text{for} \quad i = 1, \ldots, n, \ \rho = 1, \ldots, G_i
\]

\[
x_{ik}^{(r)}, y_j \in \{0, 1\}.
\]

The objective function multiplies the \( k \)-th smallest coverage level by the appropriate component of the modeling vector \( \lambda \). Note that index \( k \) refers to the position of the node-coverage level combination in the ordered list used in the objective function. Constraint (12) require that \( p \) facilities be located, and constraints (13) ensure that node \( v_i \) can receive coverage level \( r \) only if one or more facilities are open in the set \( J_i(r) \). Constraints (14) state there is at most one node-coverage level
Consider a triangle network with link lengths receiving coverage zero, since zero coverage does not contribute to the objective function (such combinations would have index \( r = 0 \), but we always keep \( r \geq 1 \)). Thus, some positions \( k \) may not be assigned to any node-coverage level combination if the number of positive combinations is less than \( n \). That is why constraints (14) are inequalities. Similarly, constraints (15) require that each node \( i \) must be assigned to at most one coverage level and one position. Constraints (16) are sorting constraints, ensuring that the node-coverage level combination assigned to position \( k \) does not have higher coverage than the combination assigned to position \( k + 1 \). These constraints are not necessary when the components of the modeling vector are non-decreasing, since they will be automatically enforced by the objective function. Finally, constraints (17) specify that each node be assigned to the highest possible coverage level; these constraints are not necessary when the components of the modeling vector \( \lambda \) are non-negative.

The improved IP formulation has \( n + n \sum_{i=1}^{n} G_i \leq n + n^2 \) decision variables; when the number of distinct coverage levels is \( n \) for each node \( v_i \in V \) it is equivalent to the previous formulation. However, as noted earlier, \( G_i \) may be much smaller than \( n \) for many applications: e.g., \( G_i = 1 \) for the standard cover model and \( G_i = s - 1 \) when the coverage decay function \( f_i(t) \) is a \( s \)-level step function. In these cases, the improved IP formulation is significantly more compact than the original one. Moreover, as proved in [2] for the GCLP case, the LP relaxation for the improved formulation is just as tight as for the original formulation, thus using the improved formulation cannot hurt in terms of the problem solvability. The improved IP formulation is illustrated in the following example.

**Example 4.1.** Consider a triangle network with link lengths \( l(1, 2) = 2, l(1, 3) = 2, l(2, 3) = 1 \). The coverage decay function is a simple cover with radius 1 (i.e., only node 1 is covered from 1, while nodes 2 and 3 are covered from either 2 or 3). The node weights are \( w_1 = 5, w_2 = 2, w_1 = 1 \), one facility is to be located, and \( \lambda = \{1, 1, 0\} \), indicating that we wish to maximize the total coverage of two worst-covered nodes.

Here the coverage values are the same as node weights, thus, \( c_{10}^{(0)} = c_{20}^{(0)} = c_{30}^{(0)} = 0, c_{11}^{(1)} = 5, c_{21}^{(1)} = 2, \) and \( c_{31}^{(1)} = 1 \). It follows that \( G_1 = G_2 = G_3 = 1 \) and \( J_1(1) = \{1\} \) and \( J_2(1) = J_3(1) = \{2, 3\} \).

The objective function is: \( \max 5(x_{11}^{(1)} + x_{12}^{(1)}) + 2(x_{21}^{(1)} + x_{22}^{(1)}) + (x_{31}^{(1)} + x_{32}^{(1)}) \) (the values corresponding to \( k = 3 \) are skipped since \( \lambda_3 = 0 \)).

Constraint (12) is \( y_1 + y_2 + y_3 = 1 \). The coverage constraints (13) are:

\[
\begin{align*}
   x_{ik}^{(1)} &\leq y_i; \quad k = 1, 2, 3 \\
   x_{ik}^{(1)} &\leq y_2 + y_3 \quad \text{for } i = 2, 3 \text{ and } k = 1, 2, 3.
\end{align*}
\]

Constraints (14) are:

\[
\begin{align*}
   x_{1k}^{(1)} + x_{2k}^{(1)} + x_{3k}^{(1)} &\leq 1, \quad \text{for } k = 1, 2, 3.
\end{align*}
\]

Similarly, constraints (15) are:

\[
\begin{align*}
   x_{i1}^{(1)} + x_{i2}^{(1)} + x_{i3}^{(1)} &\leq 1, \quad \text{for } i = 1, 2, 3.
\end{align*}
\]

Finally, the sorting constraint (16) are:

\[
\begin{align*}
   5x_{11}^{(1)} + 2x_{21}^{(1)} + x_{31}^{(1)} &\leq 5x_{12}^{(1)} + 2x_{22}^{(1)} + x_{32}^{(1)} \\
   5x_{11}^{(1)} + 2x_{21}^{(1)} + x_{31}^{(1)} &\leq 5x_{12}^{(1)} + 2x_{22}^{(1)} + x_{32}^{(1)}.
\end{align*}
\]

Constraints (17) are not required since the modeling vector is non-negative. The solution can be obtained by inspection.

First suppose \( y_1 = 1, y_2 = y_3 = 0 \). Constraints (13) imply that \( x_{ik}^{(1)} = 0 \) for \( i = 2, 3 \) and \( k = 1, 2, 3 \). Constraints (16), together with (14) and (15) imply that \( x_{i1}^{(1)} = 1, x_{i2}^{(1)} = 0, \) and \( x_{i3}^{(1)} = 0 \), leading to the objective function value of 0. Note that no node-coverage level combination is assigned to positions 1 and 2 in this case, signifying that the corresponding coverage levels equal to 0.

On the other hand, if \( y_2 = 1, y_1 = y_3 = 0 \) then \( x_{ik}^{(1)} = 0 \) for \( k = 1, 2, 3 \). Checking the last constraint we see that the only feasible solution is \( x_{21}^{(1)} = 1, x_{31}^{(1)} = 1, \) with all other components of \( x_{ik}^{(1)} \) equal to 0. This solution corresponds to the objective function value of 1.

The case of \( y_3 = 1, y_1 = y_2 = 0 \) leads to the same solution, which must be optimal. Thus, the optimal solution is to locate a facility either at nodes 2 or 3, obtaining the objective function value of 1.

### 4.1. Computational results

To test the solvability of the improved IP formulation we conducted a series of computational experiments. We generated random networks with \( n = 5, 10, 15, \) and 20 nodes. A step-function coverage decay function was used with 5 coverage levels; the corresponding coverage radii were set to 20%, 40%, 60%, 80%, and 100% of the average shortest distance between any two nodes on the network. The number of facilities \( p \) to be located was set to \( p = 1, 3, 5 \). Five types of modeling vectors \( \lambda \) were used, as described below:
Table 1
Computational results for the improved IP for OGCLP for all modeling vector types.

<table>
<thead>
<tr>
<th>N</th>
<th>p</th>
<th>Time (s)</th>
<th>Opt. Gap (%)</th>
<th>% Optimal</th>
<th># Variables</th>
<th># Constraints</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>0.05</td>
<td>57</td>
<td>100</td>
<td>70</td>
<td>93</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>0.04</td>
<td>48</td>
<td>100</td>
<td>70</td>
<td>93</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0.01</td>
<td>46</td>
<td>100</td>
<td>70</td>
<td>93</td>
</tr>
<tr>
<td>5 total</td>
<td></td>
<td>0.03</td>
<td>51</td>
<td>100</td>
<td>70</td>
<td>93</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.24</td>
<td>71</td>
<td>100</td>
<td>360</td>
<td>415</td>
</tr>
<tr>
<td>3</td>
<td>5.03</td>
<td>38</td>
<td>100</td>
<td>360</td>
<td>415</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>4.87</td>
<td>38</td>
<td>100</td>
<td>360</td>
<td>415</td>
<td></td>
</tr>
<tr>
<td>10 total</td>
<td></td>
<td>3.38</td>
<td>50</td>
<td>100</td>
<td>360</td>
<td>415</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>41.01</td>
<td>46</td>
<td>100</td>
<td>915</td>
<td>1005</td>
</tr>
<tr>
<td>3</td>
<td>2444.98</td>
<td>16</td>
<td>100</td>
<td>915</td>
<td>1005</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3173.87</td>
<td>11</td>
<td>100</td>
<td>915</td>
<td>1005</td>
<td></td>
</tr>
<tr>
<td>15 total</td>
<td></td>
<td>1886.62</td>
<td>30</td>
<td>100</td>
<td>915</td>
<td>1005</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>2537.44</td>
<td>62</td>
<td>100</td>
<td>1800</td>
<td>1929</td>
</tr>
<tr>
<td>3</td>
<td>5482.61</td>
<td>16</td>
<td>100</td>
<td>1800</td>
<td>1929</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>5201.48</td>
<td>8</td>
<td>100</td>
<td>1800</td>
<td>1929</td>
<td></td>
</tr>
<tr>
<td>20 total</td>
<td></td>
<td>4407.18</td>
<td>40</td>
<td>100</td>
<td>1800</td>
<td>1929</td>
</tr>
<tr>
<td>Grand total</td>
<td></td>
<td>1574.30</td>
<td>44</td>
<td>100</td>
<td>81</td>
<td></td>
</tr>
</tbody>
</table>

Table 2
Computational results for the improved IP for OGCLP for Type 2 only.

<table>
<thead>
<tr>
<th>N</th>
<th>p</th>
<th>Time (s)</th>
<th>Opt. Gap (%)</th>
<th>% Optimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>1</td>
<td>0.01</td>
<td>29</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>0.02</td>
<td>11</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.01</td>
<td>16</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5 total</td>
<td></td>
<td>0.01</td>
<td>19</td>
<td>100</td>
</tr>
<tr>
<td>10</td>
<td>1</td>
<td>0.03</td>
<td>43</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>0.97</td>
<td>16</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.39</td>
<td>15</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>10 total</td>
<td></td>
<td>0.46</td>
<td>25</td>
<td>100</td>
</tr>
<tr>
<td>15</td>
<td>1</td>
<td>1.38</td>
<td>9</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>6.26</td>
<td>5</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>2.72</td>
<td>4</td>
<td>100</td>
<td></td>
</tr>
<tr>
<td>15 total</td>
<td></td>
<td>3.45</td>
<td>6</td>
<td>100</td>
</tr>
<tr>
<td>20</td>
<td>1</td>
<td>6.42</td>
<td>33</td>
<td>100</td>
</tr>
<tr>
<td>3</td>
<td>4423.51</td>
<td>14</td>
<td>40</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>3757.57</td>
<td>9</td>
<td>60</td>
<td></td>
</tr>
<tr>
<td>20 total</td>
<td></td>
<td>2729.16</td>
<td>22</td>
<td>67</td>
</tr>
<tr>
<td>Grand total</td>
<td></td>
<td>683.27</td>
<td>17</td>
<td>92</td>
</tr>
</tbody>
</table>

Type 1: $\lambda = (1, \ldots, 1, 0, \ldots, 0)$
Type 2: $\lambda = (0, \ldots, 0, 1, \ldots, 1)$
Type 3: Components of $\lambda$ are ascending random numbers in $[0, 1]$
Type 4: $\lambda = (1, \ldots, 1, -1, \ldots, -1)$
Type 5: Components of $\lambda$ were randomly generated in $[0, 1]$.

In total, 300 problem instances were generated. The instances were solved using the CPLEX solver version 8.1 on a Pentium 4 desktop with 3.2 Ghz CPU and 1 MB of RAM. The time limit for each instance was set to 2 h of CPU time. We first generated a random distance matrix with elements drawn uniformly in $[0, 200]$; then the shortest path algorithm was applied to obtain the shortest distance matrix. Next for each node $i$ we generated $K$ random numbers from a uniform distribution in $[0, 10]$. These number were sorted in a non-decreasing order to obtain $r_{ki}^{(1)}$, $k = 1, \ldots, K$. For each node $i$ we generated $K$ random numbers from a uniform distribution in $[0, 1]$. These numbers were sorted in non-increasing order to obtain $\alpha_{ki}^{(2)}$, $k = 1, \ldots, K$. The weights $w_i$ are set to 1 for all $i = 1, \ldots, N$. The same formulation was used for all instances — i.e., constraints (16) and (17) were not dropped even when the structure of the $\lambda$ vector indicated that these constraints were not necessary (this was done to facilitate comparisons across different problem instances). The overall results are presented on Tables 1 and 2. Table 1 summarizes the results for all $\lambda$ types, while Table 2 contains results for Type 2 $\lambda$ vectors only.

In both tables the fourth column (Opt. Gap) is the relative error of using the relaxed LP formulation over the optimal IP formulation. It is equal to

$$\frac{Opt^{LP} - Opt^{IP}}{Opt^{IP}}$$
where $Opt^{IP}$ and $Opt^{RIP}$ are respectively the optimal solutions of the IP and the relaxed LP. The fifth column in Tables 1 and 2 (% Optimal) gives the percentage of times the IP formulation was solved to optimality by CPLEX within 2 h. The sixth column in Table 1 gives the number of decision variables which is equal to the number $n$ (number of $y_j$ decision variables) plus the number of $x^{(r)}_{ik}$ decision variables. Note that the number of decision variables and the number of constraints are included only in Table 1 (since they are identical to these in Table 2 they are not shown).

Overall, the OGCLP appears to be quite difficult to solve except for small instances. As expected, the number of variables and constraints grows rapidly with $n$ — the problem with 20 nodes results in an IP with nearly 2000 variables and constraints. The majority of 20-node problems with more than 1 facility could not be solved within 2 h of CPU time. Moreover, the optimality gap (relative difference between the IP solution and the LP relaxation) is large in most cases, indicating that the formulation is not tight. Recall that the number of decision variables in Tables 1 and 2 is mainly determined by the number of the $x^{(r)}_{ik}$ decision variables which grows rapidly in $n$.

The IP difficulty appears to be largely due to the sorting and largest-level constraints (16) and (17). Recall that when the components of $\lambda$ are non-negative and non-decreasing, these constraints can be removed from the formulation; Type 2 $\lambda$ vectors satisfy this requirement. The resulting formulation is then quite close to the GCLP formulation in [2], which is known to be very integer-friendly — with LP relaxation often having an integer solution or IP achieving integrality after just a few iterations of the solver. Table 2 shows that these properties appear to translate to the OGCLP case as well. Even though constraints (16) and (17) were retained in the formulation, they are redundant (for the IP). It can be seen that optimality gaps are much smaller than for the general case, all instances with less than 20 nodes were solved in a few seconds, and the solution failures for 20-node instances with $p = 3$ and 5 were mostly due to memory issues (i.e., problem size) rather than to the tightness of the formulation. Similar improvements were observed for Type 3 $\lambda$ vectors and, to a lesser extent for Type 4 vectors—in the latter case, only constraints (16) can be removed. On the other hand, the worst results were observed for decreasing $\lambda$ vectors of Type 1 — insufficient tightness of the formulation appears to be particularly severe in this case.

In summary, our results indicate that IP-based approach to general OGCLP instances can only deal with relatively small problem instances; as noted in [14], the sorting constraints appear to cause particular problems for integer programming approaches in OMP-type models. The situation is better when the components of the modeling vector are non-decreasing and non-negative, however the large dimensionality of the formulation is a concern in that case as well. On one hand, this lack of solvability is hardly surprising — after all, as shown earlier, OGCLP is a very general model including most standard location models as special cases; no easily-exploitable special structure is present for this case. On the other hand, further work on exact and approximate solution techniques for OGCLP is clearly in order, as the potential payoff (having a general method for most types of location problems) is large.

5. Summary

In this paper we formulated a new network location model – Ordered Gradual Cover Location Problem – which generalizes location problems with median, center and cover objectives, as well as their extensions, such as Ordered Median and Gradual Cover. Finite Dominating Set results were obtained for many types of modeling vectors for the 1-facility case; more restrictive results were derived for the multi-facility case. We also investigated exact solutions of discrete version of OGCLP via Integer Programming.

Clearly, much work remains to be done, particularly in the area of efficient solution techniques. Sorting constraints needed to capture the effects of modeling vectors lead to formulations that are large and not very tight. Constraint programming approaches (which may have an easier time of representing such constraints) may be particularly effective here. Heuristic algorithms should be investigated in future work as well — after all, such algorithms are quite effective for gradual cover problems. Of course, having an effective algorithm (either exact or approximate) for such a general problem would be extremely useful.

Acknowledgments

This research has been partially supported by the German Academic Exchange Service (DAAD) (grant D/08/11096) and by the NSERC grants of Berman and Krass.

References