Computers & Operations Research 38 (2011) 1493–1500

Contents lists available at ScienceDirect

Computers & Operations Research

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

The probabilistic gradual covering location problem on a network with discrete random demand weights

Oded Berman [a], Dmitry Krass [a], Jiamin Wang [b,*]

[a] Joseph L. Rotman School of Management, University of Toronto, 105 St. George Street, Toronto, Ontario, Canada M5S 3E6
[b] College of Management, Long Island University, C. W. Post, 720 Northern Blvd, Brookville, NY 11548, United States of America

ARTICLE INFO

Available online 19 January 2011

Keywords:
Location
Network
Covering
Probability
Optimization

ABSTRACT

We study the gradual covering location problem on a network with uncertain demand. A single facility is to be located on the network. Two coverage radii are defined for each node. The demand originating from a node is considered fully covered if the shortest distance from the node to the facility does not exceed the smaller radius, and not covered at all if the shortest distance is beyond the larger radius. For a distance between these two radii, the coverage level is specified by a coverage decay function. It is assumed that demand weights are independent discrete random variables. The objective of the problem is to find a location for the facility so as to maximize the probability that the total covered demand weight is greater than or equal to a pre-selected threshold value. We show that the problem is NP-hard and that an optimal solution exists in a finite set of dominant points. We develop an exact algorithm and a normal approximation solution procedure. Computational experiment is performed to evaluate their performance.

1. Introduction

The maximal covering location problem [7] is one of the classical models in facility location theory. A demand point is considered completely covered if it is within a reasonable travel distance, i.e., the *coverage radius* from at least one facility, and not covered at all if it is outside the coverage radius from any facility. The objective of the model is to establish facilities so as to cover as many customers as possible. The model can be applied to various settings, including the location of fire stations, retail stores, etc. Readers are referred to ReVelle and Williams [11] for a recent review of related literature.

As an extension of the above model, the gradual covering location problem [8,1] relaxes the all-or-nothing coverage assumption. Two coverage radii are specified. A demand point is "fully" covered if it is within the smaller coverage radius from the facilities and not covered if it is at or beyond the larger coverage radius away from the facilities. It is deemed "partially" covered if its distance to the closest facility is between the two coverage radii. A coverage decay function is introduced to determine the proportion of customers at a partially covered demand point that are covered by the facilities. The coverage decay function is usually non-increasing with distance.

In network location models, demand weights are used to gauge the number of potential customers originated from nodes. Demand weights are generally assumed to be constant and known. It is noted, however, that treating demand weights as random variables is more reasonable [9].

The *expected value optimization model* is commonly used to solve stochastic decision making problems. This model is appropriate for a decision maker who can make the decision repetitively. However, it is usually costly or infeasible to change a strategic decision such as locating facilities. Thus the model to maximize the probability of achieving a given threshold value (referred to as "*P*" *model* by Charnes and Cooper [6]) shall be ideal for the decision maker in such a situation who is concerned with the risk of poor performance.

As Charnes and Cooper [6] commented, a "*P*" model can be linked to the concept of "satisfying" (in contrast to "optimizing") that was developed by Simon [12] in behavior theories. In this perspective, the specified threshold value can be interpreted as an "aspiration level", while the objective is to maximize the likelihood of attaining the target.

There have been a few investigations of the "*P*" model formulations in location theory (see, e.g., [9,2,3,5]). Berman and Wang [4] motivated and discussed the maximal covering location problem with probabilistic demand weights.

In this paper, we study the gradual covering problem on a network where demand weights are independent discrete random variables. The objective of the problem is to locate a facility

* Corresponding author. Tel.: +1 516 299 3914.
E-mail address: jiamin.wang@liu.edu (J. Wang).

0305-0548/$ - see front matter © 2011 Elsevier Ltd. All rights reserved.
doi:10.1016/j.cor.2011.01.005

so as to maximize the probability that the total demand weight covered exceeds a pre-selected threshold value.

In the next section, a mathematical formulation is introduced and the problem is shown to be NP-hard. In Section 3, we develop an exact solution procedure to search for the optimal solution. As a generalization, we also discuss the case where coverage decay functions are piece-wise functions. Since the problem is NP-hard, a normal approximation approach is suggested in Section 4. In Section 5, a computation experiment is conducted to evaluate the performance of the normal approximation approach for linear coverage decay functions.

2. Problem statement

Let G be an undirected network with a set of nodes N ($|N|=n$) and a set of links L ($|L|=l$). Denote by l_{ij} the length of link $(i,j)\in L$ and by $d(h,x)$ the shortest distance between a node h and an arbitrary point $x\in G$. When there is no ambiguity, the same notation $x\in[0,l_{ij}]$ represents both, a point on link (i,j), and the distance of the point from node i.

Two coverage radii $\hat{r}_h$ and $\tilde{r}_h$ ($\hat{r}_h<\tilde{r}_h$) are defined for every node $h\in N$. If $d(h,x)$ is between $\hat{r}_h$ and $\tilde{r}_h$, node h is partially covered by x with the level of coverage given by a coverage decay function $\lambda_h(d(h,x))$. It is intuitive that the function $\lambda_h(\cdot)$ is non-increasing with $\lambda_h(\hat{r}_h)=1$ and $\lambda_h(\tilde{r}_h)=0$. Here we require that the coverage decay function $\lambda_h(\cdot)$ for every node h is monotone decreasing and continuous.

We assume that demand generates from the nodes only. The demand weight W_h associated with node $h\in N$ is a *discrete* random variable with possible values $w_h[k]$, $k=1,2,\ldots,K_h$, where K_h represents the number of realizations of W_h. Let $p_h[k]=P(W_h=w_h[k])$. Without loss of generality, it is assumed that $0\le w_h[1]<w_h[2]<\cdots<w_h[K_h]$, $\forall h\in N$.

Denote by $c_h(x)$ the proportion of demand weight at node h that is covered by the facility established at the point x. $c_h(x)$ is expressed as

$$c_h(x)=\begin{cases}1, & d(h,x)\le\hat{r}_h,\\ \lambda_h(d(h,x)), & \hat{r}_h<d(h,x)<\tilde{r}_h,\\ 0, & d(h,x)\ge\tilde{r}_h.\end{cases}$$

Given the point x, $\sum_{h\in N}c_h(x)W_h$ is the total demand weight covered by the facility located at x. Since W_h associated with each node h follows a discrete probability distribution, $\sum_{h\in N}c_h(x)W_h$ is also a discrete random variable. The objective of the probabilistic gradual covering problem (referred to as the problem (P) hereinafter) is to locate a facility at some point x so as to maximize the probability of the total covered demand weight exceeding a pre-determined threshold value $t>0$:

$$\max_{x\in G} f(x)=P\left(\sum_{h\in N}c_h(x)W_h\ge t\right). \tag{P}$$

Note that $f(x)=0$ if $c_h(x)=0$ for every node $h\in N$ (that is, none of the nodes is even partially covered). If the demand weights are deterministic, the problem (P) reduces to the deterministic gradual covering problem,

$$\max_{x\in G} f'(x)=\sum_{h\in N}c_h(x)W_h. \tag{1}$$

Suppose that the random demand weights are *independent*. Given the facility location x, every node out of coverage can be excluded from consideration without affecting the computation of the objective value $f(x)$. Let $\mathbf{W}(x)$ be the random vector of demand weights associated with the nodes in the set $E(x)=\{h\in N|\ c_h(x)>0\}$. Denote by $B(x)$ the Cartesian product of realizations of W_h, $\forall h\in E(x)$. We note that every vector $\mathbf{w}$ in $B(x)$ is a realization of $\mathbf{W}(x)$ and can be expressed as $\mathbf{w}=(w_{h^{(1)}},w_{h^{(2)}},\ldots,w_{h^{(|E(x)|)}})$. Given a vector $\mathbf{w}$, we define

$$g_{\mathbf{w}}(x)=\begin{cases}1 & \text{if } \sum_{h^{(q)}\in E(x)}c_{h^{(q)}}(x)w_{h^{(q)}}\ge t,\\ 0 & \text{otherwise.}\end{cases} \tag{2}$$

The objective value $f(x)$ is calculated as

$$f(x)=\sum_{\mathbf{w}\in B(x)}g_{\mathbf{w}}(x)p_{\mathbf{w}}, \tag{3}$$

where $p_{\mathbf{w}}=\prod_{h^{(q)}\in E(x)}P(W_{h^{(q)}}=w_{h^{(q)}})$ for a given vector $\mathbf{w}\in B(x)$.

We note that the problem (P) reduces to the probabilistic maximal covering problem studied by Berman and Wang [4] if $\hat{r}_h=\tilde{r}_h$ for every node h. They prove that computing the objective value for the probabilistic maximal covering problem is NP-hard when the demand weights are discrete random variables. This conclusion can be easily generalized to the problem (P). Therefore, the problem (P) is also NP-hard.

We next discuss some special cases for which an optimal solution can be easily determined. Suppose that $x^{(1)}$ and $x^{(2)}$ are, respectively, the optimal solutions to the deterministic gradual covering problem (1) with $W_h=w_h[1]$ and $W_h=w_h[K_h]$, $\forall h\in N$. Let $t_1=\sum_{h\in N}c_h(x^{(1)})w_h[1]$ and $t_2=\sum_{h\in N}c_h(x^{(2)})w_h[K_h]$. In the following lemmas we show that the problem (P) can be solved easily if t is sufficiently small or large.

Lemma 2.1. *If $t\le t_1$, $x^{(1)}$ is optimal to the problem* (P).

Proof. It is easy to see that $f(x^{(1)})=1$. □

Lemma 2.2. *If $t>t_2$, the objective value $f(x)=0$ at every point $x\in G$.*

Proof. Since $\sum_{h^{(q)}\in E(x^{(2)})}c_{h^{(q)}}(x^{(2)})w_{h^{(q)}}<t$ for any $\mathbf{w}\in B(x^{(2)})$, we have $f(x^{(2)})=0$. Note that $\sum_{h^{(q)}\in E(x)}c_{h^{(q)}}(x)w_{h^{(q)}}[K_h]\le\sum_{h^{(q)}\in E(x^{(2)})}c_{h^{(q)}}(x^{(2)})w_{h^{(q)}}[K_h]<t$ holds at every point $x\in G$. In a similar way, we can show that $f(x)=0$ at x. □

Next, ranges of radii are established for which an optimal solution to the problem (P) can be easily identified. Suppose that $x^{(0)}$ is the 1-center of the network [10], i.e., $x^{(0)}=\operatorname{argmin}_{x\in G}\max_{h\in N}d(h,x)$ and hence $\max_{h\in N}d(h,x^{(0)})$ is the network radius.

Lemma 2.3. *If $\min_{h\in N}\hat{r}_h\ge\max_{h\in N}d(h,x^{(0)})$, $x^{(0)}$ is an optimal solution to the problem* (P).

Proof. For every node h, we have $d(h,x^{(0)})\le\hat{r}_h$ and hence $c_h(x^{(0)})=1$. Therefore, $x^{(0)}$ is optimal. □

Let node $\hat{h}=\operatorname{argmax}_{h\in N}P(W_h\ge t)$ and link $(a,b)=\operatorname{argmin}_{(i,j)\in L}l_{ij}$ (ties are broken arbitrarily). That is, (a,b) is the shortest link.

Lemma 2.4. *If $\max_{h\in N}\tilde{r}_h\le\frac{1}{2}l_{ab}$, node $\hat{h}$ is optimal to the problem* (P).

Proof. Because at most one node is fully or partially covered by a facility established at every point, one of the nodes must be optimal. The problem (P) reduces to $\max_{h\in N}f(h)=P(W_h\ge t)$. □

3. Exact solution procedure

In this section, we consider solving a general instance of the problem (P) where $\min_{h\in N}\hat{r}_h<\max_{h\in N}d(h,x^{(0)})$, $\max_{h\in N}\tilde{r}_h>\frac{1}{2}l_{ab}$, and $t_1<t\le t_2$. It is sufficient to solve the problem on every link of the network. We now develop a procedure for finding an optimal solution on an arbitrarily selected link $(i,j)\in L$. Without loss of generality, we assume $d(i,j)=l_{ij}$ because otherwise link (i,j) can be eliminated without affecting optimality.

Suppose that the facility is located at x. Given a vector $\mathbf{w}\in B(x)$, $\theta_{\mathbf{w}}(x)=\sum_{h\in E(x)}c_h(x)w_h$ is the total demand weight covered by the facility. As will be shown later, the objective function $f(x)$ is a

step-wise function of the facility location x along link (i,j) with a finite number of jump points $x=y$, where $\theta_{\mathbf{w}}(y)=t$ holds for some vector $\mathbf{w}\in B(y)$. Furthermore, the optimum within the link can be found at one of the jump points of $f(x)$. In order to identify these jump points, we develop a procedure in Section 3.1 to divide the entire link into *primary intervals* over which the closed form expression of function $\theta_{\mathbf{w}}(x)$ for every $\mathbf{w}\in B(x)$ is invariant. It will become clear later that since all functions $\theta_{\mathbf{w}}(x)$ have identical break points along the link, such a method enables us to simplify the computational process and also focus on a subset of potential solutions by taking advantage of dominance relations between these jump points within each primary interval.

3.1. Primary intervals

The shortest distance from node h to the point x is computed as $d(h,x)=\min(d(h,i)+x,\ d(h,j)+l_{ij}-x)$. Define four *network intersection points* for every node h: $\hat{y}_h=\hat{r}_h-d(h,i)$, $\tilde{y}_h=\tilde{r}_h-d(h,i)$, $\hat{z}_h=d(h,j)+l_{ij}-\hat{r}_h$, and $\tilde{z}_h=d(h,j)+l_{ij}-\tilde{r}_h$.

To help conceptualize a network intersection point, we note that the length of the shortest path connecting $\hat{y}_h$ ($\tilde{y}_h$) and node h via node i is exactly $\hat{r}_h$ ($\tilde{r}_h$), if $\hat{y}_h$ ($\tilde{y}_h$) is between 0 and l_{ij} (i.e., if it is an *internal point* of the link). Similarly, if $\hat{z}_h$ ($\tilde{z}_h$) is an internal point, then the shortest path from $\hat{z}_h$ ($\tilde{z}_h$) to node h via node j has a length of $\hat{r}_h$ ($\tilde{r}_h$). In addition, $\hat{y}_h$ ($\tilde{y}_h$) <0 or $\hat{z}_h$ ($\tilde{z}_h$) $>l_{ij}$ means that node h is out of the coverage radius $\hat{r}_h$ ($\tilde{r}_h$) from node i or node j, respectively. Therefore, node h is not within the coverage radius $\hat{r}_h$ ($\tilde{r}_h$) from any point on link (i,j) if $\hat{y}_h$ ($\tilde{y}_h$) <0 and $\hat{z}_h$ ($\tilde{z}_h$) $>l_{ij}$.

The next two lemmas are introduced to facilitate our analysis.

Lemma 3.1. *Given node h and a constant r, $d(h,x)\le r$ holds at every point $x\in(i,j)$ if $d(h,j)+l_{ij}-r\le r-d(h,i)$.*

Proof. Since $d(h,x)=\min(d(h,i)+x,\ d(h,j)+l_{ij}-x)$ at a given point $x\in(i,j)$, it follows that $\min\{d(h,i),\ d(h,j)\}\le d(h,x)\le \frac{1}{2}[d(h,j)+l_{ij}+d(h,i)]$. Since $d(h,j)+l_{ij}-r\le r-d(h,i)$ is equivalent to $\frac{1}{2}[d(h,j)+l_{ij}+d(h,i)]\le r$, we have $d(h,x)\le r$ at every point x. □

Lemma 3.2. *Given node h and a constant r, $d(h,x)\ge r$ holds at every point $x\in(i,j)$ if $r-d(h,i)\le 0$ and $d(h,j)+l_{ij}-r\ge l_{ij}$.*

Proof. Note $\min\{d(h,i),\ d(h,j)\}\ge r$. It follows that $d(h,x)\ge\min\{d(h,i),\ d(h,j)\}\ge r$ holds at every point x. □

We now divide N into four subsets:

$$N_{fc}=\{h\in N|\hat{z}_h\le\hat{y}_h\},$$
$$N_{uc}=\{h\in N|\tilde{y}_h\le 0 \text{ and } \tilde{z}_h\ge l_{ij}\},$$
$$N_{pc}=\{h\in N|\hat{y}_h\le 0,\ \hat{z}_h\ge l_{ij}, \text{ and } \tilde{z}_h\le\tilde{y}_h\},$$
$$N_{vc}=N-N_{fc}-N_{uc}-N_{pc}.$$

The next lemma characterizes N_{fc}, N_{uc} and N_{pc}.

Lemma 3.3. *For every point $x\in(i,j)$, $c_h(x)=1$ if $h\in N_{fc}$, $c_h(x)=0$ if $h\in N_{uc}$ and $0\le c_h(x)\le 1$ if $h\in N_{pc}$.*

Proof. The lemma is a natural result of Lemmas 3.1 and 3.2. □

Lemma 3.3 suggests that node h is fully covered, not covered, and partially covered by a facility sited anywhere on link (i,j) if $h\in N_{fc}$, $h\in N_{uc}$ and $h\in N_{pc}$, respectively.

As the potential facility location x moves along the link, the closed form expression of $c_h(x)$ in terms of x for some node h changes when (i) the coverage class of h in the set N_{vc} varies (e.g., from full coverage to partial coverage, or from partial coverage to non-coverage) at a corresponding network intersection point; or (ii) the shortest path to h in the set N_{vc} or N_{pc} shifts (hence the closed form expression of the function $d(h,x)$ varies). These changes pose difficulties to formulate function $\theta_{\mathbf{w}}(x)$ for any given vector $\mathbf{w}$. Next we identify segments of the link where the above changes cannot occur.

According to Lemma 3.1, we note that the coverage class of a node h in N_{vc} does not change at $\tilde{y}_h$ and $\tilde{z}_h$ when $\tilde{z}_h\le\tilde{y}_h$ even though they are internal points of the link. We call such network intersection points *redundant*. Let J be the collection of the two nodes i and j, and the non-redundant network intersection points that are between 0 and l_{ij} on the link for every node $h\in N_{vc}$. Sort the elements in the set J in ascending order and arbitrarily select two adjacent elements, denoted by $\hat{x}$ and $\tilde{x}$. For a given segment $[\hat{x},\tilde{x}]$, N_{vc} is further divided into three subsets,

$$M_{fc}=\{h\in N_{vc}|\hat{y}_h\ge\tilde{x} \text{ or } \hat{z}_h\le\hat{x}\},$$
$$M_{uc}=\{h\in N_{vc}|\tilde{y}_h\le\hat{x} \text{ and } \tilde{z}_h\ge\tilde{x}\},$$
$$M_{pc}=N_{vc}-M_{fc}-M_{uc}.$$

Similar to Lemma 3.3, we have $c_h(x)=1$ if $h\in M_{fc}$, $c_h(x)=0$ if $h\in M_{uc}$ and $0<c_h(x)<1$ if $h\in M_{pc}$ at every point $x\in(\hat{x},\tilde{x})$. In words, node h is fully covered, not covered, and partially covered by a facility established everywhere inside the segment $(\hat{x},\tilde{x})$ if $h\in N_{fc}\cup M_{fc}$, $h\in N_{uc}\cup M_{uc}$ and $h\in N_{pc}\cup M_{pc}$, respectively. It follows that $E(x)=N-N_{uc}\cup M_{uc}$. It is easy to show that the random vector $\mathbf{W}(x)$ and the set $B(x)$ do not change with x within the segment $[\hat{x},\tilde{x}]$. For simplicity, we use the notation B to replace $B(x)$.

The *antipode* on link (i,j) with respect to node h is obtained as a point x such that $d(h,i)+x=d(h,j)+l_{ij}-x$. In other words, the shortest paths from x to node h through node i and node j have the same length. Note that the closed form expression of the function $d(h,x)$ in terms of x changes at the antipode with respect to h. In the segment $[\hat{x},\tilde{x}]$, let J' be the set of the two end points $\hat{x}$, $\tilde{x}$ and the antipode within the segment with respect to every node $h\in N_{pc}\cup M_{pc}$.

A sub-interval $[\hat{x}',\tilde{x}']$ is called a primary interval if $\hat{x}'$ and $\tilde{x}'$ are two adjacent elements in J'. Note that $[\hat{x}',\tilde{x}']$ coincides with the segment $[\hat{x},\tilde{x}]$ if $\hat{x}$ and $\tilde{x}$ are the only two elements in J'. For every vector $\mathbf{w}\in B$, we have $\theta_{\mathbf{w}}(x)=\sum_{h\in N_{pc}\cup M_{pc}}c_h(x)w_h+\sum_{h\in N_{fc}\cup M_{fc}}w_h$ at any point $x\in[\hat{x},\tilde{x}]$. It becomes obvious that the closed form expression of $\theta_{\mathbf{w}}(x)$ in terms of x is invariant within any primary interval.

Link (i,j) can now be regarded as a union of primary intervals. On the link, there are at most four network intersection points and one antipode with respect to every node. Hence, the number of primary intervals is of the order of $O(n)$. Subsequently, we seek the optimum within a primary interval $[\hat{x}',\tilde{x}']$.

3.2. Critical points

Define a *critical point* x, if available in the primary interval $[\hat{x}',\tilde{x}']$ with respect to a vector $\mathbf{w}\in B$ such that $\theta_{\mathbf{w}}(x)=t$. Note that one of the following four scenarios applies at a critical point x: (1) $\theta_{\mathbf{w}}(x-\varepsilon)<t$ and $\theta_{\mathbf{w}}(x+\varepsilon)>t$; (2) $\theta_{\mathbf{w}}(x-\varepsilon)>t$ and $\theta_{\mathbf{w}}(x+\varepsilon)<t$; (3) $\theta_{\mathbf{w}}(x-\varepsilon)<t$ and $\theta_{\mathbf{w}}(x+\varepsilon)<t$; (4) $\theta_{\mathbf{w}}(x-\varepsilon)>t$ and $\theta_{\mathbf{w}}(x+\varepsilon)>t$ (ε is a positive number sufficiently small).

The critical point in the primary interval $[\hat{x}',\tilde{x}']$ with respect to each vector $\mathbf{w}$ can be obtained by solving the equation $\theta_{\mathbf{w}}(x)=t$ for x (note that the equation may have no root, one root or alternative roots within the primary interval). Denote by S the set of all these critical points with respect to every vector in B at which one of the first three scenarios applies. The theorem below claims that the objective function $f(x)$ is step-wise with the critical points in the set S as break points.

Theorem 3.1. *If the set S is not empty, then one of the elements in S is optimal to the problem* (P) *over the primary interval* $[\hat{x}',\tilde{x}']$.

Proof. If S is empty, then the objective function $f(x)$ is constant over $[\hat{x}', \tilde{x}']$. Otherwise, suppose that $\overline{x} \in S$ is a critical point inside $[\hat{x}', \tilde{x}']$ with respect to a vector $\overline{\mathbf{w}}$. It is evident that $\theta_{\overline{\mathbf{w}}}(\overline{x}) = t$, $\theta_{\overline{\mathbf{w}}}(\overline{x}-\varepsilon) \neq t$, $\theta_{\overline{\mathbf{w}}}(\overline{x}+\varepsilon) \neq t$, and $g_{\overline{\mathbf{w}}}(x)$ changes value only at $\overline{x}$ when x moves from $\overline{x}-\varepsilon$ to $\overline{x}+\varepsilon$ along the link. By (3), $f(x)$ is a step-wise function that breaks at the critical points in S. Hence, one of the elements in S is optimal. □

By the above theorem, any critical point is a jump point of the objective function $f(x)$. A straightforward approach to find the maximum within a primary interval is to evaluate and compare values of $f(x)$ at the critical points in set S. However, we note that dominance relations in S can be developed so that some critical points could be excluded from further consideration.

Sort all the elements in S in ascending order and denote by s_q the qth element. We further construct the following three subsets of S:

$S_l = \{x | x$ is a critical point with respect to $\mathbf{w}$, $\theta_{\mathbf{w}}(x-\varepsilon) > t$ and $\theta_{\mathbf{w}}(x+\varepsilon) < t\}$,
$S_g = \{x | x$ is a critical point with respect to $\mathbf{w}$, $\theta_{\mathbf{w}}(x-\varepsilon) < t$ and $\theta_{\mathbf{w}}(x+\varepsilon) > t\}$,
$S_m = \{x | x$ is a critical point with respect to $\mathbf{w}$, $\theta_{\mathbf{w}}(x-\varepsilon) < t$ and $\theta_{\mathbf{w}}(x+\varepsilon) < t\}$.

We note that a point in S may be "critical" with respect to various vectors in B and thus may belong to one, two or all three of the subsets defined above.

We now introduce two lemmas that characterize the dominated points in the set S_l.

Lemma 3.4. *If s_q, $s_{q+1} \in S_l$ with $q < |S|$ and $s_{q+1} \notin S_g \cup S_m$, then $f(s_q) > f(s_{q+1})$.*

Proof. There exists a vector $\overline{\mathbf{w}} \in B$ such that $\theta_{\overline{\mathbf{w}}}(s_q-\varepsilon) > t$, $\theta_{\overline{\mathbf{w}}}(s_q) = t$ and $\theta_{\overline{\mathbf{w}}}(s_q+\varepsilon) < t$. The Lemma follows because $g_{\overline{\mathbf{w}}}(s_q) = 1$ and $g_{\overline{\mathbf{w}}}(s_{q+1}) = 0$. □

Lemma 3.5. *If $s_q \in S_g \cup S_m$, $s_{q+1} \in S_l$ with $q < |S|$ and $s_{q+1} \notin S_g \cup S_m$, then $f(s_q) \geq f(s_{q+1})$.*

Proof. If $s_q \in S_l$, by Lemma 3.4 we have $f(s_q) > f(s_{q+1})$. In a similar way to the proof for Lemma 3.4, we can show that $f(s_q) > f(s_{q+1})$ also holds if $s_q \in S_m$, and $f(s_q) = f(s_{q+1})$ holds if $s_q \in S_g$, but $s_q \notin S_l \cup S_m$. □

The above two lemmas suggest that a critical point s_q belonging to the set S_l is dominated and can be ignored in the optimization process unless it is the smallest element ($q = 1$) or it is also in the set S_g or the set S_m.

The next two lemmas on the set S_g can be proven easily.

Lemma 3.6. *If s_q, $s_{q+1} \in S_g$ with $q < |S|$ and $s_q \notin S_l \cup S_m$, then $f(s_q) < f(s_{q+1})$.*

Lemma 3.7. *If $s_q \in S_g$, $s_{q+1} \in S_m$ with $q < |S|$ and $s_q \notin S_l \cup S_m$, then $f(s_q) < f(s_{q+1})$.*

Lemmas 3.6 and 3.7 imply that a critical point s_q in the set S_g can be excluded from consideration unless (1) it also belongs to the set S_l or the set S_m; or (2) it is the largest element in the set S_g; or (3) its neighbor s_{q+1} is in the set S_l only.

Taking advantage of the properties stated above, we can rule out dominated critical points in our search for an optimum point within the primary interval $[\hat{x}', \tilde{x}']$. Actually, we can consider only the dominant points below: (1) $s_q \in S_l \cap S_g$; (2) $s_q \in S_m$; (3) $s_q \in S_g$, if s_q is the largest element in the set S_g or $s_{q+1} \notin S_g \cup S_m$; and (4) s_1 if $s_1 \in S_l$.

For a critical point $x \in S$, define

$$p_{S_l}(x) = \begin{cases} \sum_{\mathbf{w} \in R(x)} p_{\mathbf{w}} & \text{if } x \in S_l, \\ 0 & \text{otherwise,} \end{cases}$$

where $R(x) = \{\mathbf{w} \in B | x$ is a critical point with respect to $\mathbf{w}$, $\theta_{\mathbf{w}}(x-\varepsilon) > t$ and $\theta_{\mathbf{w}}(x+\varepsilon) < t\}$. Note that for every vector $\mathbf{w} \in R(x)$ (the cardinality of the set $R(x)$ may be greater than 1), we have $g_{\mathbf{w}}(x-\varepsilon) = g_{\mathbf{w}}(x) = 1$ and $g_{\mathbf{w}}(x+\varepsilon) = 0$. Similarly, we can define $p_{S_g}(x)$ and $p_{S_m}(x)$.

Since there may exist more than one critical point inside the primary interval $[\tilde{x}', \hat{x}']$ with respect to a vector $\mathbf{w}$, we further define a collection of the combinations of critical points and the corresponding random vectors:

$Q = \{(x, \mathbf{w}) | x \in S_l$, $\mathbf{w} \in R(x)$, x is the only or the smallest critical point in the primary interval $[\hat{x}', \tilde{x}']$ with respect to $\mathbf{w}\}$. Given $(x, \mathbf{w}) \in Q$, we note $g_{\mathbf{w}}(y) = 1$ at every point $y \in [\hat{x}', x]$ and $g_{\mathbf{w}}(y) = 0$ at every point $y \in (x, x']$, where x' is either the second smallest critical point with respect to $\mathbf{w}$ if available, or the right end point $\tilde{x}'$ otherwise.

Let $A = \{\mathbf{w} \in B | \theta_{\mathbf{w}}(x) > t$ for every $x \in [\hat{x}', \tilde{x}']\}$. That is, $g_{\mathbf{w}}(x) = 1$ always holds over the primary interval $[\hat{x}', \tilde{x}']$ for every vector $\mathbf{w} \in A$.

The next lemma is natural.

Lemma 3.8. $f(\hat{x}') \geq \sum_{\mathbf{w} \in A} p_{\mathbf{w}} + \sum_{(x, \mathbf{w}) \in Q} p_{\mathbf{w}}$.

We now introduce an algorithm to solve the problem on link (i,j). The procedure starts with dividing the link into primary intervals. Then the algorithm formulates the function $\theta_{\mathbf{w}}(x)$ for every vector $\mathbf{w} \in B$ and identifies and evaluates dominant critical points within each primary interval. The dominant point with the highest objective value is returned as the optimum. f^* and x^* contain, respectively, the incumbent optimal objective value and optimal solution. To quickly determine whether a specific critical point s_q is the last element in the set S_g for a given primary interval, we introduce a variable q' to indicate the position of the current critical point under examination.

Algorithm 1 (*Solving the Problem (P) on Link (i,j).*).

Step 1: On link (i,j) compute the network intersection points $\hat{y}_h$, $\tilde{y}_h$, $\hat{z}_h$, and $\tilde{z}_h$ for every node h and then construct the sets N_{fc}, N_{uc}, N_{pc}, N_{vc} and J as defined above. Sort the set J in an ascending order. Let $f^* = -\infty$.
Step 2: Select a pair of consecutive elements $\hat{x}$ and $\tilde{x}$ in the set J. If all such pairs have been examined, go to Step 10.
Step 3: Construct the sets M_{fc}, M_{uc}, M_{pc}, B and J' as defined above for the segment $[\hat{x}, \tilde{x}\,]$. Sort the set J' in ascending order.
Step 4: Select a pair of consecutive elements $\hat{x}'$ and $\tilde{x}'$ in the set J'. If all such pairs have been examined, go to Step 2.
Step 5: For the primary interval $[\hat{x}', \tilde{x}']$, construct the sets S, S_l, S_g, S_m, Q and A and compute $p_{S_l}(x)$, $p_{S_g}(x)$ and $p_{S_m}(x)$ at each critical point $x \in S$. Sort the set S in an ascending order.
Step 6: Let $q = q' = 0$ and $c = \sum_{\mathbf{w} \in A} p_{\mathbf{w}} + \sum_{(x, \mathbf{w}) \in Q} p_{\mathbf{w}}$. If $c > f^*$, let $x^* = \hat{x}'$ and $f^* = c$.
Step 7: If $q > 0$ and $c > f^*$, let $x^* = s_q$ and $f^* = c$.
Step 8: Increase q by 1. If $q > |S|$, go to Step 4. Let $c = c - p_{S_l}(s_{q-1}) - p_{S_m}(s_{q-1}) + p_{S_g}(s_q) + p_{S_m}(s_q)$ (let $p_{S_l}(s_{q-1}) = p_{S_m}(s_{q-1}) = 0$ when $q = 1$). If $s_q \notin S_g \cup S_m$, repeat Step 8. If $s_q \in S_g$, increase q' by 1.
Step 9: If any of the following cases occurs, go to Step 7; otherwise, go to Step 8.
- $s_q \in S_l \cup S_m$;
- $q' = |S_g|$ (i.e., s_q is the last element in the set S_g);

- $q' < |S_g|$ and $s_{q+1} \notin S_g \cup S_m$ (i.e., s_q is not the last element in the set S_g and s_{q+1} is not in the set S_g nor the set S_m).

Step 10: Stop. Return x^* as the optimal solution with an objective value f^*.

To illustrate Algorithm 1, consider the 4-node network as shown in Fig. 1. The probabilistic distributions of the demand weights are defined in Table 1. Assume that the coverage decay function is linear in $d(h,x)$ and formulated as $\lambda_h(d(h,x)) = \alpha_h + \beta_h d(h,x)$ for every node h. The coverage radii and the coefficients of the coverage decay functions are presented in Table 2. Let $t = 11$. We next apply Algorithm 1 to solve the problem (P) on link (1,2).

From the network structure, we can obtain the shortest distances from any node to the two end nodes of link (1,2): $d(1,2) = d(2,1) = 10.0$, $d(3,1) = 6.0$, $d(3,2) = 6.0$, $d(4,1) = 7.0$ and $d(4,2) = 4.0$. The network intersection points on link (1,2) are summarized in Table 3. By definition, we have $N_{fc} = \emptyset$, $N_{uc} = \{3\}$, $N_{pc} = \emptyset$, $N_{vc} = \{1,2,4\}$ and $J = \{0,1.0,3.0,4.0,6.0,7.0,8.0,10.0\}$. Let us examine segment [6.0, 7.0]. It is easy to derive $M_{fc} = \{2\}$, $M_{uc} = \emptyset$ and $M_{pc} = \{1,4\}$. Hence for this segment, $\mathbf{W}(x) = (W_1, W_2, W_4)$ is the random demand weight vector of interest. The elements in the set B are shown in Table 4. Since the antipode with respect to node 4 is $x = 3.5$ on the link, we have $J' = \{6.0,7.0\}$, i.e., [6.0, 7.0] is a primary interval.

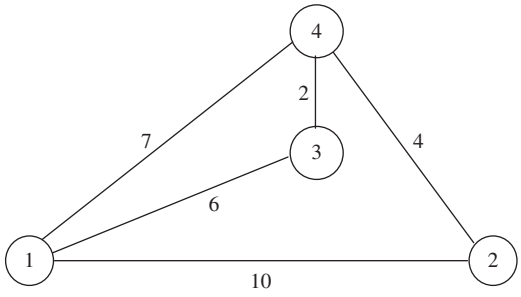

Fig. 1. A 4-node network.

Table 1
Probability distributions of the demand weights.

	k	$w_h[k]$	$p_h[k]$
W_1	1	2	0.3
	2	4	0.7
W_2	1	5	0.6
	2	8	0.4
W_3	1	0	0.2
	2	5	0.8
W_4	1	2	0.4
	2	4	0.4
	3	6	0.2

Table 2
Coverage decay functions.

h	$\hat{r}_h$	$\tilde{r}_h$	α_h	β_h
1	4	8	2	−0.25
2	4	9	1.8	−0.2
3	3	6	2	−1/3
4	7	10	10/3	−1/3

Table 3
Network intersection points.

h	$\hat{y}_h$	$\tilde{y}_h$	$\hat{z}_h$	$\tilde{z}_h$
1	4	8	16	12
2	−6	−1	6	1
3	−3	0	13	10
4	0	3	7	4

Note that for the vector $\mathbf{w}_3 = (2,5,6)$ in the set B, the total demand weight covered by a facility at some point $x \in [6.0,7.0]$ is expressed as $\theta_{\mathbf{w}_3}(x) = 1.5x + 1$. Solving the equation $1.5x + 1 = 11$, we obtain $x = 6.67$ as the critical point with respect to $\mathbf{w}_3$. Since $1.5(6.67 - \varepsilon) + 1 < 11$ and $1.5(6.67 + \varepsilon) + 1 > 11$, we add $x = 6.67$ into the set S_g. A similar analysis on all vectors in the set B results in the set of critical points $S = \{6.0,6.67,7.0\}$ with $S_l = \{7.0\}$, $S_g = \{6.0,6.67\}$, $p_{S_l}(7.0) = 0.112$, $p_{S_g}(6.0) = 0.084$ and $p_{S_g}(6.67) = 0.036$. It is easy to see that $Q = \{(7.0, \mathbf{w}_{10})\}$ and $A = \{\mathbf{w}_5, \mathbf{w}_6, \mathbf{w}_{11}, \mathbf{w}_{12}\}$. The process of identifying and evaluating the dominant critical points within the segment is shown below:

Step 6: Let $q = q' = 0$ and $c = 0.048 + 0.024 + 0.112 + 0.056 + 0.112 = 0.352$.
Step 7: Go to Step 8.
Step 8: $q = 1$. $c = 0.352 + 0.084 = 0.436$. Because $s_1 \in S_g$, $q' = 1$.
Step 9: Go to Step 8.
Step 8: $q = 2$. $c = 0.436 + 0.036 = 0.472$. $q' = 2$.
Step 9: Because s_2 is the last element in the set S_g, go to Step 7.
Step 7: Because $c > f^*$, let $x^* = 0.667$ and $f^* = 0.472$.
Step 8: $q = 3$. Repeat Step 8.
Step 8: $q = 4$. Since $q > |S| = 3$, go to Step 4.

In the above process, $x = 6.67$ is found to be the dominant critical point and an optimal point within the segment.

An optimal solution within any other segment can be obtained in a similar way (see Table 5). Comparing these optimal points, we obtain $x = 6.67$ as an optimal solution on the entire link.

3.3. Piece-wise coverage decay functions

Though not stated explicitly, it is assumed in the previous two sub-sections that the closed form expression of any coverage

Table 4
Elements in the set B for the segment [6.0, 7.0].

m	$\mathbf{w}_m$	$p_{\mathbf{w}_m}$
1	(2,5,2)	0.072
2	(2,5,4)	0.072
3	(2,5,6)	0.036
4	(2,8,2)	0.048
5	(2,8,4)	0.048
6	(2,8,6)	0.024
7	(4,5,2)	0.168
8	(4,5,4)	0.168
9	(4,5,6)	0.084
10	(4,8,2)	0.112
11	(4,8,4)	0.112
12	(4,8,6)	0.056

Table 5
An optimal solution within each segment.

	Optimal point x	$f(x)$
[0,1.0]	0	0
[1.0,3.0]	1.0	0
[3.0,4.0]	3.5	0
[4.0,6.0]	6.0	0.436
[6.0,7.0]	6.67	0.472
[7.0,8.0]	7.0	0.472
[8.0,10.0]	8.0	0.36

decay function $\lambda_h(d(h,x))$ does not change in terms of $d(h,x) \in [\hat{r}_h, \tilde{r}_h]$. Now consider a generalized case where for any node h, $\lambda_h(d(h,x))$ is a continuous piece-wise function with $\eta_h - 1$ break points. Suppose $\lambda_h(d(h,x)) = \lambda_h^{(e)}(d(h,x))$ for any $d(h,x) \in [r_h^{(e-1)}, r_h^{(e)}]$, where $\hat{r}_h = r_h^{(0)} < r_h^{(1)} < \cdots < r_h^{(\eta_h)} = \tilde{r}_h$ and $\lambda_h^{(e)}(d(h,x))$ is a monotone decreasing function with no break points inside the interval $(r_h^{(e-1)}, r_h^{(e)})$.

Since $c_h(x) = \lambda_h(d(h,x))$ if $\hat{r}_h < d(h,x) < \tilde{r}_h$, we note that piece-wise coverage decay functions may also cause functions $\theta_{\mathbf{w}}(x)$ to break along link (i,j). To identify such additional break points, we examine an arbitrarily selected primary interval $[\hat{x}', \tilde{x}']$ defined in Section 3.1. Given node h, define *coverage intersection points* $y_h^{(e)} = r_h^{(e)} - d(h,i)$ and $z_h^{(e)} = d(h,j) + l_{ij} - r_h^{(e)}$ for $e = 1,2,\ldots,\eta_h - 1$. Following our analysis in Section 3.1, we regard $y_h^{(e)}$ and $z_h^{(e)}$ as redundant if $z_h^{(e)} \le y_h^{(e)}$ holds. It turns out that any function $\theta_{\mathbf{w}}(x)$ breaks within the primary interval $[\hat{x}', \tilde{x}']$ at each non-redundant coverage intersection point, if available, associated with some node $h \in N_{pc} \cup M_{pc}$. These break points divide $[\hat{x}', \tilde{x}']$ into non-overlapping sub-intervals. It follows that the closed form expression of any function $\theta_{\mathbf{w}}(x)$ remains the same in each of these sub-intervals. Algorithm 1 can be easily adapted for this case.

4. Normal approximation

For each primary interval $[\hat{x}', \tilde{x}']$ examined in Algorithm 1, the cardinality of the set B and the set S in the worst-case is in the order of K^n, where $K = \max_{h \in N} K_h$. Note that the basic principle underlying Algorithm 1 is to identify, evaluate and compare the critical points in S for each primary interval $[\hat{x}', \tilde{x}']$. We realize that this exact solution procedure will consume lots of CPU time and computer memory when the number of nodes n increases. In this section, a normal approximation approach is developed to solve the problem (P).

Denote by μ_h and σ_h^2, respectively, the mean and variance of the demand weight W_h which are calculated as follows:

$$\mu_h = \sum_{k=1}^{K_h} p_h[k] w_h[k],$$

$$\sigma_h^2 = \sum_{k=1}^{K_h} p_h[k] (w_h[k] - \mu_h)^2.$$

According to the Central Limit Theorem, the total covered demand weight $\sum_{h \in E(x)} c_h(x) W_h$ at a given point x on link (i,j) is approximately a normal random variable with a mean $\mu(x) = \sum_{h \in E(x)} c_h(x) \mu_h(x)$ and a variance $\sigma^2(x) = \sum_{h \in E(x)} c_h(x)^2 \sigma_h^2$ if $|E(x)|$ is sufficiently large. $P(\sum_{h \in E(x)} c_h(x) W_h \ge t)$ can therefore be computed as

$$P\left(\sum_{h \in E(x)} c_h(x) W_h \ge t \right) \approx 1 - \Phi\left[\frac{t - \mu(x)}{\sigma(x)} \right],$$

where $\Phi(\,)$ denotes the cumulative distribution function of the standard normal variable.

Let $\hat{f}(x) = (t - \mu(x))/\sigma(x)$. Since $\Phi(\,)$ is increasing in $\hat{f}(x)$, the problem (P) is equivalent to minimizing $\hat{f}(x)$. For the primary interval $[\hat{x}', \tilde{x}']$ defined in the previous section, divide the set $N_{pc} \cup M_{pc}$ into two subsets N_L and N_R, where

$$N_L = \{h \in N_{pc} \cup M_{pc} \mid d(i,h) + \tilde{x}' \le d(j,h) + l_{ij} - \hat{x}'\},$$

$$N_R = N_{pc} \cup M_{pc} - N_L.$$

It is obvious that the two sets N_L and N_R remain the same and the functional form of $d(x,h)$ for every node $h \in N_{pc} \cup M_{pc}$ does not change within the primary interval. Therefore, the closed form expression of $\hat{f}(x)$ remains unchanged inside the primary interval $[\hat{x}', \tilde{x}']$. We next discuss minimizing the function $\hat{f}(x)$ over such a primary interval. An analytical form of the stationary point for $\hat{f}(x)$ is derived below when the coverage decay functions $\lambda_h(d(h,x)) = \alpha_h + \beta_h d(h,x)$ are linear. A similar approach can be applied when the coverage decay functions are nonlinear.

For a given node $h \in N_{pc} \cup M_{pc}$, we have $d(x,h) = d(i,h) + x$ if $h \in N_L$ and $d(x,h) = l_{ij} + d(j,h) - x$ otherwise. Let $\kappa_h = \alpha_h + \beta_h d(h,i)$ if $h \in N_L$ and $\kappa_h = \alpha_h + \beta_h d(h,j) + \beta_h l_{ij}$ if $h \in N_R$. Given a point $x \in [\hat{x}', \tilde{x}']$, $\hat{f}(x)$ is expressed as

$$\hat{f}(x) = \frac{t - \sum_{h \in N_{fc} \cup M_{fc}} \mu_h - \sum_{h \in N_L} (\kappa_h + \beta_h x) \mu_h - \sum_{h \in N_R} (\kappa_h - \beta_h x) \mu_h}{\sqrt{\sum_{h \in N_{fc} \cup M_{fc}} \sigma_h^2 + \sum_{h \in N_L} (\kappa_h + \beta_h x)^2 \sigma_h^2 + \sum_{h \in N_R} (\kappa_h - \beta_h x)^2 \sigma_h^2}}. \tag{4}$$

Taking the first-order derivative of (4) with respect to x, we have

$$\frac{d\hat{f}(x)}{dx} = \frac{\delta x + \eta}{\left(\sqrt{\sum_{h \in N_{fc} \cup M_{fc}} \sigma_h^2 + \sum_{h \in N_L} (\kappa_h + \beta_h x)^2 \sigma_h^2 + \sum_{h \in N_R} (\kappa_h - \beta_h x)^2 \sigma_h^2} \right)^3}, \tag{5}$$

where

$$\delta = \sum_{h \in N_L} \beta_h \sigma_h^2 \left[\sum_{g \in N_L} (\beta_h \kappa_g - \beta_g \kappa_h) \mu_g + \sum_{g \in N_R} (\beta_h \kappa_g + \beta_g \kappa_h) \mu_g - \beta_h t + \beta_h \sum_{g \in N_{fc} \cup M_{fc}} \mu_g \right]$$

$$+ \sum_{h \in N_R} \beta_h \sigma_h^2 \left[\sum_{g \in N_L} (\beta_h \kappa_g + \beta_g \kappa_h) \mu_g + \sum_{g \in N_R} (\beta_h \kappa_g - \beta_g \kappa_h) \mu_g - \beta_h t + \beta_h \sum_{g \in N_{fc} \cup M_{fc}} \mu_g \right],$$

and

$$\eta = \sum_{h \in N_L} \kappa_h \sigma_h^2 \left[\sum_{g \in N_L} (\beta_h \kappa_g - \beta_g \kappa_h) \mu_g + \sum_{g \in N_R} (\beta_h \kappa_g + \beta_g \kappa_h) \mu_g - \beta_h t + \beta_h \sum_{g \in N_{fc} \cup M_{fc}} \mu_g \right]$$

$$+ \sum_{h \in N_R} \kappa_h \sigma_h^2 \left[\sum_{g \in N_L} (-\beta_h \kappa_g - \beta_g \kappa_h) \mu_g + \sum_{g \in N_R} (-\beta_h \kappa_g + \beta_g \kappa_h) \mu_g + \beta_h t - \beta_h \sum_{g \in N_{fc} \cup M_{fc}} \mu_g \right]$$

$$+ \sum_{h \in N_{fc} \cup M_{fc}} \sigma_h^2 \left(- \sum_{g \in N_L} \beta_g \mu_g + \sum_{g \in N_R} \beta_g \mu_g \right).$$

Equating (5) to 0 gives us a stationary point $x^* = -\eta/\delta$. Since δ and η are constant over the primary interval, the numerator of (5) is a linear function of x. The first-order derivative $d\hat{f}(x)/dx$ changes sign at x^* and therefore x^* must be a minimum or maximum (not an inflection point) as long as x^* is inside the primary interval. The conclusion of the above discussion is that function $\hat{f}(x)$ is unimodal within the sub-interval $[\hat{x}', \tilde{x}']$.

We recommend the following algorithmic approach to solve the problem (P) with linear coverage decay functions using the normal approximation method: divide link (i,j) into non-overlapping primary intervals as defined in Section 3; for each primary interval $[\hat{x}', \tilde{x}']$, $\hat{f}(x)$ at the two end points $\hat{x}$, $\tilde{x}$ and the stationary point x^* (if x^* is an internal point) are calculated and compared to find the minimal solution. By comparing these minima obtained for all the sub-intervals, we identify an optimum point on the link.

Our previous analysis suggests that there are $O(n)$ primary intervals on each link. Hence, $\hat{f}(x)$ is a piece-wise function on a link with $O(n)$ break points. It takes $O(n)$ steps to compute these break points and $O(n \log n)$ steps to sort them. For each primary interval $[\hat{x}', \tilde{x}']$, determining the coverage class for all nodes and computing the stationary point requires $O(n)$ time and $O(n^2)$ time, respectively. It takes $O(n)$ time to evaluate $\hat{f}(x)$ at a given point. Hence, the time complexity of the normal approximation approach is in the order of $O(n^3)$ on every link and $O(n^3 l)$ on the entire network.

5. Computational experience

We conducted a computational experiment to evaluate the solution quality and CPU time of the exact solution procedure and the normal approximation approach for the problem (P) with linear coverage decay functions. Test instances of the problem (P) were randomly generated. We chose the number of nodes $n=5$, 10,15 and 20; the number of links l randomly selected from the set $\{n(n-1)/8, n(n-1)/4, 3n(n-1)/8, n(n-1)/2\}$ or $n-1$, whichever is larger; and the number of realizations of the random demand weight associated with each node $h \in N$ is $K=K_h=2$, 3 and 4. For each combination (n, l, K), 12 instances were randomly generated (in total we had 576 test instances). For each instance, n points were produced at random in a square with 100 units length in each side. These points served as the nodes of the network. The node-distance matrix $\{d_{hj}\}$ was calculated using Euclidean distances. For every node h, $w_h[k]$ values were selected randomly in the set $\{1,2,3,\ldots,10\}$ and $p_h[k]$ values were first generated from the uniform distribution in the interval (0,1) and then normalized. The two coverage radii $\hat{r}_h$ and $\tilde{r}_h$ were randomly generated from the intervals $(0, r_1)$ and (r_1, r_2) for every node h, respectively, where r_1 is the network radius and $r_2 = \max_{i \in N} \max_{x \in G} d(i,x)$ is the longest distance between a node and a point of the network. The value of t for each test instance was randomly generated from the uniform distribution in the interval (t_1, t_2), where t_1 and t_2 (defined in Section 2) were obtained by solving the corresponding deterministic gradual covering problems.

Denote the exact solution procedure and the normal approximation procedure, respectively, by EP and NA. Both procedures were employed to solve the 576 test instances on a Pentium IV PC with 512 MB memory. EP was terminated and the incumbent solution was returned if 10,800 s (3 h) CPU time had been reached. The following performance measures were computed for these two methods:

Hit rate: The proportion of test instances for which the *best solution* was returned. (EP returned the exact optimal solution if it stopped within the specified time limit. Otherwise, NA might obtain the best solution.)

Error: The average relative deviation of the objective value returned by a procedure from the best solution obtained.

Time: The average CPU time in seconds that a procedure took to solve an instance.

A summary of the computational results in terms of n is presented in Table 6. We can clearly see the effect of n on the performance of EP. As n increased, the CPU time required to solve an instance grew dramatically and thus the procedure was often terminated after reaching the time limit. As a consequence, the solution quality of EP deteriorated. We note that the effect of K on the performance of EP shows a similar pattern, but the effect of l is not significant (this observation is based on summaries of the computation results in terms of K and l, which are not shown here).

Table 6
Summary of computational results in terms of n.

Measure	n	EP	NA
Hit rate	5	100%	60.42%
	10	98.61%	78.47%
	15[a]	90.62%	72.92%
	20[b]	75%	100%
Error	5	0	0.0297
	10	3.15×10^{-7}	1.20×10^{-3}
	15[a]	0.0129	5.60×10^{-4}
	20[b]	3.99×10^{-4}	0
Time	5	<1	<1
	10	1098.82	<1
	15[a]	6733.62	<1
	20[b]	3483.33	<1

[a] Instances with $K=4$ were not solvable by EP.
[b] Instances with $K=3$ and 4 were not solvable by EP.

Table 7
Summary of performance measures when $n=15$.

Measure	K	EP	NA
Hit rate	2	100%	85.42%
	3	81.25%	60.42%
Error	2	0	1.02×10^{-3}
	3	0.0259	1.06×10^{-4}
Time	2	27.23	<1
	3	10,800	<1

In the experiment, EP failed to run when $n=15$, $K=4$ and $n=20$, $K=3$ and 4 due to insufficient memory. Though it could run to solve the test instances with $n=20$ and $K=2$, it did not stop within 8 h CPU time for 25% of them. The above observations validated our remark on Algorithm 1 made at the beginning of the previous section. Since the set $B(x)$ at some point x is in the order of K^n, to store and evaluate the critical points in the set S makes EP very demanding with respect to the system memory and CPU time. We therefore conclude that EP is generally not appropriate for instances where K^n is greater than 2×10^7 or n is greater than 20.

NA performed well even for medium-sized instances. Compared to EP, the solution quality of NA improved in general as the problem size increased. Table 7 shows that the Error measure of NA dropped rapidly and that NA and EP became comparable in terms of solution quality as n reached 15. In addition, NA solved all the test instances within 1 s. The experiment suggested that EP became less efficient when $K^n > 3^{15} = 1.43 \times 10^7$. Note that neither the time complexity nor the space complexity of NA is dependent of K^n. Therefore, we recommend the normal approximation method for solving the problem (P) with linear coverage decay functions if the number of nodes n is 15 or above or if K^n reaches the magnitude of 1×10^7. The exact solution approach is recommended for solving smaller instances.

Acknowledgements

The authors would like to thank the referee for her (his) comments and suggestions. Oded Berman and Dmitry Krass were supported by grants from NSERC. Jiamin Wang was supported by a grant from the Faculty Research Committee at the C.W. Post Campus of Long Island University.

References

[1] Berman O, Krass D, Drezner Z. The gradual covering decay location problem on a network. European Journal of Operational Research 2003;35:474–80.
[2] Berman O, Wang J. Probabilistic location problems with discrete demand weights. Networks 2004;44:47–57.
[3] Berman O, Wang J. The 1-median and 1-antimedian problems with continuous probabilistic demand weights. INFOR 2006;44:267–83.
[4] Berman O, Wang J. The probabilistic 1-maximal covering problem on a network with discrete demand weights. Journal of the Operational Research Society 2008;59:1398–405.
[5] Berman O, Wang J, Drezner Z, Wesolowsky GO. The minimax and maximin location problems with uniform distributed weights. IIE Transactions 2003;35:1017–25.
[6] Charnes A, Cooper WW. Deterministic equivalents for optimizing and satisfying under chance constraints. Operations Research 1963;11:18–39.

[7] Church RL, ReVelle C. The maximal covering location problem. Papers of the Regional Science Association 1974;32:101–18.
[8] Church RL, Roberts KL. Generalized coverage models and public facility location. Papers of the Regional Science Association 1984;53:117–35.
[9] Frank H. Optimum locations on a graph with probabilistic demands. Operations Research 1966;14:409–21.
[10] Hakimi SL. Optimal locations of switching centers and the absolute centers and medians of a graph. Operations Research 1964;12:450–9.
[11] ReVelle C, Willimas JC. Reserve design and facility siting. In: Drezner Z, Hamacher ZW, editors. Facility location: applications and theory. Berlin: Springer-Verlag; 2002. p. 307–28.
[12] Simon HA. Models of man. New York: Wiley; 1957. p. 243–73.