SIAM J. DISC. MATH.
Vol. 4, No. 4, pp. 550-567, November 1991

008

OBNOXIOUS FACILITY LOCATION ON GRAPHS*

ARIE TAMIR†

Abstract. This paper discusses new complexity results for several models dealing with the location of obnoxious or undesirable facilities on graphs. The focus is mainly on the continuous p-Maximin and p-Maxisum dispersion models, where the facilities can be established at the nodes or in the interiors of the edges. For the general (nonhomogeneous) case it is shown that both models are strongly NP-hard even when the underlying graph consists of a single edge.

For the homogeneous p-Maximin model it is proven that even the problem of finding a $\frac{2}{3}$-approximation solution is NP-hard, and a polynomial heuristic which provides a $\frac{1}{2}$-approximation to the model is presented. Tree graphs are considered, and new algorithms with lower complexity bounds for several versions of the model are presented.

For the p-Maxisum problem we show that the homogeneous case is NP-hard on general graphs. Turning to the homogeneous case on trees, a certain concavity property is identified and then utilized to improve upon the best known methods to solve this model.

Key words. location theory, obnoxious facilities, network center problems

AMS(MOS) subject classifications. 05C35, 90C27, 68A20, 90B05

1. Introduction. Let $G=(V,E)$ be an undirected graph with node set $V=\{v_1,\cdots,v_n\}$ and edge set E. Let $|E| = m$. Each edge has a positive length and is assumed to be rectifiable. We refer to interior points on an edge by their distances (along the edge) from the two nodes of the edge. Let $A(G)$ denote the continuum set of points on the edges of G. The edge lengths induce a distance function on $A(G)$; for any x, y in $A(G)$, $d(x, y)$ will denote the length of a shortest path connecting x and y. Also, for any subset $Y\subseteq A(G)$, $d(x, Y) = \text{Infimum }\{d(x,y)\,|\, y\in Y\}$.

Let $X=\{x_1,\cdots,x_p\}$ be a finite set of points in $A(G)$. Define the following matrices $D(X,X)$ and $\bar{D}(V,X)$:

$$D(X,X)=\{d(x_i,x_j)\},\qquad 1\leqq i,j\leqq p$$

$$\bar{D}(V,X)=\{d(v_i,x_j)\},\qquad 1\leqq i\leqq n,\quad 1\leqq j\leqq p.$$

Let $f(X)=f(\bar{D}(V,X),D(X,X))$ be a real function which is isotone in the components of the matrices $\bar{D}(V,X)$ and $D(X,X)$. (A real function g defined on R^k is isotone if for any w and z in R^k, $w\leqq z$ implies $g(w)\leqq g(z)$.) A variety of location models in the literature are defined by optimizing various forms of f over classes of subsets $X\subseteq A(G)$, $|X| = p$. For example, the unweighted 1-center of the graph G is obtained by setting $p = 1$ and minimizing the function

$$f(\bar{D}(V,X),D(X,X))=\text{Maximum }\{d(v_i,x_1)\,|\,1\leqq i\leqq n\}$$

over all points x_1 in $A(G)$.

Using location theory terminology, the set X is referred to as the set of new facilities, e.g., suppliers, that must be set up, and the set V is identified as the set of existing facilities, e.g., customers. Conventional or ordinary location models are frequently defined by minimizing an isotone objective f, since the goal is to minimize some function of the distances between all facilities.

In this paper we consider the location of obnoxious or undesirable facilities, e.g., garbage depots and nuclear reactors. Thus, our interest is in studying maximization

* Received by the editors June 26, 1989; accepted for publication (in revised form) July 9, 1990.
† School of Mathematical Sciences, Tel Aviv University, Tel Aviv, Israel.

models with isotone criteria. The reader is referred to the recent papers by Moon and Chaudhry [25] and Erkut and Neuman [11], which survey analytical models and approaches as well as various applications for the location of obnoxious facilities.

In this study we focus on maximizing the following two particular objective functions which seem to be the most popular among researchers interested in obnoxious facility location. The last section of this paper will be devoted to two other related optimization criteria.

Let α_{ij}, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, and β_{ij}, $1 \leqq i, j \leqq p$ be nonnegative weights.

The p-Maximin problem. This model, which has also been labeled as the p-Dispersion model, has the following objective function to be maximized:

$$\begin{aligned} f_1(X) = \text{Minimum}\,\{&\text{Minimum}\,\{\alpha_{ij} d(v_i, x_j) \mid 1 \leqq i \leqq n, 1 \leqq j \leqq p\}, \\ &\text{Minimum}\,\{\beta_{ij}(d(x_i, x_j) \mid 1 \leqq i \neq j \leqq p\}\}. \end{aligned} \tag{1.1}$$

The p-Maxisum problem. This maximization model is defined by the objective

$$f_2(X) = \left\{ \sum_{i=1}^{n} \sum_{j=1}^{p} \alpha_{ij} d(v_i, x_j) + \sum_{i=1}^{p} \sum_{j=1}^{p} \beta_{ij} d(x_i, x_j) \right\}. \tag{1.2}$$

By the homogeneous versions of the above models we refer to the case where the new facilities $\{x_1, \cdots, x_p\}$ provide and receive identical services. Formally, in the homogeneous case $\beta_{ij} = 1$, $1 \leqq i, j \leqq p$, and $\alpha_{ij} = \alpha_i$, for $1 \leqq i \leqq n$ and $1 \leqq j \leqq p$.

2. The p-Maximin problem. The p-Maximin problem is NP-hard when defined on a general graph, even for the homogeneous case in which $\beta_{ij} = 1$, $1 \leqq i \neq j \leqq p$, $\alpha_{ij} = \infty$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$; that is, we want to maximize the minimum distance between new facilities. Note that if each x_i, $1 \leqq i \leqq p$, is further restricted to a node, then the p-Maximin model generalizes the independent set problem [14]. Therefore, the discrete version is also NP-hard. For the continuous case we show that if there exists a polynomial procedure to find an ε-approximation solution with $\varepsilon > \frac{2}{3}$, then $P = \mathrm{NP}$.

PROPOSITION 2.1. *Let ε be a real number satisfying $\varepsilon > \frac{2}{3}$. Let z^* denote the optimal solution value of the* p-Maximin *problem. The problem of finding a set $X = \{x_1, \cdots, x_p\}$ in $A(G)$ such that $f_1(X) \geqq \varepsilon z^*$ is* NP-*hard.* (*X is called an ε-approximation solution.*)

Proof. Consider the case where $\alpha_{ij} = \infty$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, and $\beta_{ij} = 1$, $1 \leqq i \neq j \leqq p$. We reduce the independent set problem on a graph G [14] to the p-Maximin problem. Given an undirected graph $G = (V, E)$, $V = \{v_1, \cdots, v_n\}$, suppose that each edge is of unit length. Consider the graph $G^1 = (V^1, E^1)$, where $V^1 = V \cup \{u_1, \cdots, u_n\}$, $E^1 = E \cup \{(v_1, u_1), \cdots, (v_n, u_n)\}$. For $1 \leqq i \leqq n$ let the length of the edge (v_i, u_i) be $\frac{1}{2}$.

To complete the proof we show that G has an independent set of cardinality p if and only if every ε-approximation solution X for the p-Maximin problem on G^1 satisfies $f_1(X) > 2$. Suppose first that G has an independent set of cardinality p, i.e., there exist $\bar{V} \subseteq V$, $|\bar{V}| = p$, and for each pair of distinct nodes, $v_i, v_j \in \bar{V}$, $d(v_i, v_j) \geqq 2$. Define $V^* \subseteq V^1$ by $V^* = \{u_i \mid v_i \in \bar{V}\}$. Then, if u_i and u_j are two distinct nodes in V^* we have $d(u_i, u_j) \geqq 3$. In particular, the optimal solution to the p-Maximin problem on G^1 is at least 3. Therefore, if X is an ε-approximation with $\varepsilon > \frac{2}{3}$, $f_1(X) \geqq 3\varepsilon > 2$.

Next suppose that $X = \{x_1, \cdots, x_p\}$ is some ε-approximation solution with $f_1(X) > 2$. If x_i, $1 \leqq i \leqq p$, is on some edge (v_t, u_t), $1 \leqq t \leqq n$, replace x_i by u_t. If x_i is on some edge (v_t, v_q), where $d(x_i, v_t) \leqq d(x_i, v_q)$, replace x_i by u_t. Let X^1 be the solution obtained in this process. (Note that by the construction all the elements in X^1 are distinct.)

It is easily seen that $f_1(X^1) > 2$. Therefore, $f_1(X^1) \geqq 3$. Define $\bar{V} \subseteq V$ by

$$\bar{V} = \{v_i \mid u_i \in X^1\}.$$

Then $\bar{V}$ is an independent set of G of cardinality p. □

The above proposition suggests that unless P = NP there is no polynomial ε-approximation algorithm for the homogeneous p-Maximin problem on a general graph with $\varepsilon > \frac{2}{3}$. However, there exists a simple polynomial greedy algorithm which generates a $\frac{1}{2}$-approximation solution. We will consider a slightly more general model for this homogeneous version of the p-Maximin problem.

Let D and S be two nonempty compact subsets of $A(G)$. The homogeneous p-Maximin and p-Minimax problems are defined as follows.

The homogeneous *p*-Maximin model on *D*. Find $X_p = \{x_1, \cdots, x_p\}$, a set of p points in D, such that $\text{Minimum}_{1 \leqq i \neq j \leqq p} \{d(x_i, x_j)\}$ is maximized.

The homogeneous *p*-Minimax model on *D* and *S*. Find $X_p = \{x_1, \cdots, x_p\}$, a set of p points in S, such that $\text{Maximum}_{x \in D} \{d(x, X_p)\}$ is minimized.

Let R_p and r_p denote the optimal solution values of the above two models, respectively.

We also introduce the following related problems. Let $r > 0$.

The *r*-cover problem on *D* and *S*. Find $p = p(r)$, the smallest integer value, and points $x_1, \cdots, x_p$ in S, such that $d(x, X_p) \leqq r$, for every x in D. ($X_p = \{x_1, \cdots, x_p\}$.)

The *r*-anticover problem on *D*. Find $q = q(r)$, the largest integer, and points $x_1, \cdots, x_q$ in D, such that $d(x_i, x_j) \geqq r$, $1 \leqq i \neq j \leqq q$.

The open *r*-anticover problem on *D*. Find $q = q^+(r)$, the largest integer and points $x_1, \cdots, x_q$ in D, such that $d(x_i, x_j) > r$, $1 \leqq i \neq j \leqq q$.

The above types of cover problems generalize and unify several models cited and discussed in the literature, e.g., [11], [20], [25].

LEMMA 2.2. *Let D and S be two compact sets of $A(G)$, and let $p \geqq 1$. Then the solution values to the homogeneous* p-Maximin *and* p-Minimax *models satisfy* $R_{p+1} \leqq 2r_p$.

Proof. Let $r > 0$. Let $X_q = \{x_1, \cdots, x_q\}$, $q = q^+(2r)$, be a solution to the open $2r$-anticover problem, and let $Y_p = \{y_1, \cdots, y_p\}$, $p = p(r)$, be a solution to the r-cover problem. Since $d(x_i, x_j) > 2r$, $1 \leqq i \neq j \leqq q^+(2r)$, we need at least $q^+(2r)$ points in S to ensure a covering of a distance not exceeding r to each point in X_q. Thus, $p(r) \geqq q^+(2r)$, for every $r > 0$.

Let $r = r_p$. Since $p(r_p) \leqq p$ we obtain $q^+(2r_p) \leqq p$. There exist $p + 1$ points in D such that the distance between each pair of distinct points is at least R_{p+1}. If R_{p+1} were strictly greater than $2r_p$, we would have $q^+(2r_p) \geqq p + 1$, contradicting $q^+(2r_p) \leqq p$. Hence, $R_{p+1} \leqq 2r_p$. □

We now introduce a $\frac{1}{2}$-approximation heuristic to the p-Maximin problem on D. This heuristic is motivated by the 2-approximation procedure for the p-Minimax model given in Dyer and Frieze [9]. The idea of the procedure is to construct a sequence of p points such that each point is as far apart from the preceding set of points as possible.

ALGORITHM 2.3

Step 0. Choose an arbitrary point x_1 in D. Let $X_1 = \{x_1\}$.

Step 1. While $j < p$ do

Determine x_{i+1} in D by $d(x_{i+1}, X_i) = \text{Max}_{x \in D} \{d(x, X_i)\}$.

Let $\bar{R}_{i+1} = d(x_{i+1}, X_i)$ and set $X_{i+1} = X_i \cup \{x_{i+1}\}$.

THEOREM 2.4. *Let $X_p = \{x_1, \cdots, x_p\}$ be the set of points generated by Algorithm 2.3. Then X_p is a $\frac{1}{2}$-approximation solution to the p-*Maximin *problem on D, i.e.,*

$$d(x_i, x_j) \geqq \tfrac{1}{2} R_p, \qquad 1 \leqq i \neq j \leqq p.$$

Proof. Let $X_p = \{x_1, \cdots, x_p\}$ be the set of points generated by the algorithm. Consider the subset $X_{p-1} = \{x_1, \cdots, x_{p-1}\}$, and view it as a feasible solution to the $(p-1)$-Minimax problem with $S = D$. By construction, for each x in D, $d(x, X_{p-1}) \leqq d(x_p, X_{p-1}) = \bar{R}_p$. Therefore, $\bar{R}_p \geqq r_{p-1}$.

From Lemma 2.2 we obtain

$$\bar{R}_p \geqq r_{p-1} \geqq \tfrac{1}{2} R_p.$$

Again, by construction, $d(x_i, x_j) \geqq \bar{R}_p$, $1 \leqq i \neq j \leqq p$, and the result follows. □

It should be noted that when G is a tree and $S = A(G)$, the homogeneous p-Maximin problem is equivalent (dual) to the homogeneous $(p-1)$-Minimax problem, [2], [20], [27], more commonly known as the $(p-1)$-center problem. The inequality in Lemma 2.2 holds as an equality. There exist several polynomial algorithms to solve the p-Minimax problem on a tree network for various cases of D. Focusing on the case studied in this paper, i.e., $D = A(G)$, and using the above duality result to solve the $(p+1)$-Maximin model, we can use any of the known algorithms that solve the p-Minimax model, [1], [2], [13], [22], [32]. The algorithms with the known lowest complexity bounds appear in [13], [22]. The algorithm in [22] has an $O(n \log^2 n)$ bound when we implement the improvement in [7], while that of [13] is $O(n \min(p, n) \log(\max(p/n, n/p)))$. Note that the latter bound dominates the former only when $p = O(\log n)$.

The only published algorithm that solves the p-Maximin problem on a tree graph directly appears in [1]. While the algorithms for the p-Minimax problem use a simple $O(n)$ procedure to solve the main subroutine, the r-cover problem, the algorithm in [1] for the p-Maximin relies on an $O(n \log n)$ scheme to solve the r-anticover problem.

Due to the importance of the r-anticover problem, we next present a very simple linear time algorithm to solve this problem on tree graphs. We will consider the following generalization of the r-anticover problem studied also by Moon and Goldman [26].

The generalized anticover problem. Let r and r_i, $1 \leqq i \leqq n$, be a collection of $n+1$ positive numbers. Find $q = q(r)$, the largest number, and points $x_1, \cdots, x_q$ in $A(G)$, such that

$$d(v_i, x_j) \geqq r_i \quad \text{for } 1 \leqq i \leqq n \text{ and } 1 \leqq j \leqq q, \tag{2.1}$$

$$d(x_i, x_j) \geqq r \quad \text{for } 1 \leqq i \neq j \leqq q. \tag{2.2}$$

Moon and Goldman [26] have presented a complicated algorithm to solve the above problem on a tree. However, the exact complexity of their algorithm is not specified. In contrast, our algorithm is quite simple and has a linear complexity.

The first phase of our algorithm identifies the feasible set for the points $\{x_i\}$ induced by the constraints (2.1). It is easy to see that the intersection of this feasible set with any edge of the tree is a segment (subedge) of the edge. Therefore, we characterize in linear time the endpoints in $A(G)$ of all those subedges. We augment all these endpoints to the node set of the tree (note that at most $2(n-1)$ nodes are added), and update the edge set accordingly.

In the second phase we solve an r-anticover problem with the additional supposition that the points $\{x_i\}$ can only be located on a distinguished specified subset of edges.

Tree terminology and notation. Let $T = (V, E)$, $V = \{v_1, \cdots, v_n\}$, be an undirected tree with $n \geqq 3$. We define the following partial ordering on the nodes and edges of T. Suppose that T is rooted at some node which is not a tip (leaf), i.e., its degree is at least two. Without loss of generality let v_1 be the root. Let v_i and v_j be a pair of distinct nodes. v_i is a descendant of v_j if v_j is on the unique path connecting v_i to v_1. Furthermore, if v_i and v_j are also connected by an edge in E we say that v_i is a son of v_j, and v_j is the father of v_i. v_i is the son endpoint of this edge, and v_j is its father endpoint. Note that v_1 has no father and every tip node of T has no sons. For each node v_i, which is not the root, we use $e(i)$ to denote the edge of T connecting v_i to its father.

For each node v_i which is not a tip we let C_i denote its set of sons. If all nodes in C_i are tips then C_i is called a cluster.

If (v_i, v_j) and (v_j, v_k) are two distinct edges in E, and v_j is a father of v_i and a son of v_k, then we say that (v_i, v_j) is a son of (v_j, v_k), and (v_j, v_k) is a father of (v_i, v_j). An edge (v_i, v_j) is called a tip edge if its son endpoint v_i is a tip of the tree.

Phase I. *Identify the feasibility subedges*. Suppose that the tree $T = (V, E)$ is rooted at some node, say v_1. Given the constraints (2.1) our task is to identify for each edge in E, the subedge which is consistent with (2.1). Such a subedge, if it is not empty, will be identified by its two endpoints. Using the father-son relationship induced by the rooted tree, we label these two endpoints as the son and father endpoints, respectively.

The procedure to identify the subedges is based on scanning the tree twice. First, starting with the tips of the tree we recursively compute the son endpoints for all edges. The father endpoints are also computed recursively while starting at the root, scanning all its descendants according to the partial ordering and terminating at the tips.

Step 1. Let v_i be a node in V.
If v_i is a tip of the rooted tree set $\delta_i = r_i$.
If v_i is not a tip set
$\delta_i = \text{Maximum}\ \{r_i, \text{Maximum}\ \{\delta_j - d(v_i, v_j) \mid v_j \in C_i\}\}$.

Step 2. Let v_i be a node in V.
If v_i is the root of the tree set $\varepsilon_i = \delta_i$.
If v_i is not a root let v_j be the father of v_i.
Set $\varepsilon_i = \text{Maximum}\ \{\delta_i, \varepsilon_j - d(v_i, v_j)\}$.

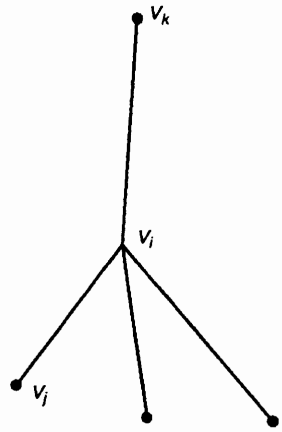
We are now ready to compute the endpoints of all the feasibility subedges. Let (v_i, v_j) be an edge of the tree, where v_j is the father of v_i. If $\delta_i + \varepsilon_j > d(v_i, v_j)$ the feasibility subedge is empty. Otherwise, the son endpoint of the subedge is the point x_i on (v_i, v_j) satisfying $d(v_i, x_i) = \delta_i$, and its father endpoint is the point y_j on (v_i, v_j) satisfying $d(y_j, v_j) = \varepsilon_j$. If $x_i = y_j$ the feasibility subedge is reduced to a point.

It is clear that the above procedure yields the endpoints of all subedges in $O(n)$ total time. We augment the endpoints to the node set of the tree and update the edge set accordingly. In particular, if an original edge has a nonempty subedge it will be replaced by either two or three new edges in the augmented tree. We then solve a modified r-anticover problem on the augmented tree.

Phase II. *The modified r-anticover problem*. This phase corresponds to constraints (2.2). Let $T = (V, E)$ be a rooted tree. Let V^1 be a subset of V and let E^1 be a subset of E. Given a positive real r, find $q = q(r)$, the largest integer, and points $x_1, \cdots, x_q$, where x_i, $1 \leqq i \leqq q$, is in V^1 or is on some edge in E^1, such that $d(x_i, x_j) \geqq r$, for $1 \leqq i \neq j \leqq q$.

ALGORITHM 2.5. The algorithm starts with the rooted tree and recursively processes and eliminates its clusters. (Without loss of generality assume that no node in V^1 is on an edge in E^1.) The main step, the elimination of a cluster, is an improved and modified version of the $O(n \log n)$ procedure presented by Chandrasekaran and Daughety [1] for the standard case, i.e., $E^1 = E$. Our algorithm will run in $O(n)$ time.

Consider a cluster with a father v_i, and let C_i be the set of its sons. (Note that each son of v_i is a tip of the current updated tree.) Let v_k be the father of v_i.



The cluster elimination procedure.

Step 1. For each v_j in C_i apply the following: Suppose first that v_j is in V^1. If $d(v_j, v_i) \geqq r$, add v_j to the set of selected points and remove the edge (v_j, v_i) from the current tree. If $d(v_j, v_i) < r$ set $\lambda_j = d(v_j, v_i)$. Next, suppose that v_j is not in V^1. If the edge (v_j, v_i) is not in E^1, remove it from the current tree. Otherwise, compute

$$t_j = \left[\frac{d(v_j, v_i)}{r} \right].$$

(As usual, $\lfloor a \rfloor$ is the largest integer that is smaller or equal to a.) If $t_j \geqq 1$, select t_j points on the edge (v_i, v_j) as follows: The first point is v_j itself and the others are selected such that the distance between consecutive ones is r. Update the tree by reducing the length of (v_j, v_i) by rt_j. If the modified length λ_j is zero, remove the edge (v_i, v_j) from the current tree.

Step 2. Define $J_1 = \{j \mid \lambda_j < r/2\}$ and $J_2 = \{j \mid r > \lambda_j \geqq r/2\}$. (If both J_1 and J_2 are empty the cluster is eliminated. Stop.) If J_1 is nonempty let $j(1)$ in J_1 be an index satisfying $\lambda_{j(1)} = \text{Maximum}\,\{\lambda_j \mid j \in J_1\}$, and for each $j \in J_1, j \neq j(1)$, remove the edge (v_j, v_i) from the current tree. Set $J_1 = \{j(1)\}$. If J_2 is nonempty let $j(2)$ in J_2 be an index satisfying $\lambda_{j(2)} = \text{Minimum}\,\{\lambda_j \mid j \in J_2\}$.

Step 3. If J_1 is empty, then for each $j \in J_2, j \neq j(2)$, add the tip of the modified edge (v_j, v_i), obtained after the above length reduction, to the set of points already selected and remove that edge from the current tree. Go to Step 5 with $u = v_{j(2)}$ as the only son of v_i, and $d(u, v_i) = \lambda_{j(2)}$. If J_1 is nonempty and J_2 is empty go to Step 5 with $u = v_{j(1)}$ as the only son of v_i, and $d(u, v_i) = \lambda_{j(1)}$. Otherwise, go to Step 4.

Step 4. (J_1 and J_2 are nonempty.) If $\lambda_{j(1)} + \lambda_{j(2)} < r$, remove the edge $(v_{j(1)}, v_i)$ from the current tree, set $J_1 = \phi$ and go to Step 3. If $\lambda_{j(1)} + \lambda_{j(2)} \geqq r$, then for each $j \in J_2$, add the tip of the (modified) edge (v_j, v_i) to the set of points already selected and remove that edge from the current tree. Go to Step 5 with $u = v_{j(1)}$ as the only son of v_i, and $d(u, v_i) = \lambda_{j(1)}$.

Step 5. (u is the only son of v_i, and $d(u, v_i)$ is the length of the edge (u, v_i).) Consider the edge (v_i, v_k), connecting v_i to its father. If (v_i, v_k) is in E^1 or if $d(u, v_i) + d(v_i, v_k) < r$, replace the pair of edges (v_i, v_k), (u, v_i) by a single edge (u, v_i) having the length $d(u, v_i) + d(v_i, v_k)$, and augment this new edge to E^1 if it is not already there. Stop.

If (v_i, v_k) is not in E^1 and $d(u, v_i) + d(u, v_k) \geqq r$, add the point u (the tip of the edge (u, v_i)) to the set of points already selected, and remove the edges (u, v_i) and (v_i, v_k) from the current tree. Stop.

The complexity of the cluster elimination procedure is linear in its size. Therefore, when we apply this procedure recursively to the original tree until we reach its root, it terminates in $O(n)$ time. $q(r)$ is given by the cardinality of the set of points selected in this process.

The validity of the above algorithm follows from the arguments given in Chandrasekaran and Daughety [1] who provided an $O(n \log n)$ algorithm to solve the standard r-anticover problem, i.e., $E^1 = E$.

We note that the above $O(n)$ procedure for solving the generalized r-anticover problem can be implemented, as in [2] and [22], to yield a polynomial algorithm to maximize the following homogeneous p-Maximin problem on a tree $T = (V, E)$.

Find a set of points $X_p = \{x_1, \cdots, x_p\}$ in $A(T)$ maximizing the objective

$$(2.3) \qquad f_1(x_1, \cdots, x_p) = \text{Minimum}\left\{\underset{1 \leqq i \leqq n}{\text{Minimum}}\ \{\alpha_i d(v_i, X_p)\}, \underset{1 \leqq i \neq j \leqq p}{\text{Minimum}}\ \{d(x_i, x_j)\}\right\}.$$

We demonstrate the approach with the case $p = 1$. We show how to use Phase I of the above algorithm to improve the complexity of the best known algorithm to solve the 1-Maximin problem on a tree. Drezner and Wesolowsky [8] have solved this model on a path graph in $O(n^3)$ time. Tamir [30] has presented an $O(n \log n)$ algorithm for paths, and an $O(H(T) \log^2 n)$ algorithm for a general tree graph T, where $H(T)$ is a parameter depending on the topology of the tree. ($H(T)$ is always bounded between n and n^2.)

We now provide an $O(n \log^2 n)$ algorithm, using the above. Let x^* be an optimal solution to the 1-Maximin problem. Then there exists a pair of distinct nodes v_i and v_j of the tree such that x^* is on the unique path connecting v_i and v_j and $\alpha_i d(v_i, x^*) = \alpha_j d(v_j, x^*)$. (Since $p = 1$, we write x^* for x_1^*, and α_i for α_{i1}, $1 \leqq i \leqq n$.) Thus, we have the following proposition.

PROPOSITION 2.6. *Let z^* be the optimal solution value to the* 1-Maximin *problem on a tree. Then z^* is an element in the set R,*

$$(2.4) \qquad R = \left\{\frac{d(v_i, v_j)}{\alpha_i^{-1} + \alpha_j^{-1}} \,\middle|\, 1 \leqq i \neq j \leqq n\right\}.$$

z^* is fully characterized by the following property. Let z be a positive real and consider the problem of finding whether there exists an x in $A(T)$ such that

$$(2.5) \qquad d(v_i, x) \geqq z/\alpha_i, \qquad 1 \leqq i \leqq n.$$

Then, z^* is the largest element in the set R, defined by (2.4), *such that the system* (2.5) *is feasible.*

Given a positive real z, the feasibility of (2.5) can be solved by Phase I above. Setting $r_i = z/\alpha_i$, $1 \leqq i \leqq n$, we note that (2.5) is feasible if and only if Phase I identifies at least one nonempty feasible subedge. Thus, the feasibility of (2.5) can be tested in $O(n)$ time.

With this linear time test we can implement the sophisticated search procedures of [7] and [22] and locate z^* in R in $O(n \log^2 n)$ total time.

We have presented above a polynomial time algorithm to solve the homogeneous p-Maximin problem on a tree. In contrast, the next result shows that the general (nonhomogeneous) case is NP-hard even on a single edge.

PROPOSITION 2.7. *The p-*Maximin *problem is strongly* NP-*hard even when the underlying tree consists of a single edge.*

Proof. Let $G = (V, E)$, $V = \{v_1, \cdots, v_n\}$, be an undirected graph with unit edge lengths. We reduce the Hamiltonian path problem on G [14] to the n-Maximin model on a single edge tree. Let $d(v_i, v_j)$, $1 \leqq i \neq j \leqq n$ be the distance between v_i and v_j on G. Consider the problem of locating a set of n points $\{x_1, \cdots, x_n\}$ on the (real) interval $[0, n-1]$, such that $|x_i - x_j| \geqq d(v_i, v_j)$, $1 \leqq i \neq j \leqq n$.

We claim that G has a Hamiltonian path if and only if the above location problem is feasible. Suppose first that $(v_{i(1)}, \cdots, v_{i(n)})$ indicates the node permutation on a Hamiltonian path. Define the n points on the interval by setting $x_{i(k)} = k - 1$, $k = 1, \cdots, n$. Using the triangle inequality for the distance function on G, we obtain $|x_i - x_j| \geqq d(v_i, v_j)$, for all $1 \leqq i \neq j \leqq n$. Conversely, suppose that $x_{i(1)} < x_{i(2)} \cdots < x_{i(n)}$ indicates the locations of the n points on $[0, n-1]$. By the constraints $x_{i(k)} - x_{i(k-1)} \geqq d(v_{i(k)}, v_{i(k-1)}) \geqq 1$, $k = 2, \cdots, n$, and $x_{i(n)} - x_{i(1)} \leqq n - 1$, we must have $x_{i(k)} - x_{i(k-1)} = 1$, $k = 2, \cdots, n$. Therefore, $(v_{i(1)}, \cdots, v_{i(n)})$ indicates a node ordering of some Hamiltonian path on G. □

Remark 2.8. We have not dealt here with exact algorithms to solve the p-Maximin problem on general graphs. We briefly note several related references and results.

Consider first the homogeneous case of (1.1), where $\alpha_{ij} = \infty$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, $\beta_{ij} = 1$, $1 \leqq i, j \leqq p$. We can use the approaches in Tamir [28] and [29] for the related homogeneous p-Minimax problem, and derive similar results for this case of the p-Maximin model. In particular, we have obtained a result, similar to [29, Thm. 5], identifying a finite set containing the optimal objective value. Exact algorithms to solve the discrete version of this case, where the points selected must be nodes, appear in Erkut [10] and Kuby [21].

Referring next to the general form of (1.1) we note that the model can be solved in polynomial time when p is fixed. We have derived such an algorithm by adopting the approach used in [28] for the p-Minimax problem. We skip the details since the algorithm is practically inefficient due to its high complexity bound, which is exponential in p. For small values of p the algorithm can be significantly accelerated by using recent developments in linear programming. For example, it is shown in Tamir [30] that for $p = 1$, the optimal solution can be obtained in $O(mn)$ time.

Remark 2.9. We have mentioned above that Algorithm 2.3 is based on the heuristic of Dyer and Frieze [9] which generates a 2-approximation solution to the continuous symmetric p-Minimax model. We note in passing that the proof of Proposition 2.1 can be used to show that if there exists a polynomial heuristic to find an ε-approximation solution, with $\varepsilon < \frac{4}{3}$, to the p-Minimax problem, then $P = \mathrm{NP}$. We conjecture that this result actually holds for any $\varepsilon < 2$. Similarly, we conjecture that the result in Proposition 2.1 holds for any $\varepsilon > \frac{1}{2}$.

3. The p-Maxisum problem. We start by showing that the general model is NP-hard even for the trivial case where the graph consists of a single edge. We need the following lemma.

LEMMA 3.1. *Let $T = (V, E)$ be a tree graph. Then there is an optimal solution X^* to the p-*Maxisum *problem, where each $x \in X^*$ is a tip of T.*

Proof. Let X be an optimal solution to the p-Maxisum model. Let $u(X)$ denote the number of points in X that are not tips of T. Among all optimal solutions to the model let X^* have the additional property that $u(X)$ is minimized.

Suppose $\bar{x} \in X^*$ is not a tip of X^*. Fix all points in X^* but $\bar{x}$, and view the objective f_2 as a single variable $(\bar{x})$ function. For every fixed point y on the tree, the function $d(y, \bar{x})$ is convex on every path in T. Thus, $f_2(\bar{x})$ is convex on every path in T. Therefore, its maximum is attained at a tip of T. This contradicts the minimality property of X^*. □

PROPOSITION 3.2. *The* p-Maxisum *problem is* NP-*hard even when* G *is a graph consisting of a single edge.*

Proof. Consider the model with $\alpha_{ij} = 0$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$. From Lemma 3.1 there is an optimal solution X^*, where each $x_i^* \in X^*$ coincides with v_1 or v_2, the two nodes of G. Therefore, the p-Maxisum model reduces to the following Maximum cut problem, which is known to be NP-hard [14]:

$$\text{Find a subset } S \subseteq \{1, 2, \cdots, p\}, \text{ such that } \sum_{i \in S} \sum_{j \notin S} \beta_{ij} \text{ is maximized.} \qquad \square$$

For comparison purposes, it is interesting to note that the minimization of (1.2) on a tree graph can be performed in polynomial time by solving a sequence of $O(n)$ minimum cut problems on a graph with $O(p)$ nodes [19].

We will later show that the homogeneous model can be solved in $O(np)$ time on tree networks. However, on general graphs even the homogeneous model is NP-hard. Hansen and Moon [16] have considered the discrete version of the homogeneous case (with the additional supposition that $\alpha_{ij} = 0$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$), where the points $x_1, \cdots, x_p$ must be selected among the nodes of G. They have shown that the independent set problem [14] on a general graph is reducible to their model. Combining their reduction with the construction in the proof of Proposition 2.1, we obtain the following result for the (continuous) p-Maxisum problem.

PROPOSITION 3.3. *The* p-Maxisum *problem* (1.2) *is* NP-*hard on a general graph even when* $\alpha_{ij} = 0$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, $\beta_{ij} = 1$, $1 \leqq i, j \leqq p$.

Proof. Let $G = (V, E)$, $V = \{v_1, \cdots, v_n\}$, be a graph with unit edge lengths. Let $G' = (V, E')$, be a complete graph with V as its node set. If e is in E let its length be one, otherwise set it equal to two. Extend G' to $G'' = (V'', E'')$ as follows: $V'' = V \cup \{u_1, \cdots, u_n\}$, $E'' = E' \cup \{(v_1, u_1), \cdots, (v_n, u_n)\}$. For $1 \leqq i \leqq n$, let the length of the edge (v_i, u_i) be one. Consider the homogeneous p-Maxisum problem on G'', with all the α-coefficients being equal to zero. It is easy to verify that the graph G has an independent set of cardinality p if and only if the solution value to the above p-Maxisum problem on G'' is equal to $4p(p-1)$. □

When p is fixed (1.2) can be solved in polynomial time on general graphs. Consider first the single facility case, i.e., $p = 1$. Church and Garfinkel [6] have studied this model and provided an $O(n^3)$ algorithm to find the optimal location of the new center x_1. We can improve this bound by using the following observation. Suppose that we restrict the new center to be located on a given edge. Then for each node v_i, $\alpha_{i1} d(v_i, x_1)$ is a concave piecewise linear function on this edge, and it has at most one breakpoint there. Thus, the objective $f_2(x_1)$ is piecewise linear and concave. The maximum point of f_2 on a given edge can be obtained in $O(n)$ time using the recent general algorithms developed by Zemel [35]. Therefore, we conclude that the 1-Maxisum problem on a general graph can be solved in $O(mn)$ time, provided that the distances between all the nodes are given.

The case $p = 2$ can also be solved by using a similar approach. In this case the objective takes on the following form:

$$f_2(x_1, x_2) = \sum_{i=1}^{n} \sum_{j=1}^{2} \alpha_{ij} d(v_i, x_j) + (\beta_{12} + \beta_{21}) d(x_1, x_2). \tag{3.1}$$

When $\alpha_{i1} = \alpha_{i2} = 0$ for $1 \leqq i \leqq n$, the problem reduces to the problem of finding the generalized diameter of a graph. The latter has been solved by Chen and Garfinkel [5]. They have identified a discrete finite set of points that will include at least one optimal solution. Specifically, there is an optimal solution where x_j, $j = 1, 2$, is either a node, or else there exists a node v_i and a simple cycle containing v_i and x_j whose length is $2d(v_i, x_j)$. To maximize (3.1) we solve $O(m^2)$ restricted subproblems. A subproblem is obtained by restricting x_1 and x_2 to be located either on the same edge or on a given pair of edges. In either case we can easily verify that $f_2(x_1, x_2)$ is piecewise linear and concave over the restricted domain. Furthermore, due to the nature of $f_2(x_1, x_2)$, the algorithms in Zemel [35] are applicable in this case and the optimal solution to a subproblem can be obtained in $O(n)$ time. Therefore the global maximizer of (3.1) can be computed in $O(m^2n)$ time.

The above approach can be generalized to yield an $O(m^pn)$ algorithm to solve the p-Maxisum problem (1.2) for every fixed p.

In light of Propositions 3.2 and 3.3 we focus now on the homogeneous case when the graph is a tree. The objective takes on the following form:

$$f_2(X) = \sum_{i=1}^{n} \sum_{j=1}^{p} \alpha_i d(v_i, x_j) + \sum_{i=1}^{p} \sum_{j=1}^{p} d(x_i, x_j). \tag{3.2}$$

Ting [34] has presented an $O(np^2)$ algorithm for this model. Hansen and Moon [16] have studied the case where $\alpha_i = 0$, $1 \leqq i \leqq n$, and restricted the p new facilities to the nodes, allowing no two to be located at the same node. Their algorithm also has the $O(np^2)$ complexity bound.

We improve the above complexity bound by reformulating the objective (3.2). The new formulation will identify useful concavity properties.

Using Lemma 3.1, we may assume that each of the p points $x_1, \cdots, x_p$ is a node of the given tree. For $1 \leqq i \leqq n$, let y_i denote the number of new points (facilities) established at node v_i. The optimization model is now formulated as:

$$\text{Maximize } f_2(y_1, \cdots, y_n)$$

$$\text{subject to}$$

$$f_2(y_1, \cdots, y_n) = \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i d(v_i, v_j) y_j + \sum_{i=1}^{n} \sum_{j=1}^{n} d(v_i, v_j) y_i y_j \tag{3.3}$$

$$\sum_{j=1}^{n} y_j = p, \; y_j \text{ nonnegative and integer, } j = 1, \cdots, n.$$

LEMMA 3.4. *Let $T = (V, E)$ be a tree with nonnegative edge lengths, $\{w_e\}$, $e \in E$. Then the quadratic $f_2(y_1, \cdots, y_n)$ defined by* (3.3) *is concave when restricted to the hyperplane*

$$\left\{ y \mid y \in R^n, \sum_{j=1}^{n} y_j = p \right\}.$$

Proof. It is sufficient to establish the concavity for the quadratic portion of f_2.

Suppose that the tree is rooted at node v_1 and v_1 is not a tip. Consider an edge $e \in E$, and let V^e be the subset of V consisting of all nodes that are disconnected from v_1 by the removal of e. It is easy to verify that w_e, the length of e, will appear exactly

$2(\sum_{\{i|v_i\in V^e\}}\sum_{\{j|v_j\in V-V^e\}} y_iy_j)$ times in the quadratic expression. Therefore,

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d(v_i,v_j)y_iy_j=\sum_{e\in E} 2w_e\left(\sum_{\{i|v_i\in V^e\}}\sum_{\{j|v_j\in V-V^e\}} y_iy_j\right)$$

$$=\sum_{e\in E} 2w_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)\left(\sum_{\{j|v_j\in V-V^e\}} y_j\right).$$

Using the constraint

$$\sum_{j=1}^{n} y_j=p,$$

we obtain

$$\sum_{i=1}^{n}\sum_{j=1}^{n} d(v_i,v_j)y_iy_j=\sum_{e\in E} 2w_e\left(p\sum_{\{i|v_i\in V^e\}} y_i-\left(\sum_{\{i|v_i\in V^e\}} y_i\right)^2\right),$$

which is a concave function. □

The concavity result of Lemma 3.3 can be utilized to yield an $O(np)$ algorithm to solve (3.3). The objective in (3.3) is now formulated as:

$$f_2(y_1,\cdots,y_n)=\sum_{e\in E} w_e\left[\left(\sum_{\{j|v_j\in V-V^e\}}\alpha_j\right)\sum_{\{i|v_i\in V^e\}} y_i+\left(\sum_{\{j|v_j\in V^e\}}\alpha_j\right)\left(p-\sum_{\{i|v_i\in V^e\}} y_i\right)\right]$$

$$+\sum_{e\in E} 2pw_e\sum_{\{i|v_i\in V^e\}} y_i-\sum_{e\in E} 2w_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)^2.$$

Letting

$$\bar{w}_e=w_e\left(\sum_{\{j|v_j\in V-V^e\}}\alpha_j-\sum_{\{j|v_j\in V^e\}}\alpha_j\right)+2pw_e,\qquad e\in E,$$

(3.4) $$f_2(y_1,\cdots,y_n)=\sum_{e\in E}\bar{w}_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)-\sum_{e\in E} 2w_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)^2$$

$$+p\sum_{e\in E} w_e\left(\sum_{\{i|v_i\in V^e\}}\alpha_i\right).$$

Note that the coefficients $\{\bar{w}_e\}$, $e\in E$, can easily be computed in $O(n)$ total time.

To simplify the presentation we augment a node v_0 to the given tree $T=(V,E)$ and connect v_0 to v_1, the current root of T, by an edge which is consistently labeled $e(1)$. We view v_0 as the super root of the augmented tree, $T^1=(V\cup\{v_0\},E\cup\{e(1)\})$.

For each e in E, let $T_e=(V^e,E^e)$ be the subtree induced by V^e. In particular, we note that $T_{e(1)}=T=(V,E)=(V^{e(1)},E^{e(1)})$. We maximize (3.4) recursively by considering its restriction to the subtrees $\{T_e\}$. We will start at the tips of the tree and recursively proceed until we reach the super root of the tree. (To simplify the notation we delete and ignore the constant term in (3.4) while maximizing $f_2(y_1,\cdots,y_n)$.)

For each edge z of the augmented tree T^1 that is not a tip edge define

(3.5) $$f_2(z:y)=\sum_{e\in E^z}\bar{w}_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)-\sum_{e\in E^z} 2w_e\left(\sum_{\{i|v_i\in V^e\}} y_i\right)^2,$$

and

(3.6)

$$F_2(z{:}k) = \text{Maximum}\left\{ f_2(z{:}y) \mid \sum_{\{i \mid v_i \in V^z\}} y_i = k, y_i \text{ nonnegative and integer } 1 \leqq i \leqq n \right\}.$$

The solution value to our model is given by $F_2(e(1){:}p)$.

PROPOSITION 3.5. *For every edge z, $F_2(z{:}k)$ is concave in k, i.e., the difference function $F_2(z{:}k+1) - F_2(z{:}k)$ is monotone nonincreasing.*

Proof. The result follows directly from the concavity of the function $f_2(z{:}y)$. □

ALGORITHM 3.6. To compute the optimal solution value to our model, $F_2(e(1){:}p)$, we give the recursive equations for computing $F_2(z{:}k)$ for all edges z and integer $1 \leqq k \leqq p$.

We use the tree terminology presented above. We start the recursion by defining F_2 for edges z having the property that their son endpoint, say v_i, is such that C_i is a cluster. Consider such an edge $z = e(i)$, and let v_i be its son endpoint. (Recall that for each son v_j of v_i $e(j)$ denotes the edge connecting v_j to v_i.) Then

$$(3.7) \quad F_2(z{:}k) = \text{Maximum}\left\{ \sum_{\{j \mid v_j \in C_i\}} \bar{w}_{e(j)} y_j - \sum_{\{j \mid v_j \in C_i\}} 2 w_{e(j)} y_j^2 \mid \sum_{\{j \mid v_j \in C_i\}} y_j = k, \; y_j \text{ nonnegative integer, } 1 \leqq j \leqq n \right\}.$$

In general, when v_i is a node and C_i is not a cluster, we have the following:

Suppose that v_i is the son endpoint of an edge $e(i)$. Then

$$(3.8) \quad F_2(e(i){:}k) = \text{Maximum}\left\{ \sum_{\{j \mid v_j \in C_i\}} (F_2(e(j){:}k_j) + \bar{w}_{e(j)} k_j - 2 w_{e(j)} k_j^2) \mid \sum_{\{j \mid v_j \in C_i\}} k_j = k, \; k_j \text{ nonnegative integer } 1 \leqq j \leqq n \right\}.$$

Using Proposition 3.5 we note that the maximization defining $F_2(e(i){:}k)$ is a special case of the standard discrete resource allocation model with a separable concave objective function. Using known algorithms for the latter model (see, e.g., [18, Chap. 4]), we conclude that the complexity of computing $F_2(e(i){:}k)$ for all values of $k = 1, \cdots, p$ combined is $O(|C_i| p)$. Therefore, the total effort for computing the optimal solution to the homogeneous p-Maxisum problem on a tree is

$$O\left(p \sum_{i=1}^{n} |C_i|\right) = O(np).$$

The above $O(np)$ algorithm was motivated by the case where p is relatively small. For example, the complexity is linear when p is fixed. We now show that the model is polynomially solvable even when p is a variable integer given as part of the input.

Consider the representation of the objective function in (3.4). For each edge $e(j)$ of the rooted tree define

$$z_j = \sum_{\{i \mid v_i \in V^{e(j)}\}} y_i.$$

The homogeneous p-Maxisum problem is now formulated as an integer quadratic program with a separable objective.

$$\text{Maximize} \sum_{j=1}^{n} \bar{w}_{e(j)} z_j - \sum_{j=1}^{n} 2 w_{e(j)} z_j^2$$

subject to

$$z_j \geqq \sum_{\{i \mid v_i \in C_j\}} z_i, \ 1 \leqq j \leqq n, \text{ and } v_j \text{ is not a tip node,} \tag{3.9}$$

$$z_1 = p,$$

$$z_j \text{ is a nonnegative integer, } 1 \leqq j \leqq n.$$

We note in passing that Lemma 3.1 implies that there is an optimal solution $\{y_1^*, \cdots, y_n^*\}$ to (3.4) where $y_i^* = 0$ if node v_i is not a tip. Therefore, there is an optimal solution $\{z_1^*, \cdots, z_n^*\}$ to (3.9) with

$$z_j^* = \sum_{\{i \mid v_i \in C_j\}} z_i^*, \ 1 \leqq j \leqq n, \text{ and } v_j \text{ is not a tip node.}$$

It is now easy to observe that the constraints in (3.9) represent a flow problem on the tree where p units of a single commodity are to be transferred from the tips of the tree to its root v_1. Therefore, (3.9) is a special case of the flow model discussed in Minoux [24]. Applying his approach to our tree flow problem yields an $O(n^3 \log p)$ algorithm. By implementing more sophisticated data structures we were able to reduce the bound to $O(n^2 \log n \log p)$. For the sake of brevity we skip the details and present, instead, a polynomial algorithm whose bound is independent of p.

Consider first the fractional relaxation obtained from (3.9) by deleting the integrality constraints on the variables. Let $\bar{z}$ be an optimal solution to the fractional relaxation and let τ be the complexity bound to compute $\bar{z}$.

We apply the proximity results in the recent paper by Granot and Skorin-Kapov [15]. Specifically, given $\bar{z}$, the problem of finding the integer solution to (3.9) is now reduced to a flow problem of the same type, where p, the number of units that must flow into the root node, is replaced by some polynomial in n. Thus, the linear constraints, defined by a totally unimodular flow matrix, are independent of p. (Note that p appears only in the linear portion of the objective in (3.4) and (3.9). Therefore, p will appear only in the linear portion of the objective of the reduced problem.) The solution to the reduced integer quadratic program can be obtained by the algorithm in Minoux [24], mentioned above. The running time will depend (polynomially) on n only, i.e., $O(n^2 \log^2 n)$.

To conclude we now have an $O(\tau + n^2 \log^2 n)$ algorithm to solve (3.9), where τ is the complexity bound to compute a solution to the fractional relaxation of (3.9). To solve the latter we apply the algorithm in Chandrasekaran and Kabadi [4]. Since p does not appear in the quadratic portion of the objective, τ is independent of p. It depends (polynomially) on n and the sizes of the edge lengths.

The algorithm in [4] is a general quadratic programming algorithm. We have developed a special purpose, strongly polynomial algorithm to solve the fractional relaxation of (3.9) with $\tau = O(n^2)$. This algorithm is presented in [31].

Remark 3.7. The above algorithm for the (continuous) p-Maxisum problem is based on Lemma 3.1, which limits the search for the optimal solution to the set of tip nodes. This algorithm can easily be modified to the case where the solution is originally confined to any other discrete set of points on the tree. We can even allow upper bounds on the total number of points in $\{x_1, \cdots, x_p\}$ that can be established at each one of the points in the given discrete set. The version of the problem discussed by Hansen and Moon [16] is of that nature. They have confined the p new facilities to the node set, but allow no pair of these to be located at the same node. Therefore, their model can also be solved in $O(np)$ time.

Remark 3.8. The $O(np)$ bound applies to a general tree graph. Improvements are possible for several special cases. For example, suppose that $T = (V, E)$ is a star tree, i.e., T is a cluster where v_1, its root, is the only node which is not a tip. In this case the solution to the problem is given by $F_2(e(1){:}p)$ where $F_2(e(1){:}p)$ is defined by (3.7) for $z = e(1)$ and $k = p$. Therefore, the p-Maxisum problem is reduced to the standard discrete effort allocation problem with a separable concave quadratic objective. The latter problem can be solved in $O(n)$ time [18].

Remark 3.9. For exact algorithms to solve the discrete version of the homogeneous p-Maxisum problem with $\alpha_{ij} = 0$, $1 \leqq i \leqq n$, $1 \leqq j \leqq p$, on a general graph the reader is referred to Erkut, Baptie, and Von Holenbalken [12], Hansen and Moon [16], and Kuby [21]. There are also several related papers cited in [16].

4. Related optimization models. We have studied above the two most common objectives used for locating obnoxious facilities, the p-Maximin and the p-Maxisum criteria. There are several other models mentioned in the literature (see the surveys in [11] and [25]). In this section we discuss briefly two models that we find to be more challenging combinatorially. We report on some of our results and pose a few open problems.

The first model that we consider has been introduced and motivated by Moon and Chaudhry [25]. They have labeled it as the p-Defense problem.

Find points $\{x_1, \cdots, x_p\}$ in $A(G)$ that will maximize the following objective:

$$f_3(x_1, \cdots, x_p) = \sum_{i=1}^{p} \text{Minimum } \{d(x_i, x_j) \mid 1 \leqq j \leqq p, j \neq i\}. \tag{4.1}$$

The second model has been recently suggested by Ting in his Ph.D. dissertation [34].

Find points $\{x_1, \cdots, x_p\}$ in $A(G)$ that will maximize the following objective:

$$f_4(x_1, \cdots, x_p) = \sum_{i=1}^{n} \alpha_i \text{ Minimum } \{d(v_i, x_j) \mid 1 \leqq j \leqq p\} + \sum_{i=1}^{p} \sum_{j=1}^{p} d(x_i, x_j). \tag{4.2}$$

We are not aware of any analytic or algorithmic results for these two models. However, few results and approaches discussed above in §§ 2 and 3 can be modified and applied for models (4.1) and (4.2). For example, both models are NP-hard when defined on general graphs. (The same result holds even when we consider the discrete versions and confine the points $\{x_1, \cdots, x_p\}$ to the node set of the underlying graph.)

Turning to tree graphs, the recursive solution approach of § 3 seems to be applicable for (4.1) and (4.2) as well. However, this recursive approach is discrete in nature, and therefore it requires the optimal points to belong to some prespecified discrete set of points. Moreover, this set must be of polynomial cardinality if we wish the solution procedure to be of polynomial complexity. If such a set is identified we reduce the model to its discrete version by augmenting the points in this set to the node set of the tree. Indeed, we have used the recursive approach and constructed polynomial algorithms of complexity $O(p^2n^3)$ and $O(p^2n^2)$ for the discrete versions of (4.1) and (4.2), respectively.

Therefore, to obtain polynomial procedures for solving the continuous version we need to identify a discrete set that includes at least one optimal solution. Such a set is called a finite dominating set (FDS).

There are several continuous network location models for which an FDS of polynomial cardinality has been found. More recently, Hooker, Garfinkel, and Chen [17], have unified most of these models by identifying common convexity-concavity properties. However, we could not see how to apply their framework to (4.1) and (4.2). Even for tree graphs the objectives in (4.1) and (4.2) do not seem to possess the convexity conditions needed in the general framework of [17]. By using a direct approach we have been able to prove that the set of tip nodes of a tree constitutes an FDS for the following generalization of (4.2):

$$(4.3)\qquad f_5(x_1, \cdots, x_p) = \sum_{i=1}^{n} \alpha_i \text{ Minimum } \{d(v_i, x_j) \mid 1 \leqq j \leqq p\} + \sum_{i=1}^{p} \sum_{j=1}^{p} \beta_{ij} d(x_i, x_j).$$

LEMMA 4.1. *Let* α_i, $1 \leqq i \leqq n$, β_{ij}, $1 \leqq i, j \leqq p$, *be nonnegative numbers. Suppose that* $G = (V, E)$ *is a tree graph. Then there exists an optimal solution* $\{x_1^*, \cdots, x_p^*\}$ *maximizing* $f_5(x_1, \cdots, x_p)$ *such that* x_j^* *is a tip of the tree*, $1 \leqq j \leqq p$.

For the sake of brevity we skip the details of the proof. Unlike the proof of Lemma 3.1, which exhibits a similar result, our proof of Lemma 4.1 is fairly involved. In fact, we have not been able to identify any convexity property, which usually suffices for the existence of a maximum solution at the extreme points. Surprisingly, (4.2) might have isolated maximum solutions which contain some nontip nodes even for the case where the tree is a path connecting a pair of nodes.

Example 4.2. Consider the four node path tree depicted in Fig. 4.1, with $d(v_1, v_2) = d(v_3, v_4) = 1$, and $d(v_2, v_3) = 2$. Let $p = 3$. Let $\alpha_1 = \alpha_4 = 0$, $\alpha_2 = \alpha_3 = 1$, $\beta_{ij} = 1$, $1 \leqq i, j \leqq 3$. An optimal solution satisfying the property in Lemma 4.1 is $x_1 = v_1$, $x_2 = x_3 = v_4$. An isolated optimal solution that contains a nontip node is $x_1 = v_1$, $x_2 = v_4$, and x_3 is the midpoint of the path connecting v_1 and v_4.

Next we turn to the p-Defense problem, defined by (4.1), on tree graphs. As mentioned above, we have constructed a polynomial recursive algorithm to solve the discrete version of the model when the p points are restricted to the node set of the tree. So far we have not been successful in our attempt to obtain an FDS of polynomial cardinality (in n and p) for the continuous problem on a general tree. For a star tree we have identified an FDS of $O(n^3p^2)$ cardinality.

When the graph is a path connecting two tip nodes the solution to the p-Defense model coincides with the unique solution to the homogeneous p-Maximin problem. It has been conjectured that this result holds in general. However, this is not the case even for star trees. First, we observe the following simple result for the p-Defense model.

PROPOSITION 4.3. *Let* $\{x_1^*, \cdots, x_p^*\}$ *be an optimal solution to* (4.1). *Define* $I = \{i \mid x_i^* = x_j^*, \text{for some } 1 \leqq j \leqq p, j \neq i\}$. *Suppose that* I *is nonempty. Then there exists a point* x^* *in* $A(G)$, *called a barrier, such that the set* $\{y_1, \cdots, y_p\}$, *defined by*

$$y_i = \begin{cases} x^* & \textit{if } i \in I, \\ x_i^* & \textit{otherwise} \end{cases}$$

is optimal for (4.1).

FIG. 4.1

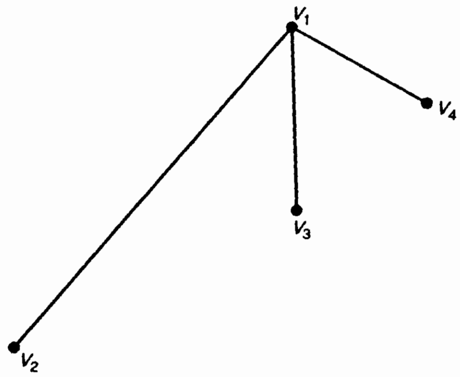


FIG. 4.2

If G is not a path and p is sufficiently large, every optimal solution to the p-Defense problem has a barrier point. No optimal solution to the p-Maximin problem has a pair of points x_i, x_j, $i \neq j$, with $d(x_i, x_j) = 0$. The following example demonstrates that the two problems might have different optimal solutions even if the solution to the p-Defense problem has no barrier point.

Example 4.4. Consider the star tree in Fig. 4.2. Let the edge lengths be $d(v_1, v_2) = 6$, $d(v_1, v_3) = 2$, $d(v_1, v_4) = 1$. The unique solution to the 3-Defense problem consists of the nodes v_2, v_3, and v_4. The unique solution to the 3-Maximin problem consists of v_2, v_3 and the midpoint of the path connecting v_2 and v_3.

5. Summary. We have considered obnoxious facility location on graphs using the p-Maximin and p-Maxisum criteria. These criteria are defined by (1.1) and (1.2), respectively.

For the general (nonhomogeneous) case we have shown that both models are strongly NP-hard even when the underlying graph consists of a single edge.

The other main results for the homogeneous p-Maximin problem are as follows. Unless $P =$ NP there is no polynomial ε-approximation algorithm for the problem on a general graph with $\varepsilon > \frac{2}{3}$. A $\frac{1}{2}$-approximation for the same model is given by the following greedy heuristic. Construct a sequence of p points such that each point is as far apart as possible from the set of points selected before.

Turning to tree graphs we present a linear time algorithm for the (homogeneous) r-anticover problem and apply it to get polynomial time algorithms for the homogeneous p-Maximin problem. For example, we improve upon previous results and solve the single facility case in $O(n \log^2 n)$ time.

For the p-Maxisum problem we have shown that the homogeneous case is strongly NP-hard on general graphs. Focusing on the homogeneous case on tree graphs we have presented an $O(np)$ dynamic programming scheme for its solution. We have then identified useful concavity properties and reformulated the model as a maximum concave separable quadratic flow problem. This formulation has led to polynomial and strongly polynomial algorithms for this homogeneous p-Maxisum problem.

In § 4 we briefly discuss two other models dealing with the location of obnoxious facilities.

Acknowledgment. The current simple and elegant proof of Proposition 2.7 is by Brandon. It improves significantly upon the original proof that was based on a complicated reduction.

REFERENCES

[1] R. CHANDRASEKARAN AND A. DAUGHETY, *Location on tree networks: p-Center and n-Dispersion problems*, Math. Oper. Res., 6 (1981), pp. 50–56.

[2] R. CHANDRASEKARAN AND A. TAMIR, *An* $O((n \log p)^2)$ *algorithm for the continuous p-Center on a tree*, SIAM J. Algebraic Discrete Meth., 1 (1980), pp. 370–375.

[3] ———, *Locating obnoxious facilities*, unpublished report, Tel Aviv University, 1979.

[4] R. CHANDRASEKARAN AND S. KABADI, *Strongly polynomial algorithm for a class of combinatorial LCPs*, Oper. Res. Lett., 6 (1987), pp. 91–92.

[5] C. K. CHEN AND R. S. GARFINKEL, *The generalized diameter of a graph*, Networks, 12 (1982), pp. 335–340.

[6] R. L. CHURCH AND R. S. GARFINKEL, *Locating an obnoxious facility on a network*, Transportation Sci., 12 (1978), pp. 107–118.

[7] R. COLE, *Slowing down sorting networks to obtain faster sorting algorithms*, J. Assoc. Comput. Mach., 34 (1987), pp. 200–208.

[8] Z. DREZNER AND G. O. WESOLOWSKY, *Location of multiple obnoxious facilities*, Transportation Sci., 19 (1985), pp. 193–202.

[9] M. E. DYER AND A. M. FRIEZE, *A simple heuristic for the p-Center problem*, Oper. Res. Lett., 3 (1985), pp. 285–288.

[10] E. ERKUT, *The discrete p-Dispersion problem*, European J. Oper. Res., 46 (1990), pp. 48–60.

[11] E. ERKUT AND S. NEUMAN, *Analytical models for locating undesirable facilities*, European J. Oper. Res., 40 (1989), pp. 275–291.

[12] E. ERKUT, T. BAPTIE, AND B. VON HOLENBALKEN, *The discrete p-Maxisum location problem*, Res. Report 88-4, Dept. of Finance and Management Science, University of Alberta, Canada, 1988.

[13] G. N. FREDERICKSON AND D. B. JOHNSON, *Finding k-th paths and p-Centers by generating and searching good data structures*, J. Algorithms, 4 (1983), pp. 61–80.

[14] M. GAREY AND D. JOHNSON, *Computers and Intractability. A Guide to the Theory of* NP-*Completeness*, W. H. Freeman, San Francisco, 1979.

[15] F. GRANOT AND J. SKORIN-KAPOV, *Some proximity and sensitivity results in quadratic integer programming*, Math. Programming, 47 (1990), pp. 259–268.

[16] P. HANSEN AND D. MOON, *Dispersing facilities on a network*, RRR #52-88, RUTCOR, Rutgers University, 1988.

[17] J. N. HOOKER, R. S. GARFINKEL, AND C. K. CHEN, *Finite dominating sets for network location problems*, Working Paper 19-87-88, GSIA, Carnegie Mellon University, 1988.

[18] T. IBARAKI AND N. KATOH, *Resource Allocation Problems. Algorithmic Approaches*, MIT Press, Cambridge, MA, 1988.

[19] A. W. J. KOLEN, *Location Problems on Trees and in the Rectilinear Plane*, Stitchting Mathematisch Centrum, Amsterdam, 1982.

[20] A. W. J. KOLEN AND A. TAMIR, *Covering problems*, in Discrete Location Theory, R. L. Francis and P. B. Mirchandani, eds., John Wiley, New York, 1990.

[21] M. J. KUBY, *Programming models for facility dispersion: the p-Dispersion and Maxisum dispersion problems*, Geographical Analysis, 19 (1987), pp. 315–329.

[22] N. MEGIDDO AND A. TAMIR, *New results on the complexity of p-Center problems*, SIAM J. Comput., 12 (1983), pp. 751–758.

[23] N. MEGIDDO, A. TAMIR, E. ZEMEL, AND R. CHANDRASEKARAN, *An* $O(n \log^2 n)$ *algorithm for the k-th longest path in a tree with applications to location problems*, SIAM J. Comput., 10 (1981), pp. 328–337.

[24] M. MINOUX, *Solving integer minimum cost flows with separable convex cost objective polynomially*, Math. Programming Stud., 26 (1986), pp. 237–239.

[25] D. MOON AND S. S. CHAUDHRY, *An analysis of network location problems with distance constraints*, Management Science, 30 (1984), pp. 290–397.

[26] D. MOON AND A. J. GOLDMAN, *Tree network location problems with minimum separations*, IIE Trans., to appear.

[27] D. R. SHIER, *A* Min-Max *theorem for p-Center problems on a tree*, Transportation Sci., 11 (1977), pp. 243–252.

[28] A. TAMIR, *A finite algorithm for the continuous p-Center location problem on a graph*, Math. Programming, 31 (1985), pp. 298–306.

[29] ———, *On the solution value of the continuous p-Center location problem on a graph*, Math. Oper. Res., 12 (1987), pp. 340–349.

[30] ———, *Improved complexity bounds for center location problems on networks by using dynamic data structures*, SIAM J. Discrete Math., 1 (1988), pp. 377–396.

[31] ———, *A strongly polynomial algorithm for minimum convex separable quadratic cost flow problems on series-parallel graphs*, Technical Report, Dept. of Statistics and Operations Research, New York University, 1989.

[32] A. TAMIR AND E. ZEMEL, *Locating centers on a tree with discontinuous supply and demand regions*, Math. Oper. Res., 7 (1982), pp. 183–192.

[33] S. S. TING, *A linear time algorithm for maxisum facility location on tree networks*, Transportation Sci., 18 (1984), pp. 76–84.

[34] ———, *Obnoxious facility location problems on networks*, Ph.D. dissertation, The Johns Hopkins University, 1988.

[35] E. ZEMEL, *An $O(n)$ algorithm for the linear multiple choice knapsack and related problems*, Inform. Process. Lett., 18 (1984), pp. 119–121.