DISCUSSION PAPER NO. 358

Polynomially Bounded Algorithms for Locating p-Centers on a Tree

by

R. Chandrasekaran* and A. Tamir

Graduate School of Management
Northwestern University
Evanston, Illinois 60201

December 1978

*University of Texas at Dallas

1. Introduction

In this paper we present an efficient, polynomially bounded algorithm for determining p centers on an undirected tree network using the minmax criterion.

To formulate the general model mathematically assume that an undirected tree T = T(N,A), with N and A denoting the set of all nodes and the set of all arcs respectively, is given. Each arc of T is associated with a positive number called the length of the arc. By a point on T we mean a point along any edge of T. In particular, the nodes of T are points on T. Using the arcs lengths, we define the distance between two points x and y on T as the length of the (unique) path connecting x and y. This distance is denoted by d(x,y).

Two finite sets of points on T, S and D are specified. S is the set of possible locations for supply centers, while D represents the demand points. (Note that neither S nor D are assumed to be subsets of N, the set of original nodes of T.) Each demand point in D is associated with a positive number called the weight. Given a number of supply points, p, the objective is to find p locations in S for the p supply centers, such that the maximum of the weighted distances of the demand points to their respective nearest supply centers is minimized.

Following Hakimi [7], minimax location problems, discrete as well as continuous, on networks, have been studied quite extensively, with emphasis given to the algorithmic aspects. The main results appear in the following list of references: [6, 7, 8, 9, 10, 11, 14].

Focusing on a tree network T = T(N,S), Handler [10], has suggested the categorization scheme $\{{N \atop A}\}/\{{A \atop N}\}/p$, where the first and second cells refer to the possible locations of facilities and demand points respectively, and the third cell indicates the number of supply centers to be established. This scheme identifies a variety of minimax facility location problems in tree networks. For example, A|A|P refers to the problem of locating p centers, where each point on the tree is both a demand point and a potential location of a supply center.

Referring to the above categorization scheme we note that efficient polynomially bounded algorithms have been given for the special cases where p, the number of centers is equal to 1 or 2, [3, 6, 7 8, 9, 10, 14]. The recent work of Hakimi and Kariv [11], provides polynomially bounded algorithms for the models A|N|P and N|N|P. Their work also contains several results on the complexity of minimax location problems on general graphs.

In this work we present a unified model for N|N|P, A|N|P and N|A|P, and give a polynomially bounded algorithm for solving the weighted cases. (Weights are associated only with the models N|N|P and A|N|P, i.e., when the set of demand points is finite.) A polynomial algorithm for A|A|P has been recently developed and reported in [2].

The organization of the paper is as follows. In Section 2, we present graph theoretic results on families of subtrees and neighborhood subtrees. These results are then used in Section 3 to develop an algorithm for the general weighted minimax location problem, described above, with general finite sets, D and S, of demand points and potential location points respectively. This algorithm has a worst case bound which is polynomial in $|N|$, $|D|$, and $|S|$. The bound does not depend on the number of supply centers, p. ($|\cdot|$ denotes the cardinality of a set.) In Section 4 it is shown that the three cases, N|N|P, A|N|P and N|A|P are special cases of the above general framework, with both sets S and D, containing at most $O(|N|^2)$ points. Section 5 focuses on computational aspects of the general algorithm. The last section presents a related problem of locating "mutually obnoxious" facilities

2. Intersection of Trees and Neighborhood Trees

Let $T = T(N,A)$ be a finite undirected tree with node set N and arc set A. Considering a finite set of subtrees of T, $\{T_i\}$ $i = 1,\ldots,m$, define the intersection graph, G, of $\{T_i\}$ as follows: G has m nodes, each corresponding to a different subtree in $\{T_i\}$. Two nodes are then connected by an (undirected) arc if and only if the two corresponding subtrees of T intersect.

Following [1], we note that the intersection graph G is a rigid circuit graph, i.e. each simple cycle of order greater than 3 contains a chord. Such graphs possess the following property due to Dirac [4] and reported also in [1].

Theorem 2.1. Let G be a rigid circuit graph. Then G contains a node u such that u and all its neighboring nodes in G form a clique, i.e. the subgraph defined by u and its neighbor is a maximal complete subgraph of G.

Nodes of G with the above property are called simplicial nodes. Also observe that the rigid circuit property is inherited. Namely, if a node and all its incident arcs are removed from a rigid circuit graph, then the remaining subgraph is also rigid circuit. In particular, this subgraph contains a simplicial node.

Next we prove that given a clique of the intersection graph G, the subtrees corresponding to the nodes of the clique have a common nonempty intersection, which is also a subtree of G.

Lemma 2.2. Let T_1 and T_2 be subtrees of the tree T. Then $T_1 \cap T_2$ and $T_1 \cup T_2$ are also subtrees of T.

Lemma 2.3. Let T_1, T_2 and T_3 be subtrees of the tree T, satisfying, $T_1 \cap T_2 \neq \emptyset$, $T_1 \cap T_3 \neq \emptyset$ and $T_2 \cap T_3 \neq \emptyset$. Then $T_1 \cap T_2 \cap T_3 \neq \emptyset$.

Proof. Suppose that $(T_1 \cap T_3)$ does not intersect $(T_2 \cap T_3)$. Then since T_3 is connected, $S_3 = T_3 - ((T_1 \cap T_3) \cup (T_2 \cap T_3))$ is not empty. For i = 1,2, let A_i be a point in $T_i \cap T_3$. Then there is a simple path P_1 on T_3 connecting A_1 and A_2. P_1 intersects S_3. A_1 and A_2 are also on the tree, $T_1 \cup T_2$, and therefore there exists a simple path P_2 on $T_1 \cup T_2$ connecting those two points. P_2 does not intersect S_3. Hence $P_1 \cup P_2$ contains a cycle-contradicting the tree property of T.

Theorem 2.4. Let $\{T_i\}$, $i = 1,\ldots,m$ be a set of subtrees of the tree T. If $T_i \cap T_j \neq \emptyset$ for all i,j, then $T_1 \cap T_2 \ldots \cap T_m$ is a nonempty tree in T.

Proof. It is sufficient to prove that the intersection is nonempty. The proof is by induction on m. Assume $m \geq 4$. Consider the collection $R = \{T_1, T_2, \ldots, T_{m-2}, \{T_{m-1} \cap T_m\}\}$. From Lemma 2.2 R consists of m-1 nonempty subtrees, while Lemma 2.3 implies that the intersection of each pair in R is nonempty. By the induction hypothesis, the intersection of all of them, i.e. $\bigcap_{i=1}^{m} T_i$, is nonempty.

Next we define neighborhood trees and present several of their properties.

Suppose that the lengths of the arcs of T are given, we define the distance between any two points on T, as the length of the (unique) path connecting them.

Let S be a finite set of points (not necessarily nodes) on T = T(N,A).

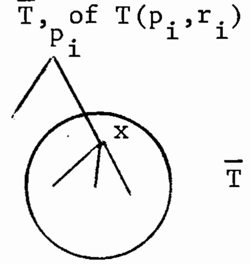
Definition 2.1. Given a point p_i on T, and a number $r_i \geq 0$, the neighborhood tree of radius r_i, with center p_i, is the minimal subtree of T containing p_i and all points x in S with $d(p_i,x) \leq r_i$. This subtree is denoted by $T(p_i,r_i)$.

It is clear that a neighborhood tree $T(p_i,r_i)$ may consist of the center point p_i only. Furthermore, all the tips of $T(p_i,r_i)$, but possibly p_i, provided it is a tip, are points in S.

We now prove that if the intersection of a collection of neighborhood trees, each containing a point in S, is nonempty, then the intersection also contains a point in S.

Lemma 2.5. Let $T(p_i,r_i)$ be a neighborhood tree in T = T(N,A), which contains at least one point in S. Let x be a point in $T(p_i,r_i)$ and define $r = r_i - d(p_i,x)$. Then the neighborhood tree, T(x,r), contained in $T(p_i,r_i)$, has at least one point of S.

Proof. It is clear from the definition that $T(x,r) \subseteq T(p_i,r_i)$. To prove that T(x,r) contains a point of S, consider $T(p_i,r_i)$ as a tree rooted at p_i. Then the subtree, $\bar{T}$, of $T(p_i,r_i)$ rooted at x contains a point of S.



(Otherwise, x would not be in $T(p_i,r_i)$). But $\overline{T} \subseteq T(x,r)$, hence, completing the proof.

Theorem 2.6. For $i = 1,\ldots,m$, let $T(p_i,r_i)$ be a neighborhood tree in T, with radius r_i and center p_i. If $T(p_i,r_i)$, $i = 1,\ldots,m$, contains a point of S and $\bigcap_{i=1}^{m} T(p_i,r_i)$ is not empty, then $\bigcap_{i=1}^{m} T(p_i,r_i)$ contains a point of S.

Proof. Let x be in $\bigcap_{i=1}^{m} T(p_i,r_i)$. For $i = 1,\ldots,m$ define $r_i' = r_i - d(x,p_i)$, and let j be such that $r_j' = \min_{1\le i\le m} r_i'$. Then,

$$T(x,r_j') \subseteq T(x,r_i') \subseteq T(p_i,r_i) \quad \text{for all} \quad i = 1,\ldots,m. \tag{2.1}$$

From Lemma 2.5 $T(x,r_j')$ contains a point in S. (2.1) then implies that this point of S is also in $\bigcap_{i=1}^{m} T(p_i,r_i)$.

3. The General Location Model

Given a finite tree $T = T(N,A)$ with distances on the arcs, a finite set of points, $D \subseteq T$, corresponding to demand points is specified. Also, a finite set of points, $S \subseteq T$ at which supply centers can be located is identified. (Points of D or S are not necessarily original nodes of T. Also D and S may intersect.) Further, there are weights associated with the demand points. Suppose that at most $p < |S|$ supply centers can be established. The objective is then, to find the locations of the supply points, such that the maximum of the weighted distances of the demand points to their respective nearest supply centers is minimized.

We introduce the following notation. Let $D = \{q_1,q_2,\ldots,q_m\}$ be the demand points and let $S = \{s_1,s_2,\ldots,s_k\}$ be the set of potential locations. $w_i > 0$, $i = 1,\ldots,m$ will denote the weight associated with the demand point q_i.

The optimal maximum of the weighted distances of the demand points to their respective nearest supply points is equal to one of the following k.m numbers: $R = \{r_{ij} = w_i d(q_i,s_j),\ i = 1,\ldots,m,\ j = 1,\ldots,k\}$. This latter observation suggests a procedure for the location model described above.

General Scheme

1) For each r in $R = \{r_{ij}\}$ find a subset $Y(r) \subseteq S$ of minimum cardinality such that

$$w_i \min_{s_j \in Y(r)} d(q_i,s_j) \leq r \quad \text{for all } q_i \text{ in } D.$$

2) Denoting by $p(r_{ij})$ the cardinality of the set $Y(r_{ij})$, the optimal location points for the supply centers is given by the set $Y(r_{ij})$ for which r_{ij} is the smallest among all values of r in R satisfying $p(r) \leq p$ (p is the maximum number of supply centers that can be established).

We note in passing that if r^1,r^2 are in R and $r^1 < r^2$, then $p(r^1) \geq p(r^2)$. This monotonicity property enables one to reduce the computational effort required by the above general scheme. A further elaboration will be provided in Section 5.

Next we present a polynomially bounded algorithm, finding the subset $Y(r) \subseteq S$, of minimum cardinality, for an arbitrary $r \geq 0$, such that

$$w_i \min_{s_j \in Y(r)} d(q_i,s_j) \leq r \quad \text{for all } q_i \text{ in } D$$

Algorithm

(1) For each demand point q_i in D, find the neighborhood tree of radius $r_i = r/w_i$, $T(q_i,r_i)$, with respect to the set S of potential location points. If $T(q_i,r_i)$ contains no point of S, stop--the problem is infeasible.

(2) Generate the intersection graph, G, corresponding to the collection of neighborhood trees $\{T(q_i,r_i)\}$, $i = 1,\ldots,m$.

(3) Find the minimum number of cliques covering all the nodes of G.

(4) For each clique found in (3), find a point of S in the intersection of the subset of neighborhood trees corresponding to the nodes of the clique.

Y(r) consists of the points of S specified in (4) for the cliques in the minimum cover. The cardinality of Y(r) is equal to the number of cliques in that cover.

To prove the validity of the algorithm, we first observe that given r the above problem is feasible if and only if each neighborhood tree, $T(q_i,r_i)$ contains at least one point of S. Equivalently, (from Section 2), the problem is infeasible if and only if there exists a demand point q_i which is not a point in S, and $T(q_i,r_i) = \{q_i\}$.

Assuming feasibility, finding Y(r) amounts to identifying the minimum number of points in S required to cover all the neighborhood trees, i.e. each tree $T(q_i,r_i)$ will contain at least one of these S points.

Given a supply point s_j, then the subset of neighborhood trees containing s_j corresponds to a complete subgraph in the intersection graph G. The results of Section 2 prove that we also have the reverse correspondence. More specifically, given a clique of G, Theorems 2.4-2.6 ensure that there exist a point of S, s_j, which is contained in all the neighborhood trees corresponding to the nodes of the clique. Moreover, the maximality of a clique as a complete subgraph shows that s_j is not contained in any tree which is not represented by a node of the clique.

The above discussion has validated the algorithm. We conclude this section by showing that the computational effort of the General Scheme for solving the location problem is bounded by a polynomial in $m = |D|$, $|k| = |S|$, and $n = |N|$, the number of nodes in T(N,A). The General Scheme applies the Algorithm at most k.m times.

In fact, due to the monotonicity of the function p(r) defined in the General Scheme, the Algorithm is applied at most O(log(k.m)) times. Hence, it suffices to show that the latter is polynomially bounded.

It is clear that steps (1), (2) and (4) of the Algorithm can be done in polynomial time. To find the minimum number of cliques covering all the nodes of the rigid circuit graph G, we can use the techniques of [5,13]. There, it is shown how to find the minimum clique cover of a rigid circuit graph in $O(m + e)$ time, where m and e are the numbers of nodes and arcs of G, respectively.

Thus the General Scheme is polynomially efficient.

4. Special Cases

The following location problems on a tree, T = T(N,A), are discussed in the literature, but only a few cases are solved in a polynomial time, ([3, 7, 8, 9, 10, 11, 14]):

I. <u>N|N|P.</u>

In this model the set of demand points, D, and the set of potential location points, S, are identical and equal to N, the set of the original nodes of T(N,A). Given weights on the nodes, the objective is to locate at most p supply centers as to minimize the weighted maximum of the distances to the respective nearest supply points.

II. <u>A|N|P</u>

The only difference between this model and N|N|P, is that here the supply points can be located anywhere on T, i.e. S is the union of all arcs in T.

III. <u>N|A|P</u>

In this problem supply centers can be located only at the nodes of T, i.e. S = N. The set of demand points consists of the whole continuum of points in T. There are no weights associated with the demands points, and the objective is to locate at most p supply points, minimizing the maximum of the distances to the respective nearest supply points.

IV. <u>A|A|P</u>

The only difference between this model and N|A|P is that the supply centers can be established anywhere along the continuum of points of T.

Next we show that the first three models described above are special cases of the general model of Section 3. A polynomially bounded algorithm for the model A|A|P is given in [2].

It is obvious that N|N|P is a special case of the general model since both sets, the demand points and the set of potential locations are finite.

We turn to $A|N|P$ and demonstrate that one may consider only $(\frac{|N|+|N|(|N|-1)}{2})$ potential locations for the supply points. The arguments used are similar to those appearing in [12]. Suppose that a supply point, x, is located at a point of T which is not a node. Then, as in [12], we may assume that there exists nodes of T, q_i and q_j, with weights w_i and w_j respectively, such that $w_i d(q_i,x) = w_j d(q_j,x)$. Observing that $d(q_i,x) + d(q_j,x) = d(q_i,q_j)$, we obtain,

$$d(q_i,x) = w_j \cdot d(q_i,q_j)/(w_i + w_j)$$

Therefore, in addition to the $|N|$ nodes, where a supply center may be located each pair of nodes contributes at most one potential location that one should consider. Hence, the model $A|N|P$ can also be solved by the general model with $|D| = |N|$ and $|S| = |N|(|N| + 1)/2$.

Finally we turn to $N|A|P$. Supply centers can be established only at the nodes of T. We demonstrate that one may assume with no loss of generality that demand points are located only at a finite set of points. In particular, one should only consider the case where demand points are located at the $|N|$ nodes and at the $|N|(|N|-1)/2$ midpoints of the paths connecting pairs of nodes. It is sufficient to show that for any setting of supply centers at nodes, the maximum of the distances of the demand points to their respective nearest supply points is attained for one of the above $|N|(|N|+1)/2$ demand points.

Consider an arbitrary arc (q_i,q_j), and let q_u be the nearest supply center to q_i (see Figure 1). If q_u is also the nearest supply center to q_j, then q_u is the nearest supply point for every point x in (q_i,q_j), with $d(x,q_u) \leq d(q_i,q_u)$. Hence, suppose q_v is the supply center nearest to q_j, (Figure 1). This implies that $d(q_j,q_v) - d(q_i,q_j) < d(q_i,q_u) < d(q_j,q_v) + d(q_i, q_j)$. A simple calculation shows that the function $\min\{d(x,q_v), d(x,q_u)\}$, defined for x in $(q_i q_j)$, is maximized at the midpoint of the path connecting

q_u with q_v. This completes the proof that only $|N|(|N|+1)/2$ demand points should be considered.

Summing up this section, we have demonstrated that the location models $N|N|P$, $A|N|P$ and $N|A|P$ are solvable by the algorithm of Section 3 in a total effort bounded by a polynomial in $|N|$.

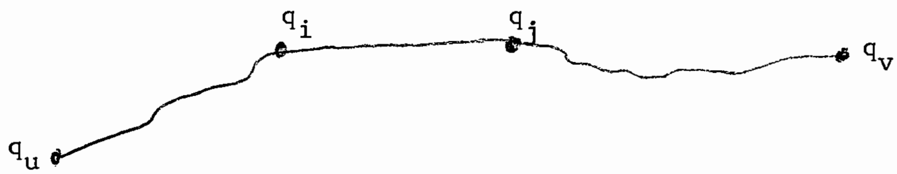


Figure 1

5. Complexity and Computational Efficiency

The initial step of the General Scheme is to compute all the distances on T(N,A) between the m demand points and the k potential location points. Generating this m X k distance matrix is done in total time of $0(m(n+k))$, since finding distances from a given demand point to all points in S is done in $0(|N|+|S|)$ time.

Turning to the algorithm, which is applied at most $0(\log(km))$ times in the General Scheme, we next show how to generate the incidence matrix of G in $0(km)$ time. Two nodes of G are connected by an arc if and only if their corresponding neighborhood trees intersect at a supply point (Theorem 2.6). (We assume that each of the m neighborhood trees contains a supply point.) An $0(m)$ effort will determine all nodes intersecting at a given supply point and therefore, the incidence matrix of G is generated in $0(km)$ time.

As mentioned above, the minimum clique cover of G can now be done in $0(m^2)$ time, using the implementation of the algorithm of [5], as suggested in [13].

Finding the location points on T(N,A) corresponding to the minimum clique cover of G, can certainly be obtained from the information acquired during the construction of the incidence matrix of G. (Recall that each clique of G corresponds to a set of neighborhood trees intersecting at a location point.)

We have thus demonstrated that the worst case time bound for the General Scheme is of order $O(mn + m(k+m)\log(km))$ time. The term $O(m(n+k))$ is the time of the initial step of computing the distances on T(N,A), between the demand points and the potential locations.

Note that the above bound is independent of p, the maximum number of supply centers that can be established. It is, therefore, quite conceivable that certain special cases of our general location model, can be solved more efficiently for small values of p. Indeed, the models $A|A|P = 1,2$ and $A|N|P = 1,2$, with equal weights are solvable in $O(|N|)$ time, see [8, 9, 10, 14].

Remark

Using the notation of Section 3, we observe that p(r) is computed in Step 3 of the algorithm, and is equal to the cardinality of the clique cover. Therefore, step 4 of the algorithm is to be executed only for the optimal value of r in R, i.e. for the smallest r in R satisfying $p(r) \leq p$, (where p is the maximum number of supply centers that can be established.)

Turning to the special models considered in section 4, we now improve the complexity bounds for those cases.

For the weighted $A|N|P$ model we have m=n and $k = O(n^2)$. First we note that two nodes of G, say q_i and q_j, are connected by an arc if and only if the sum of the radii of their respective neighborhood trees, $r_i + r_j = r/w_i + r/w_j$, is not less than $d(q_i, q_j)$. Therefore, the incidence matrix of G, (steps 1-2 of the algorithm) is found in $O(n^2)$ time. Finding the clique cover, step 3 is also done in $O(n^2)$ effort by the general algorithm. Finally, we show that step 4, which is executed only for the optimal value of r, can also be done in $O(n^2)$ time. There, for each clique in the cover, we find one supply center covering the set of demand points corresponding to the clique. Now, using the efficient algorithms for locating one center on a tree, [3,8,9,11], and observing that a demand point, which has already been covered, can be omitted from all remaining cliques, we obtain the bound $O(n^2)$ for step 4. Thus, we have achieved an $O(n^2 \log n)$ time bound for the scheme solving the weighted A|N|P model.

The weighted N|N|P problem can also be solved in $O(n^2 \log n)$ time. Steps 3 and 4 can be done in $O(n^2)$ time like in the A|N|P model. As to steps 1-2, we show that the incidence matrix of G can be generated in $O(n^2)$ time. In fact, since the demand points coincide with the nodes of T, we can easily find all the neighbors in G of a given demand point, by rooting T at the given demand point and then scanning the tree once.

Finally, we turn to the N|A|P model, where the nodes of T are the potential supply points, and only $m = O(n^2)$ demand points should be considered. Again, by rooting T at a given demand point, x, and scanning the tree, all the neighbors of x in G are found in $O(n^2)$ time. Thus, the incidence matrix of G is generated in $O(n^4)$ time. Step 3 will also consume $O(m^2) = O(n^4)$ time, by the general algorithm. Finally, since the distances between all the supply and demand points are known, step 4 is done in $O(n^3)$ time. Therefore, the N|A|P model is solvable in total effort of $O(n^4 \log n)$.

6. Locating "Mutually Obnoxious" Facilities

The following location problem is highly related to the general location model described in Section 3.

Given the tree $T = T(N,A)$ and a finite set of points S in T we wish to place a fixed number of points, p, $p \leq |S|$, in S which are as far apart as possible from one another. We show that this problem of locating "mutually obnoxious" facilities is equivalent, or dual to the problem of locating p-1 centers on T such that the maximum of the distances from the $k = |S|$ points of S to their respective nearest centers is minimized.

Let $\{\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_t\}$, $t = k(k-1)/2$, be the sorted sequence of distances on T(N,A) between all distinct pairs of points in S. Assume that the above sequence contains only $r \leq t$ distinct values, which are then relabeled $W = \{\lambda_1 < \lambda_2 < \ldots < \lambda_r\}$.

Lemma 6.1. Let S_j, $j = 1,\ldots,r$, be a subset of S with maximum cardinality such that the distances between distinct points in S_j is at least λ_j. Denote $N_j = |S_j|$.

Let Q_j, $j = 1,\ldots,r$ be a set of points in T(N,A) of minimum cardinality such that the distances between points of S to their respective nearest points in Q_j is at most $\lambda_j/2$. Denote $P_j = |Q_j|$. Then $N_j = P_{j-1}$, $j = 1,\ldots,2$. (We assume that $\lambda_o = 0$).

Proof. From the previous sections we recall that P_{j-1} is the minimum number of cliques in the optimal clique cover of the nodes of the intersection graph G, corresponding to the $k = |S|$ neighborhood trees of radius $\lambda_{j-1}/2$.

To generate S_j, we first note that two points of S are in S_j if and only if the distance between them is at least λ_j. Since $\lambda_j > \lambda_{j-1}$, a distance between two points of S is at least λ_j if and only if it is greater than λ_{j-1}. Therefore, if G is the above intersection graph corresponding to the $k = |S|$ neighborhood trees of radius $\lambda_{j-1}/2$, then N_j is the cardinality of a maximum cardinality anticlique in G. (An anticlique is a maximal set of nodes in G no two are connected with an arc).

Since G is a rigid circuit graph, we obtain that the cardinality of the largest anticlique is equal to the minimum number of cliques required to cover the nodes of G. Hence $N_j = P_{j-1}$, $j = 1,\ldots,r$.

To introduce our duality result, suppose that $S = \{q_1,q_2,\ldots,q_k\}$.

<u>Theorem 6.2.</u> Given the tree $T = T(N,A)$, the finite subset $S \subseteq T$ and an integer $|S| \geq P > 1$, we have

$$\max_{\substack{U \subseteq S \\ |U|=P}} \{\min\{d(q_i,q_j) | q_i,q_j \in U,\ q_i \neq q_j\}\} =$$

$$= 2 \min_{\substack{V \subseteq T \\ |V|=p-1}} \{ \max_{q_i \in S} \{\min_{x \in V} d(q_i,x)\}\}.$$

<u>Proof</u>. Following [12], we observe that the sets V considered on the right hand side of the above relation can be assumed to be consisted only of the set of midpoints of the different paths connecting pairs of points in S. Hence, the right hand side is equal to λ_i, where λ_i is in W. Also the left hand side of the above relation is equal to λ_j, where λ_j is in W. We prove that $\lambda_j = \lambda_i$.

Using the notation of Lemma 6.1, λ_i is the smallest element in W such that $P_i \leq P-1$, and λ_j is the largest element in W with $N_j \geq P$, i.e. $m < i$ implies $P_m > P-1$, and $m > j$ implies $N_m < P$.

Suppose that $j > i$. Then $i \leq j-1$ and therefore $P_i \geq P_{j-1}$. Applying Lemma 6.1 we obtain the contradiction

$$P \leq N_j = P_{j-1} \leq P_i \leq P-1 .$$

Hence, $j \leq i$. To see that also $j \geq i$, note that $N_i = P_{i-1} > P-1$. Therefore, $N_i \geq P$. But j was defined as the largest element in W with $N_j \geq P$, which in turn yields $j \geq i$. We have thus shown that $i = j$ and therefore $\lambda_j = \lambda_i$. The proof is now complete.

A similar duality result dealing with the continuous problem of locating obnoxious facilities, i.e. S is assumed to be the whole continuum of points in T(N,A), is presented in [14].

Finally, we turn to solving our problem of locating the obnoxious facilities. Referring to the set $W = \{\lambda_1 < \lambda_2, \ldots, < \lambda_r\}$, we have to find the largest λ_j such that $N_j \geq P$. The initial effort of finding the elements in W, or evaluating all the distances between the points in S is done in $O(k(n+k))$, where n is the number of nodes in T(N,A) and $k = |S|$.

Next, given λ_i, finding N_i, is done by computing the largest anticlique on the corresponding intersection graph G. As mentioned in Sections 3-5, the techniques of [5, 12] can be utilized, yielding a bound of order $O(k^2)$ for finding N_i. The monotonicity of N_i in i, i = 1,...,2, enables us to solve the location problem by finding N_i for at most O(log k) values in W. Therefore, the total effort involved in finding the optimal locations is $O(k \cdot n + k^2 \log k)$.

References

1. Buneman, P. "A Characterization of Rigid Circuit Graphs," Discrete Math., 9 (1974), 205-212.

2. Chandrasekaran, R. and A. Daughety. "Problems of Location on Trees," Discussion Paper No. 357, Center for Mathematical Studies in Economics and Management, Northwestern University, 1978.

3. Dearing, P.M. and R.L. Francis. "A Minimax Location Problem on a Network," Transportation Science, 8 (1974), 333-343.

4. Dirac, G.A. "On Rigid Circuit Graphs," Applied Math. Sem., University of Hamburg, 25 (1974), 71.

5. Gavril, F. "Algorithms for Minimum Coloring, Maximum Clique, Minimum Covering by Cliques, and Maximum Independent Set of a Chordal Graph," SIAM J. Comp., 1 (1972), 180-187.

6. Goldman, A.J. "Minimax Location of a Facility in An Undirected Tree Graph," Transportation Science, 6 (1972), 407-418.

7. Hakimi, S.L. "Optimum Locations of Switching Centers and the Absolute Centers and Medians of a Graph," Operations Research, 12 (1964), 450-459.

8. Hakimi, S.L., E.F. Schmeichel and J.G. Pierce. "On P-Centers in Networks," Transportation Science, 12 (1978), 1-15.

9. Handler, G.Y. "Minimax Location of a Facility in an Undirected Tree Graph," Transportation Science, 7 (1973), 287-293.

10. Handler, G.Y. "Finding Two-Centers of a Tree: The Continuous Case," Transportation Science, 12 (1978), 93-106.

11. Kariv, O. and S.L. Hakimi. "An Algorithmic Approach to Network Location Problems" to appear in SIAM J. Appl. Math.

12. Minieka, E. "The m-Center Problem," SIAM Rev., 12 (1970), 138-9.

13. Rose, D.J., R.E. Tarjan and G.S. Lueken. "Algorithmic Aspects of Vertex Elimination on Graphs," SIAM J. Comp., 5 (1976), 266-283.

14. Shier, D.R. "A Min-Max Theorem for Network Location Problems on a Tree," Transportation Science, 11 (1977), 243-252.

the third cell indicates the number of supply centers to be established. This scheme identifies a variety of minimax facility location problems in tree networks. For example, A|A|P refers to the problem of locating p centers, where each point on the tree is both a demand point and a potential location of a supply center.

Referring to the above categorization scheme we note that efficient polynomially bounded algorithms have been given for the special cases where p, the number of centers is equal to 1 or 2, [3, 6, 7 8, 9, 10, 14]. The recent work of Hakimi and Kariv [11], provides polynomially bounded algorithms for the models A|N|P and N|N|P. Their work also contains several results on the complexity of minimax location problems on general graphs.

In this work we present a unified model for N|N|P, A|N|P and N|A|P, and give a polynomially bounded algorithm for solving the weighted cases. (Weights are associated only with the models N|N|P and A|N|P, i.e., when the set of demand points is finite.) A polynomial algorithm for A|A|P has been recently developed and reported in [2].

The organization of the paper is as follows. In Section 2, we present graph theoretic results on families of subtrees and neighborhood subtrees. These results are then used in Section 3 to develop an algorithm for the general weighted minimax location problem, described above, with general finite sets, D and S, of demand points and potential location points respectively. This algorithm has a worst case bound which is polynomial in $|N|$, $|D|$, and $|S|$. The bound does not depend on the number of supply centers, p. ($|\cdot|$ denotes the cardinality of a set.) In Section 4 it is shown that the three cases, N|N|P, A|N|P and N|A|P are special cases of the above

Finding the location points on T(N,A) corresponding to the minimum clique cover of G, can certainly be obtained from the information acquired during the construction of the incidence matrix of G. (Recall that each clique of G corresponds to a set of neighborhood trees intersecting at a location point.)

We have thus demonstrated that the worst case time bound for the General Scheme is of order $0(mn + m(k+m)\log(km))$ time. The term $0(m(n+k))$ is the time of the initial step of computing the distances on T(N,A), between the demand points and the potential locations.

Note that the above bound is independent of p, the maximum number of supply centers that can be established. It is, therefore, quite conceivable that certain special cases of our general location model, can be solved more efficiently for small values of p. Indeed, the models $A|A|P = 1,2$ and $A|N|P = 1,2$, with equal weights are solvable in $0(|N|)$ time, see [8, 9, 10, 14].

Remark

Using the notation of Section 3, we observe that p(r) is computed in Step 3 of the algorithm, and is equal to the cardinality of the clique cover. Therefore, step 4 of the algorithm is to be executed only for the optimal value of r in R, i.e. for the smallest r in R satisfying $p(r) \leq p$, (where p is the maximum number of supply centers that can be established.)

Turning to the special models considered in section 4, we now improve the complexity bounds for those cases.

For the weighted $A|N|P$ model we have m=n and $k = 0(n^2)$. First we note that two nodes of G, say q_i and q_j, are connected by an arc if and only if the sum of the radii of their respective neighborhood trees, $r_i + r_j = r/w_i + r/w_j$, is not less than $d(q_i, q_j)$. Therefore, the incidence matrix of G, (steps 1-2 of the algorithm) is found in $0(n^2)$ time. Finding the

6. Locating "Mutually Obnoxious" Facilities

The following location problem is highly related to the general location model described in Section 3.

Given the tree $T = T(N,A)$ and a finite set of points S in T we wish to place a fixed number of points, p, $p \leq |S|$, in S which are as far apart as possible from one another. We show that this problem of locating "mutually obnoxious" facilities is equivalent, or dual to the problem of locating p-1 centers on T such that the maximum of the distances from the $k = |S|$ points of S to their respective nearest centers is minimized.

Let $\{\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_t\}$, $t = k(k-1)/2$, be the sorted sequence of distances on T(N,A) between all distinct pairs of points in S. Assume that the above sequence contains only $r \leq t$ distinct values, which are then relabeled $W = \{\lambda_1 < \lambda_2 < \ldots < \lambda_r\}$.

Lemma 6.1. Let S_j, $j = 1,\ldots,r$, be a subset of S with maximum cardinality such that the distances between distinct points in S_j is at least λ_j. Denote $N_j = |S_j|$.

Let Q_j, $j = 1,\ldots,r$ be a set of points in T(N,A) of minimum cardinality such that the distances between points of S to their respective nearest points in Q_j is at most $\lambda_j/2$. Denote $P_j = |Q_j|$. Then $N_j = P_{j-1}$, $j = 1,\ldots,2$. (We assume that $\lambda_o = 0$).

Proof. From the previous sections we recall that P_{j-1} is the minimum number of cliques in the optimal clique cover of the nodes of the intersection graph G, corresponding to the $k = |S|$ neighborhood trees of radius $\lambda_{j-1}/2$.

To generate S_j, we first note that two points of S are in S_j if and only if the distance between them is at least λ_j. Since $\lambda_j > \lambda_{j-1}$, a distance between two points of S is at least λ_j if and only if it is greater than λ_{j-1}. Therefore, if G is the above intersection graph corresponding to the $k = |S|$ neighborhood trees of radius $\lambda_{j-1}/2$, then N_j is the cardinality of a maximum cardinality anticlique in G. (An anticlique is a maximal set of nodes in G no two are connected with an arc).

Since G is a rigid circuit graph, we obtain that the cardinality of the largest anticlique is equal to the minimum number of cliques required to cover the nodes of G. Hence $N_j = P_{j-1}$, $j = 1,\ldots,r$.

To introduce our duality result, suppose that $S = \{q_1, q_2, \ldots, q_k\}$.

<u>Theorem 6.2.</u> Given the tree $T = T(N,A)$, the finite subset $S \subseteq T$ and an integer $|S| \geq P > 1$, we have

$$\max_{\substack{U \subseteq S \\ |U| = P}} \{\min\{d(q_i,q_j) \mid q_i, q_j \in U,\ q_i \neq q_j\}\} =$$

$$= 2 \min_{\substack{V \subseteq T \\ |V| = p-1}} \{\max_{q_i \in S} \{\min_{x \in V} d(q_i,x)\}\}.$$

<u>Proof</u>. Following [12], we observe that the sets V considered on the right hand side of the above relation can be assumed to be consisted only of the set of midpoints of the different paths connecting pairs of points in S. Hence, the right hand side is equal to λ_i, where λ_i is in W. Also the left hand side of the above relation is equal to λ_j, where λ_j is in W. We prove that $\lambda_j = \lambda_i$.

Using the notation of Lemma 6.1, λ_i is the smallest element in W such that $P_i \leq P-1$, and λ_j is the largest element in W with $N_j \geq P$, i.e. $m < i$ implies $P_m > P-1$, and $m > j$ implies $N_m < P$.

Suppose that $j > i$. Then $i \leq j-1$ and therefore $P_i \geq P_{j-1}$. Applying Lemma 6.1 we obtain the contradiction

$$P \leq N_j = P_{j-1} \leq P_i \leq P-1 .$$

Hence, $j \leq i$. To see that also $j \geq i$, note that $N_i = P_{i-1} > P-1$. Therefore, $N_i \geq P$. But j was defined as the largest element in W with $N_j \geq P$, which in turn yields $j \geq i$. We have thus shown that i = j and therefore $\lambda_j = \lambda_i$. The proof is now complete.

A similar duality result dealing with the continuous problem of locating obnoxious facilities, i.e. S is assumed to be the whole continuum of points in T(N,A), is presented in [14].

Finally, we turn to solving our problem of locating the obnoxious facilities. Referring to the set $W = \{\lambda_1 < \lambda_2, \ldots, < \lambda_r\}$, we have to find the largest λ_j such that $N_j \geq P$. The initial effort of finding the elements in W, or evaluating all the distances between the points in S is done in $O(k(n+k))$, where n is the number of nodes in T(N,A) and $k = |S|$.

Next, given λ_i, finding N_i, is done by computing the largest anticlique on the corresponding intersection graph G. As mentioned in Sections 3-5, the techniques of [5, 12] can be utilized, yielding a bound of order $O(k^2)$ for finding N_i. The monotonicity of N_i in i, i = 1,...,2, enables us to solve the location problem by finding N_i for at most O(log k) values in W. Therefore, the total effort involved in finding the optimal locations is $O(k \cdot n + k^2 \log k)$.