SIAM J. ALG. DISC. METH.
Vol. 1, No. 4, December 1980

0196-5212/80/0104-0002$01.00/0

AN $O((n \log p)^2)$ ALGORITHM FOR THE CONTINUOUS p-CENTER PROBLEM ON A TREE*

R. CHANDRASEKARAN† AND A. TAMIR‡

Abstract. This paper considers the problem of locating p facilities on a tree network in order to minimize the maximum of the distances of the points on the network to their respective nearest facilities. An $O((n \log p)^2)$ algorithm for a tree network with n nodes is presented.

Introduction. In this study we consider center location problems on undirected tree networks. Let $T = T(N, A)$ be an undirected tree, with N and A denoting the set of all nodes and the set of all arcs respectively. With each arc is associated a positive number called the length of the arc. We assume that T is embedded in the Euclidean plane, so that the arcs are line segments whose endpoints are the nodes, and arcs intersect one another only at nodes. (Any tree with positive arc-lengths can be so embedded in R^2. See [6].) Using this embedding we can then talk about points, not necessarily nodes, on the arcs, and denote by $d(x, y)$ the distance, measured along the arcs of the tree, between any two points x, y of the tree T.

In addition, a set, D, of points on T is specified. D, which may be finite or infinite in cardinality, represents the set of demand points. Assume that supply centers can be located anywhere on the tree. Given a number, p, the objective is to find locations for p supply points on T, such that the supremum of the distances of the demand points in D to their respective nearest supply centers is minimized.

Two special cases of the above model have been treated in the literature. The first corresponds to the case where demand occurs only at the nodes of T, i.e., $D = N$. Whenever $|D| < \infty$, one can also associate weights with the demand points and consider minimizing the maximum of the weighted distances to the nearest supply centers. Efficient, polynomially bounded algorithms when $D = N$ are given in [13], [3] for general p, while further specializations when $p \leqq 2$ are discussed in [6], [8], [9], [10], [11], [14].

The second special case of the general model is the continuous case when $D = T$; i.e., each point of the tree is a demand point. This model is studied in [2], where it is solved in polynomial time.

The general model introduced above is related to the following p-center dispersion problem. A set, S, of points on the tree T is specified. Given an integer p, the objective is to locate p facilities at points in S such that these p facilities are as far from each other as possible.

In this study we focus on the case when the sets D and S in the center location and center dispersion problems, respectively, are identical and equal to the entire tree. Theorem 1 below, (due to Shier [14]), shows a duality result between the p-center location and $(p + 1)$- center dispersion problems, when $D = S = T$. It is convenient for the statement of the theorem to let $U_p = \{u_1, \cdots, u_p\}$ and $V_{p+1} = \{v_1, \cdots, v_{p+1}\}$ denote any finite subsets of T of cardinalities p and $p + 1$ respectively, and to define

$$f_D(U_p) = \max_{x \in D = T} \{\min_{u_i \in U_p} d(x, u_i)\}, \tag{1}$$

* Received by the editors March 19, 1979, and in final form March 28, 1980. This work was done while the authors were visiting Northwestern University.

† University of Texas at Dallas, Dallas, Texas.

‡ Tel-Aviv University, Faculty of Social Sciences, Department of Statistics, Tel-Aviv, Israel.

and

$$g(V_{p+1}) = \min \{d(v_i, v_j)/2 : 1 \leqq i < j \leqq p+1\}. \tag{2}$$

THEOREM 1. [14]. *Let* $D = S = T$. *Then*

$$\min \{f_D(U_p) : U_p \subseteqq T, |U_p| = p\} = \max \{g(V_{p+1}) : V_{p+1} \subseteqq S = T, |V_{p+1}| = p+1\}. \tag{3}$$

We mention that the proof in [14] can be modified to validate the above duality result for the general case when $D = S$ and D is any subset of T with $|D| > p$. (One would have to replace the min and max operations of (1) and (3) by inf and sup, respectively, and also omit the equality of S and D to T in (1) and (3).) Another specialization of the general case, i.e., $D = S$ and $p < |D| < \infty$, is proved in [3], using the equality of the maximum anticlique and the minimum cardinality clique cover in perfect graphs.

Focusing on the subject of this paper, i.e., $D = S = T$, we show that for a given p the minimum value of the objective function of the p-center location problem is equal to $d(i, j)/2k$, where $d(i, j)$ is the distance between some pair of nodes, i and j, of T, and k is an integer satisfying $1 \leqq k \leqq p$. This result is then used to improve the algorithm of [2], yielding the bound of $O((n \log p)^2)$ for the continuous p-center location problem, i.e., $D = T$, on a tree $T(N, A)$ with n nodes. (Logarithms are taken to the base 2.) We also indicate how to improve the $O(n^2 \log n)$ bound of the algorithms of [13], [3] for the discrete p-center location problem, i.e., the case when $D = N$, to obtain an $O(n^2)$ time algorithm.

The continuous p-center problem. In this section we consider the problem of locating p facilities on a tree network in order to minimize the maximum of the distances of the points on the network to their respective nearest facility. Using the notation presented above, we want to find $r(p)$ such that

$$r(p) = \min \{f_T(U_p) : U_p \subseteqq T, |U_p| = p\}, \tag{4}$$

and also the locations for facilities that achieve this value.

Given a point x on T and $r > 0$, we define $N_r(x)$, the r-neighborhood of x, by $N_r(x) = \{y \in T : d(x, y) \leqq r\}$. The location problem is then to find the minimum r such that p r-neighborhoods will cover the entire T. Similarly, given $r > 0$, we consider the reverse problem of covering the tree with a minimum number of r-neighborhoods. This number is denoted by $M(r)$. It is clear that $M(r)$ is a monotone, nonincreasing, step function, which is continuous from the right. $r(p)$ is, therefore, the smallest r such that $M(r) \leqq p$.

The algorithm of [2] for finding $r(p)$ is based on an $O(n)$ subroutine for finding $M(r)$ for an arbitrary $r > 0$. (n is the number of nodes in T.)

In this section we show that $r(p) = d(i, j)/2k$, where $d(i, j)$ is the distance between some pair of tips, i and j, of T, and k is an integer satisfying $1 \leqq k \leqq p$. (A tip is a node of degree 1.) The latter property combined with the monotonicity of $M(r)$ will imply that the $O(n)$ routine for finding $M(r)$ is to be applied at most $O(n^2p)$ times, before $r(p)$ is found.

To prove our claim on $r(p)$ we will need the algorithm of [2] for finding $M(r)$. Thus, for the sake of completeness we describe it here as well.

ALGORITHM 1. Suppose that the tree is rooted at some node and arranged in levels. Define the level of a node as the number of arcs in the unique path connecting the node with the root. Node i is a son of node j if j is the immediate predecessor of i

on the path connecting i with the root. We also say that j is the father of i. Consider a maximal set of tips having the same father, say node s. If all sons of s are tips we call such a set a cluster, and denote it by $C(s)$.

The algorithm will successively eliminate clusters from the tree, where at each iteration it will find the minimum number of supply centers, (r-neighborhoods), required to cover the cluster under consideration.

We start by motivating the first step of the Cluster Elimination Routine. If the length of any arc (s, i)—i being a tip—is greater than $2r$, a facility must be located on (s, i). Without loss of generality, that facility can be established at a point on (s, i) whose distance from the tip i is r. (Note that this facility covers only points on (s, i).) One can then reduce the length of the arc by $2r$.

The Cluster Elimination Routine.

Step 0. Choose a cluster, $C(s)$, of the initial tree, (possibly one of the highest level).

Step 1. Let $\{(s, i)\}$, $i \in C(s)$, be the set of arcs connecting the tips to their predecessor s.

For each i let $d(s, i) = k_i(2r) + b_i$, where k_i is a nonnegative integer and $0 < b_i \leqq 2r$.

$$\text{Set } d(s, i) \leftarrow b_i \quad \text{for } i \in C(s).$$

(At this point k_i facilities have already been established on arc (s, i), with the distance between two adjacent facilities being $2r$. Also note that the trimmed arcs have positive lengths.)

Step 2. Let $\alpha = \min_{i \in C(s)} \{d(s, i) : d(s, i) > r\} = d(s, i_1^*)$,

and

$$\beta = \max_{i \in C(s)} \{d(s, i) : d(s, i) \leqq r\} = d(s, i_2^*).$$

In case of a tie i_1^* (i_2^*) can be chosen as the smallest index for which the minimum (maximum) is attained. Also, if $\alpha(\beta)$ is defined on an empty set it is set equal to $+\infty(-\infty)$. (Note that at least one of α, β is finite.)

(i) If $\alpha + \beta > 2r$, then for each i such that $d(s, i) > r$, locate a facility on (s, i) at a distance r from the tip i (of the reduced cluster obtained in Step 1). Remove each arc (s, i) in $C(s)$ except (s, i_2^*).

If s is the root of the tree, locate a facility at s and terminate. Otherwise remove node s so that we have the case shown in Fig. 1, and go to Step 3.

(ii) If $\alpha + \beta \leqq 2r$, then for each $i \neq i_1^*$ with $d(s, i) > r$, locate a facility on (s, i) at a distance r from the tip i. Remove all the arcs (s, i) except (s, i_1^*).

If s is the root of the tree, locate a facility on (s, i_1^*) at a distance r from i_1^* and terminate. Otherwise, remove node s as shown in Fig. 1, and go to Step 3.

Step 3. Choose a cluster of the remaining tree (possibly one of the highest level), and return to Step 1.

It is clear that the above algorithm takes $O(\max(n, M(r)))$ time, if the output is to be the $M(r)$ facility locations. However, the following method of recording the output reduces the time bound to $O(n)$. On an arc, if there are k facilities to be located at a distance $2r$ from each other, the location of only the first one and their number may be output.

THEOREM 2. *Let $r(p)$ be the solution to the continuous p-center problem, i.e. $r(p)$ is defined by* (4). *Then $r(p) = d(i, j)/2k$, where $d(i, j)$ is a distance between a pair of tips, i and j, of the tree T, and k is an integer, $1 \leqq k \leqq p$.*

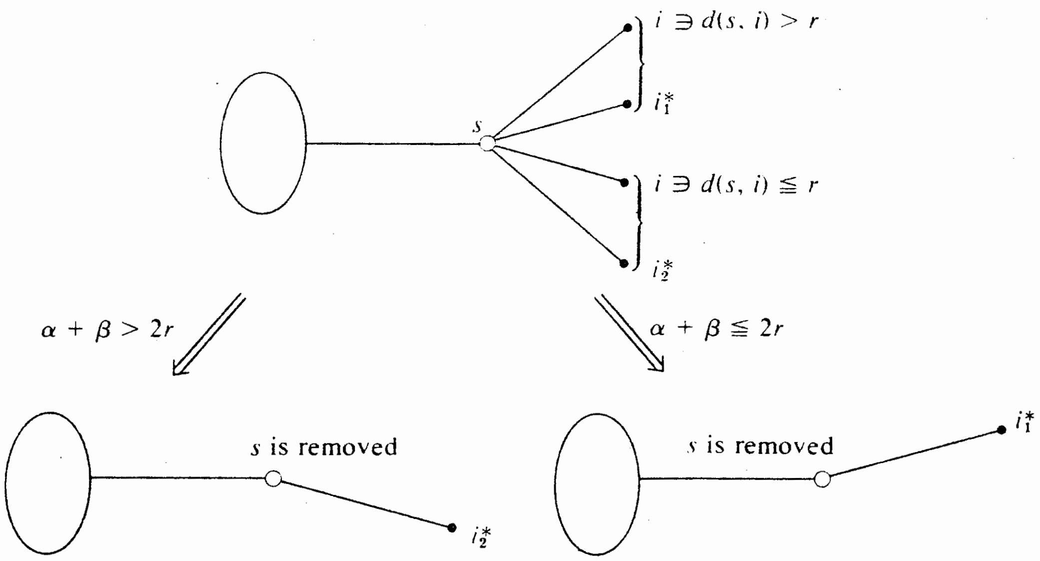

FIG. 1

Proof. Let $S = \{x_1, \cdots, x_p\}$ be the set of points on T at which the p optimal supply centers are located. Define $D = \{y : y \in T, \min_{1 \leqq i \leqq p} d(y, x_i) = r(p)\}$, and let S' be the subset of supply points serving the members of D, i.e.,

$$S' = \{x : x \in S, d(x, y) = r(p) \text{ for some } y \in D\}.$$

First we claim that without loss of generality it can be assumed that each member of S' is the midpoint of a simple path of length $2r(p)$, connecting two points of D. Suppose that $x \in S'$ does not have the above property. Then, the supply center there can be slightly perturbed to x' such that the optimality is not affected; all points y in D served by x satisfy $d(x', y) < r(p)$, and no additional points are added to D. Therefore, x can be omitted from S' and all points y in D served by x can be omitted from D. Note that the minimality of $r(p)$ ensures that the set S' remaining after this process is not empty.

To complete the proof of the theorem we show that each member of D which is not a tip of T must be the midpoint of a simple path of length $2r(p)$, connecting two points of S'.

Let $y \in D$. Then there exists $x_i \in S'$ with $d(y, x_i) = r(p)$. If y is not a tip there exists $z \neq y$, $z \in T$, and y is on the simple path between z and x_i. Considering only the subpath connecting z and y, we observe that all points on this subpath but y are not served by x_i, since they are at a distance greater than $r(p)$ from x_i. So, let x_k be the point in S, closest to y, and serving at least one point which is not y, on the above subpath. Clearly $d(y, x_k) = r(p)$, since y is in D, and therefore x_k is in S'.

Moreover, since $d(x_k, u) \leqq r(p)$ for some $u \neq y$ on that subpath, y is the only intersection point of the path connecting y and x_k and the path connecting y and x_i. Hence y is on the simple path between x_i and x_k with $d(y, x_i) = d(y, x_k) = r(p)$.

Using the above properties satisfied by the members of D and S', we start with x in S' and consider the path of length $2r(p)$, which connects two points of D and has x as its midpoint. If at least one of these endpoints is not a tip, the path can be extended by $2r(p)$ such that the new path will still connect two members of D. Continuing this process, the no-cycle property of a tree ensures that we find a simple path of the tree connecting two tips and having total length of $2kr(p)$, $1 \leqq k \leqq p$. This completes the proof.

The above theorem implies that $r(p)$, the solution to the p-center problem, can be found by applying Algorithm 1 $O(n^2p)$ times, thus yielding an $O(n^3p)$ bound for solving the continuous p-center problem. Next we show a reduction of this bound which is based on the nature of the $O(n^2p)$ possible values for $r(p)$.

Due to the monotonicity property of $M(r)$, found by Algorithm 1, it is clear that if $M(\bar{r}) \leqq p$ then $r(p) \leqq \bar{r}$, and one can ignore all values of r greater than $\bar{r}$. Similarly, if $M(\bar{r}) > p$ we have $r(p) > \bar{r}$. Let R be the set of possible values for $r(p)$ as specified by Theorem 2. We start by finding the median of R, say r_1, and then applying Algorithm 1 to find $M(r_1)$. Comparing $M(r_1)$ and p we then eliminate half of the members of R from further consideration, leaving the subset R_1. We then continue by finding the median of R_1, say r_2, computing $M(r_2)$, and so on. Let r_i denote the median found at the ith iteration and let R_i be the respective subset of R that we are left with at this iteration. Next we show that the total effort of evaluating the sequence of medians $\{r_1, r_2, \cdots\}$ is $O(n^2 \log^2 p)$.

First, an effort of $O(n^2)$ yields the distances between all tips of T. For each such distance $d(i, j)$ the sequence $\{d(i, j)/2k\}$, $k = 1, \cdots, p$, is a monotone decreasing sequence. One can then apply the methods of [7], [12] to find r_1 in $O(n^2 \log p)$ time. Applying Algorithm 1 to r_1 (for $O(n)$ time), we can then use a binary search on each one of the sequences $\{d(i, j)/2k\}$, $k = 1, \cdots, p$, to find R_1. Since there are n^2 sequences, this effort amounts to $O(n^2 \log p)$. In general, at the ith iteration, two pointers are sufficient to limit that part of a sequence $\{d(i, j)/2k\}$, $k = 1, \cdots, p$ which is contained in R_i. Hence the storage requirement is of order $O(n^2)$. Successive applications of the methods of [7], [12] for $q = \log p$ times will yield $r_1, r_2, \cdots, r_q$. By that time the remaining set of possible values, R_q, will contain $O(n^2)$ elements. Therefore, the remaining medians in the sequence are found in total effort of $O(n^2)$ using the linear time algorithm of [1]. Thus, we have demonstrated that the total effort of our procedure to find $r(p)$ is of order $O(n^2 \log^2 p)$ with $O(n^2)$ storage.

Finally, using the duality result presented in the Introduction we observe that the optimal objective value of the p-center dispersion problem is also found in $O(n^2 \log^2 p)$ time. To find the locations of the p centers achieving this optimal value, one can use the procedure given in [2]. As shown in [5] this procedure can be implemented in $O(n^2)$ time.

Remarks.

1) There are certain circumstances where the bound $O(n^2 \log^2 p)$ given above can be improved if a different method is used to find the sequence of medians. We mention two such procedures. The first one is based on the observation that the median of the set R is also the median of the set R^{-1}, consisting of the reciprocals of R. But then the sequence $\{2k/d(i, j)\}$ $k = 1, \cdots, p$, is a linear sequence. It is shown in [4] how to find a median of set consisting of n^2 linear sequences in $O(n^2 \log n)$ time. Applying the latter procedure to compute $\{r_1, r_2, \cdots\}$ yields the bound $O(n^2 \log n \log p)$ for the algorithm to find $r(p)$.

For the second procedure we first sort the sequence of the $m = O(n^2)$ distances between the tips. Denoting this sorted sequence by $c_1 \geqq c_2 \cdots \geqq c_m$, we represent R as the union of p monotone sequences. For each $k = 1, \cdots, p$ we consider the sequence $\{c_i/2k\}$, $i = 1, \cdots, m$. Applying the methods of [7], [12] to this structure yields the bound $O(n^2 \log n + p \log n \log p)$ for the total effort to find $r(p)$.

2) The discrete p-center problem, i.e. the model where demand occurs only at the nodes of T, is solved in [13], [3] by an $O(n^2 \log n)$ algorithm. We indicate that this bound can be reduced to $O(n^2)$ for the method in [13]. The set R of possible values for $r(p)$ for the discrete problem is known to contain $O(n^2)$ elements. All these elements are computed in $O(n^2)$ total effort. Then, for each given r, an $O(n)$ routine finding $M(r)$, the minimum number of r-neighborhoods covering all nodes, is given.

As was done above for the continuous p-center problem, one can generate the sequence of medians $\{r_1, r_2, \cdots\}$ and apply the procedure to find $M(r)$ a total of $O(\log(n^2)) = O(\log n)$ times. Since each time the cardinality of the remaining set R_i is cut by half, the linear time algorithm of [1] will generate the entire sequence of medians in total effort of $O(n^2)$. This latter term is then the dominating term yielding the bound $O(n^2)$ for the effort to find $r(p)$.

REFERENCES

[1] M. BLUM, R. W. FLOYD, V. R. PRATT, R. L. RIVEST, AND R. E. TARJAN, *Time bounds for selection*, J. Comput. System Sci., 7 (1973), pp. 448–461.

[2] R. CHANDRASEKARAN AND A. DAUGHETY, *Problems of location on trees*, Discussion Paper No. 357, Center for Mathematical Studies in Economics and Management, Northwestern University, 1978.

[3] R. CHANDRASEKARAN AND A. TAMIR, *Polynomially bounded algorithms for locating p-centers on a tree*, Discussion Paper No. 358, Center for Mathematical Studies in Economics and Management, Northwestern University, 1978.

[4] ———, *Optimizing over nested linear constraints*, in preparation.

[5] ———, *Locating obnoxious facilities*. Manuscript, Department of Statistics, Tel Aviv University, 1979.

[6] P. M. DEARING AND R. L., FRANCIS, *A minimax location problem on a network*, Transportation Sci., 8 (1974), pp. 333–343.

[7] Z. GALIL AND N. MEGIDDO, *A fast selection algorithm and the problem of optimum distribution of effort*, J. Assoc. Comput. Mach., 26 (1979), pp. 58–64.

[8] A. J. GOLDMAN, *Minimax location of a facility in a network*, Transportation Sci., 6 (1972), pp. 407–418.

[9] S. L. HAKIMI, E. F. SCHMEICHEL, AND J. G. PIERCE, *On p-centers in networks*, Transportation Sci., 12 (1978), pp. 1–15.

[10] G. Y. HANDLER, *Minimax location of a facility in an undirected tree graph*, Transportation Sci., 7 (1973), pp. 287–293.

[11] ———, *Finding two-centers of a tree: The continuous case*, Transportation Sci., 12 (1978), pp. 93–106.

[12] D. B. JOHNSON AND T. MIZOGUCHI, *Selecting the Kth element in $X + Y$ and $X_1 + X_2 + \cdots + X_m$*, SIAM J. Comput., 7 (1978), pp. 147–153.

[13] O. KARIV AND S. L. HAKIMI, *An algorithmic approach to network location problems. Part I: the p-centers*, SIAM J. Appl. Math., 37 (1979), pp. 513–538.

[14] D. R. SHIER, *A min-max theorem for p-center problems on a tree*, Transportation Sci., 11 (1977), pp. 243–252.