Chandrasekaran, R.; Daughety, A.

Working Paper

Location on Tree Networks: P-Centre and n-Dispersion Problems

Discussion Paper, No. 357R

Provided in Cooperation with:
Kellogg School of Management - Center for Mathematical Studies in Economics and Management Science, Northwestern University

Suggested Citation: Chandrasekaran, R.; Daughety, A. (1978) : Location on Tree Networks: P-Centre and n-Dispersion Problems, Discussion Paper, No. 357R, Northwestern University, Kellogg School of Management, Center for Mathematical Studies in Economics and Management Science, Evanston, IL

This Version is available at:
https://hdl.handle.net/10419/220717

DISCUSSION PAPER NO. 357R

Location on Tree Networks: P-Centre and n-Dispersion Problems

by

R. Chandrasekaran*

A. Daughety**

December, 1978

Revised October, 1979

+The authors wish to thank A. Tamir, E. Zemel and two anonymous referees for their comments and suggestions without necessarily implicating them in any of the remaining errors.

*University of Texas at Dallas

**Northwestern University

ABSTRACT

Location on Tree Networks: P-Centre and n-Dispersion Problems

We consider two problems of locating facilities on trees. In the first, we wish to determine the location of n points so as to maximize the minimum distance between any pair. In the second we wish to locate p centers of a single type of facility so as to minimize the maximum distance between any point and the center nearest to it. We provide polynomial algorithms for both problems, of $O(|N|^2 \log n \cdot \log |N|)$ for the first problem and of $O(|N|^2 \log p)$ for the second, where $|N|$ is the number of nodes in the tree.

1. Introduction and Plan of Paper

1.1. Introduction

In this paper we present polynomially bounded algorithms for two related problems of locating facilities on tree networks. More precisely, let $T = T(N,A)$ be an undirected tree with N and A denoting the set of all nodes and arcs respectively. Each arc of the tree is associated with a positive number called the length of the arc. Suppose T is embedded in the Euclidean plane so that the arcs are line segments whose endpoints are nodes and arcs intersect only at nodes. By a point on T we mean a point along any arc of T. If u and v are points on T, then d(u,v) represents the distance between u and v measured along the arcs of T. The two problems solved in this paper are:

1) The n-center dispersion problem: Locate n facilities on T so that they are as far apart as possible, i.e. find points $x_1,\ldots,x_n$ on T so as to maximize $\min_{1\le i<j\le n} d(x_i,x_j)$.

2) The p-center problem: locate p facilities on T so that the maximum distance for any point on T to its nearest facility is minimized, i.e. find points $y_1,\ldots,y_p$ on T so as to minimize

$$\max_{x\in T} \min_{1\le i\le p} d(x,y_i).$$

The p-center problem has received a great deal of attention in the literature [3,6-16,18-20] and therefore seems to require little motivation at this point. The dispersion problem is very closely related to the p-center problem [19] but has received surprisingly little attention and needs some motivation. This is provided below.

Daughety and Turnquist [5] provide a procedure for solving a non-linear program whose objective function has to be estimated. The search procedure to estimate the objective function is based on solving a sequence of n-center dispersion problems. Church and Garfinkel [4] have looked at the problem of locating one obnoxious facility on a network. The work in this paper can be extended [20] to solve a q-center obnoxious facility problem, where we wish to locate new facilities not only as far apart from each other as possible but also from existing facilities. Siting of strategic military facilities and of nuclear waste-storage points are examples of this problem. Thus the n-center dispersion problem is of interest in its own right. In addition, the duality relationship exhibited by Shier [19], relates the dispersion problem to the p-center problem. Shier [19] shows that the objective function values of the p-center problem and the (p+1)-center dispersion problem are equal. The results in our paper shows that there exists a pair of methods for the two problems that closely resemble each other. Indeed, using the results of this paper, it is clear that one could use any method to solve one problem to yield an algorithm for the other problem.

We have restricted our attention to locating centers on trees as opposed to general networks. There are several reasons. First of all, the obvious one: we know how to solve them efficiently as shown by the polynomial algorithms presented here. Secondly, we know [16] that the p-center problem on general network (when we restrict ourselves to the discrete case) is NP-complete; hence the likelihood of finding a polynomial algorithm for the general case is small at least for the discrete case. While we do not know whether the algorithms presented herein are extendable to other special cases, and while it is easy to show that the

duality relationship between the two problems fails for general network, these two points are unrelated since the algorithms themselves do not use the duality results. It is our expectation that a deeper understanding of the two problems for trees will lead to a better understanding of the general case. Moreover, the kind of justification that exists for examining the p-center problem clearly also exists for the n-center dispersion problem for trees.

Finally, the methods presented herein are of interest in their own right. This is especially true of the kind of binary search that is described later on for determining the right value of the two objective functions.

1.2. Plan of the Paper

The next section treats the dispersion problem. First we provide a $O(|N|\log|N|)$ algorithm to compute $N(\lambda)$, the maximum number of facilities that can be located on the tree so that the distance between any pair of them is at least λ. Since $N(\lambda)$ is a monotonic step function of λ, one could use a binary search to determine $\lambda^*(n)$, the optimal value of the objective function for the dispersion problem. However, this does not, by itself yield a polynomial algorithm. The procedure we use does and it appears that a similar process might work for some other problems wherein the monotonic step-function over which the binary search is carried out has an exponential number of steps. We consider this special binary search in itself an important result of this paper.

In section three, we provide similar algorithms for the p-center problem. The pattern of section two is repeated. First we find $P(r)$, the mini-

mum number of facilities needed so that every point on the tree has a facility at a distance at most r. The algorithm provided to do this is $O(|N|)$. This again leads to an exact algorithm to determine $r^*(p)$ (the optimal value of the objective function for the p-center problem) using an approach similar to the one used for the dispersion problem.

2.0. The n-Center Dispersion Problem

2.1. Clusters, Reductions and Computation of $N(\lambda)$

A node of the tree T whose degree is 1 is called a tip. A maximal set of tips which are connected to the same node, say s, is called a cluster C(s). The removal of nodes in C(s) along with arcs incident at them will render s a tip in the remaining tree. The algorithm to determine the maximum number $N(\lambda)$ of facilities that can be located on T so that the distance between any pair of them is at least λ proceeds as follows: 1) at each step a set of facilities will be located along arcs (s,i) where $i \in C(s)$ and then 2) the cluster C(s) is eliminated. A repeated application of this procedure gives the desired result. We first provide some simple results that motivate this cluster elimination routine.

R1: If $i \in C(s)$, $i \neq s$ and there is an optimal solution with a facility on arc (s,i) then there is an optimal solution with a facility at i. Hence if $d(s,i) \geq \lambda$ we can locate a facility at i and reduce the length of arc (s,i) by λ. This process results in a situation in which all arcs associated with a cluster are of length less than λ. Indeed, the process could be applied to any arc that is incident at a tip.

R2: Let $i,j \in C(s)$ and let $d(s,j) \geq d(s,i)$. If in some optimal solution a facility is located at i then there is an optimal solution with a facility located at j.

R3: Let C(s) be a cluster whose members are numbered so that $\lambda > d(s,i) \geq d(s,j)$ whenever $i < j$. Let

$$h^o = \max_{h \in C(s)} \{h: d(s,h) + d(s,h+1) \geq \lambda\}$$

Then there is an optimal solution with no facility on arcs (s,j) for $j \geq h^o + 2$, $j \in C(s)$. Thus these nodes may be removed along with arcs incident at them. Thus the distance between any pair of remaining tips in C(s) is at least λ.

R4: Let C(s) be a cluster with $d(s,j) < \lambda \ \forall j \in C(s)$ and $d(i,j) \geq \lambda \ \forall i \neq j$, $i,j \in C(s)$. Let t be the unique node of T not in C(s), but adjacent to s. If $d(j,t) \geq \lambda$ for some $j \in C(s)$, then there is an optimal solution which has a facility at j. This holds since the only possible reason for not locating a facility at j in such a network is that there is one on arc (t,s) at a point k with $d(k,j) < \lambda$. In this case one could simply move this facility to j and not change the value of the objective function.

R5: If C(s) is a cluster with $d(s,j) < \lambda \ \forall j \in C(s)$ and $d(i,j) \geq \lambda \ \forall i \neq j$, $i,j \in C(s)$ and there is no t as in R4 then locate a facility at each node $i \in C(s)$ none at s and Stop.

If in R4 it turns out that $d(j,t) \geq \lambda \ \forall j \in C(s)$, then we locate a facility at each $j \in C(s)$, and remove these nodes along with the arcs (s,j).

Also reduce the length of the (t,s) to $d(t,s)-\beta$ where $\beta = \lambda - \min_{j \in C(s)} d(s,j)$ (as shown in Figure 1 below). If not, we locate a facility at each $j \in C(s)$ with $d(j,t) \geq \lambda$ and remove such nodes j and arcs (s,j). Also remove node s and connect t directly with remaining nodes in C(s) and let d(t,j) remain the same for these nodes (see Figure 2). Cluster C(s) is eliminated at this point.

By using these rules repeatedly we can determine $N(\lambda)$. Some savings are obtained by noting that the lengths of arcs created by R4 are all less than λ and are already ordered for R3.

Since no newly created arc goes through R1, the number of computations in R1 is of $O(|N|)$. Proper implementation of R3 (using the information that newly created arcs are already ordered) gives a bound of $O(|N|\log|N|)$ for ordering and $O(\log|N|)$ for determining h^o. Hence the entire algorithm for determining $N(\lambda)$ is $O(|N|\log|N|)$.

So far, we have described an algorithm to compute $N(\lambda)$. In the dispersion problem, however, we are given the number n of facilities to be located, not the minimum separation λ. Using the fact that $N(\lambda)$ is a monotone nonincreasing left continuous step function of λ one could solve for $\lambda^* = \max\{\lambda : N(\lambda) \geq n\}$ using binary search that uses the above described algorithm as a subroutine. Unfortunately, there is no guarantee that the resulting algorithm will be polynomially bounded although in practice it might turn out to be quite efficient. What we describe below is a judicious binary search that is also polynomially bounded. Hence this procedure will also be practically efficient in addition to being theoretically satisfying. This procedure is also applicable to other problems in

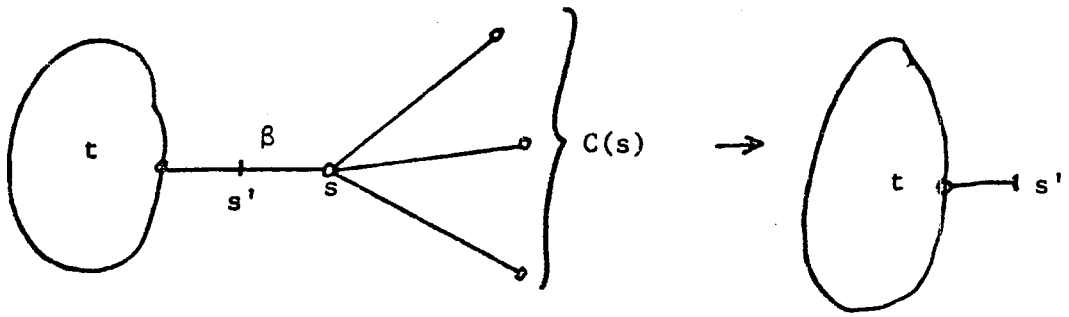

Figure 1

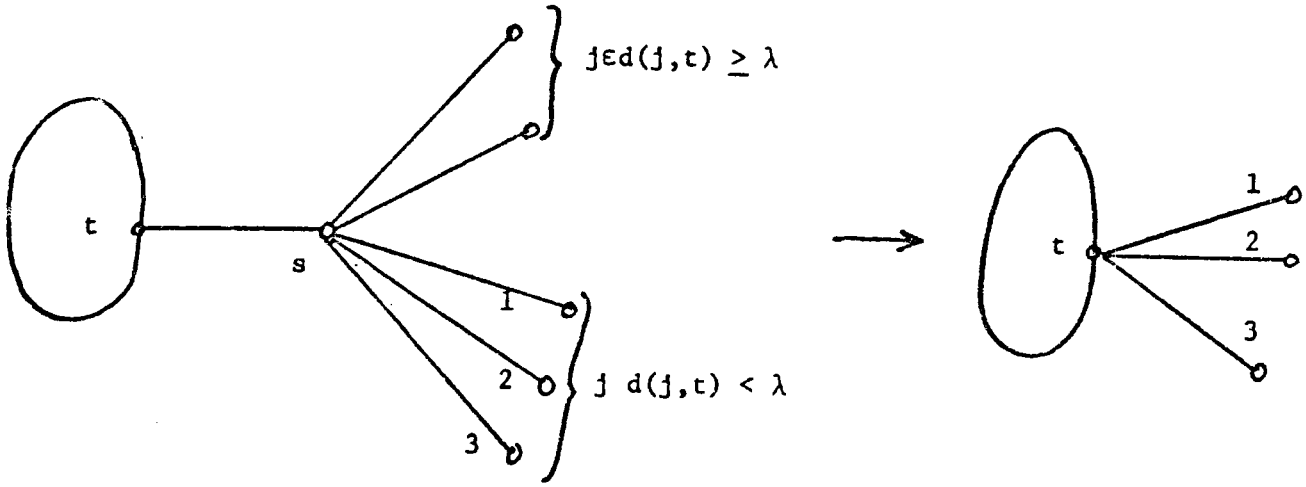

Figure 2

which the value of a parameter is to be determined by a binary search.

2.2. Determination of $\lambda^*(n)$

If we study the algorithm to determine $N(\lambda)$ carefully, we note that remaining arc lengths are always linear functions of λ . Also, it is clear that not only does $N(\lambda)$ not change for pairs of values of λ that are sufficiently close, but the algorithm itself has precisely the same realization except for slightly altered values of the reduced arc lengths. More precisely, the number of facilities located on arcs incident at tips in R1 remains the same, the ordering in R3 remains the same, the graph reduction in R4 (depicted in Figures 1 and 2) remain the same and so on. Thus, every time the algorithm has a possibility to branch into different realizations, the branching remains the same for values of λ that are sufficiently close. The idea is to zero in on an interval of values of λ that contains $\lambda^*(n)$ over which the realization of the algorithm is that same as at $\lambda^*(n)$, To do this efficiently, we first partition the real numbers (the set of possible values of λ) into a (polynomially bounded) number of disjoint intervals and by using the algorithm described above determine the interval that contains $\lambda^*(n)$. This is based on the idea that with λ in an interval of the type above, the realization of R1 is the same. For example, the number of facilities located on arc (s,i) $i \in C(s)$ in R1 equals k for all $\lambda \in (\frac{d(s,i)}{k+1}, \frac{d(s,i)}{k})]$. Thus, if we let

$$Q_i(s) = \{ \frac{d(s,i)}{k}, \quad 1 \leq k \leq n\}$$

and $$Q(s) = \bigcup_{i \in C(s)} Q_i(s)$$

$Q(s)$ is a set of real numbers with $|Q(s)| \leq |C(s)| \cdot n$. If $Q(s)$ were ordered

(by the usual ordering of the real numbers) this provides the first partition of the reals as indicated above. Within any of the intervals defined by this partition R1 has the same realization. Let us suppose that the interval of the above partition containing $\lambda^*(n)$ is $[q_r, q_{r+1})$ where q_r is r^{th} element in the ordered set Q(s). Also suppose $d(s,i) = k_i \lambda + a_i(\lambda)$ $0 \leq a_i(\lambda) < \lambda$ for $\lambda \in [q_r, q_{r+1})$; note that k_i does not change. Thus $a_i(\lambda) = d(s,i) - k_i\lambda$, a linear function of λ. We subdivide this interval $[q_r, q_{r+1})$ into a polynomial number of subintervals so that the realization of R3 is the same within each one of them. This is done in two steps. In the first step, we make sure that the ordering of $i \in C(s)$ remains the same. Hence consider numbers $a_i(\lambda) = d(s,i) - k_i\lambda$. The only values of λ of interest for which the arrangement of $a_i(\lambda)$ in R3 changes are contained in T(s) where

$$T(s) = \{\lambda \mid \lambda \in [q_r, q_{r+1}),\ \lambda = \frac{d(s,i)-d(s,j)}{k_i - k_j},\ k_i \neq k_j,\ i,j \in C(s)\}$$

Note that T(s) is a set of real numbers and $|T(s)| \leq |C(s)|^2$. Thus T(s) induces a partition of the interval $[q_r, q_{r+1})$ into subintervals and we can find a subinterval $[t_\ell, t_{\ell+1})$ such that $N(t_\ell) \geq n$, $N(t_{\ell+1}) < n$ and hence $\lambda^*(n) \in [t_\ell, t_{\ell+1})$. Now we turn to the second step in R3 which determines the index h^o. It is clear that the only values of λ for which h^o in R3 might possibly change are contained in W(s) which is designed as follows:

$$W(s) = \{\lambda \mid \lambda \in [t_\ell, t_{\ell+1}),\ a_i(\lambda) + a_{i+1}(\lambda) = \lambda \text{ for some } i\}$$

It is assumed that $a_i(\lambda)$ are arranged in decreasing order in the above definition of W(s); note that by above argument, this arrangement does

not change for $\lambda \in [t_\ell, t_{\ell+1})$ $W(s)$ is a set of reals and $|W(s)| \leq |C(s)|$. The ordered set of numbers in $W(s)$ induce the third partition. Once again we can determine the subinterval $[W_\alpha, W_{\alpha+1})$ of $[t_\ell, t_{\ell+1})$ that contains $\lambda^*(n)$. Now we proceed to R4 and the cluster elimination procedure for further partitioning of this subinterval $[W_\alpha, W_{\alpha+1})$. The set of values λ for which the configuration in R4 might change are contained in

$$X(s) = \{\lambda | \lambda \in [W_\alpha, W_{\alpha+1}),\ d(s,t) + a_i(\lambda) = \lambda\}$$

$X(s)$ is a set of real numbers and $|X(s)| \leq |C(s)|$. The ordered set of numbers in $X(s)$ induce the fourth partition. Once again we can determine the subinterval $[X_\beta, X_{\beta+1})$ of $[W_\alpha, W_{\alpha+1})$ that contains $\lambda^*(n)$. We have now completed one cycle and the process is repeated. Similar arguments apply to R5. The amount of work in one of these cycles is $O(|C(s)| \cdot \log n \cdot |N| \cdot \log|N|)$. At the end of this cycle the network has at least one fewer nodes.

However, if in R4 we are in the case depicted by figure 1, then the old ones in the cluster disappear and if we are in the Case depicted by figure 2, then these ones do not have to go through many of the steps again. This can be used to reduce the work and we can use two ordered sets to create a larger order set without starting from scratch. When all this is done the total effort involved in the algorithm is of $O(|N|^2 \log n \cdot \log|N|)$. Hence the algorithm is polynomially bounded.

3. The p-Center Problem

3.1. Computation of P(r)

The notions of tip and cluster are as in section 2.1. The procedure is to determine the minimum number of facilities required to be located along arcs (s,i) for $i \in C(s)$ and then to eliminate the cluster C(s). A repeated application of this procedure gives the desired result. As some of the details of the process are slightly different, we provide them below.

Cluster Elimination Routine:

Step 0: Choose a cluster C(s).

Step 1: Let $\{(s,i) | i \in C(s)\}$ be the set of arcs connecting the tips to their predecessor s. For each $i \in C(s)$ let $d(s,i) = 2rk_i + b_i$ where k_i is a nonnegative integer and $0 < b_i \leq 2r$.
Set $d(s,i) \leftarrow b_i$ for $i \in C(s)$.

(At this point k_i facilities have already been located on arc (s,i) with distance between adjacent facilities equal to 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

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

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 taken to be $+\infty$ ($-\infty$). (Note that at least one of α,β is finite).

(i): If $(\alpha+\beta) > 2r$, then for each $i \in C(s)$ 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 arc (s,i) in $C(s)$ except (s,i_2^*). If s now becomes a tip, locate a facility at s and terminate. Otherwise remove node s as shown in Figure 3 and go to Step 3.

(ii): If $(\alpha+\beta) \leq 2r$, then for each $i \neq i_1^*$, $i \in C(s)$ with $d(s,i) > r$, locate a facility on (s,i) at a distance r from the tip i. Remove all arcs except (s,i_1^*). If s is now a tip, locate a facility on (s,i_1^*) at a distance r from i_1^* and terminate. Otherwise remove node s as shown in Figure 3 and to to Step 3.

(iii): Choose a cluster of the remaining tree and return to Step 1.

The reasoning for this procedure is along the same lines as that for the dispersion problem. It is clear that for $i \in C(s)$ if $d(s,i) > 2r$, then a facility must be located on arc (s,i) at a distance not greater than r from the tip i. Since this facility may be located at a distance exactly equal to r (from i) Step 1 follows. It is clear that after reductions in Step 1 are done, there must be a facility on each arc (s,i) with $d(s,i) > r$ and if $\alpha + \beta \leq 2r$, one of these serves the arcs (s,i) with $d(s,i) \leq r$. If not, we need one additional facility and this has to serve arc (s,i_2^*) and hence the reductions in Step 2 are justified. If in Step 1 we only record the number k of facilities then this algorithm is $O(|N|)$.

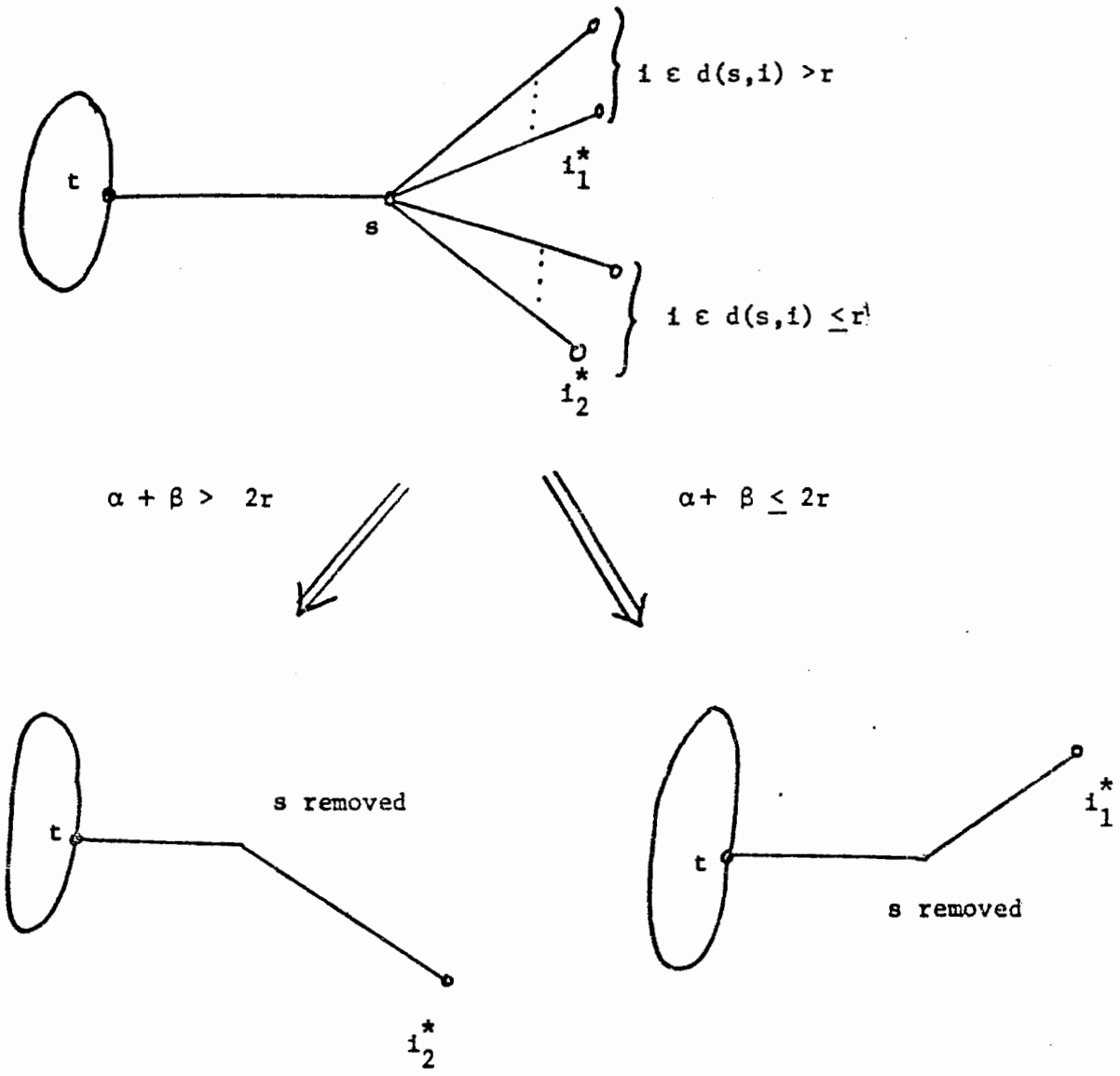

Figure 3

3.2 Determination of $r^*(p)$

The approach used is similar to the one used for determining $\lambda^*(n)$. Again we note that $P(r)$ is a monotone non-increasing step function. The first partition this time is based on keeping the values k_i the same in Step 1. For this we let

$$Q_i(s) = \{\frac{d(s,i)}{k} \quad , \quad 1 \leq k \leq p\} \qquad i \in C(s)$$

$$Q(s) = \bigcup_{i \in C(s)} Q_i(s)$$

$Q(s)$ is a set of real numbers with $|Q(s)| \leq p \cdot |C(s)|$. If $Q(s)$ were ordered (by the usual ordering for real numbers), this provides the first partition . Suppose in this ordering we determine $r^*(p) \in [q_\ell, q_{\ell+1})$. The lengths $d(s,i)$ of the arcs (s,i) after Step 1 become $b_i(r) = d(s,i) - 2rk_i$. (Note that k_i are fixed for $r \in [q_\ell, q_{\ell+1})$). The only possible candidate values of r of interest for which the realization of Step 2 might change are contained in $T(s)$ given by

$$T(s) = \{r | r \in [q_\ell, q_{\ell+1}); 2r = b_i(r) + b_j(r),\ i,j \in C(s) \qquad i \neq j\}$$

$T(s)$ is a set of real numbers and $|T(s)| \leq |C(s)|^2$. The ordered set $T(s)$ gives us the partition of the interval $[q_\ell, q_{\ell+1})$ so that within each sub-interval the realization of Step 2 remains the same. We have now completed one cycle and the process is repeated. The amount of work in one of these cycles is $O(|C(s)| \cdot \log p \cdot |N|)$. At the end of this cycle the network has at least one fewer nodes. Thus the total effort $O(|N|^2 \log p)$, if the implementation uses work reduction as in the case of dispersion problem.

References

1. Chandrasekaran, R. "Minimal Ratio Spanning Tree," Networks, Vol. 7, pp. 335-342 (1977).

2. Chandrasekaran, R. and A. Tamir. "Polynomially Bounded Algorithms for Locating p-Centres on a Tree," Discussion Paper No. 358, Center for Mathematical Studies in Economics and Management Science, Northwestern University, 1978.

3. Christofides, N. and P. Viola. "The Optimum Location of Multi-Centres on a Graph," Operational Research Quarterly, Vol. 22, pp. 145-154 (1971).

4. Church, Richard L. and Robert S. Garfinkel. "Locating an Obnoxious Facility on a Network," Transportation Science, Vol. 12, pp. 107-118 (1978).

5. Daughety, A.F. and M.A. Turnquist. "Budget Constrained Optimization of Simulation Models Via Estimation of Their Response Surfaces," Working Paper No. 425-06, The Transportation Center, Northwestern University, April 1979.

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

7. Garfinkel, R.S., A.W. Neebe and M.R. Rao. "The m-Center Problem: Minimax Facility Location," Management Science, Vol. 23, pp. 1133-1142 (1977).

8. Goldman, A.J. "Optimal Center Location in Simple Networks," Transportation Science, Vol. 5, pp. 212-221 (1971).

9. Goldman, A.J. "Minimax Location of a Facility on a Network," Transportation Science, Vol. 6, pp. 407-418 (1972).

10. Hakimi, S.L. "Optimal Location of Switching Centers and the Absolute Centers and Medians on a Graph," Operations Research, Vol. 12, pp. 450-459 (1964).

11. Hakimi, S.L. "Optimal Distribution of Switching Centers in a Communications Network and Some Related Graph Theoretic Problems," Operations Research, Vol. 13, pp. 462-475 (1965).

12. Hakimi, S.L., E.F. Schmeichel and J.G. Pierce. "On p-Centers in Networks," Transportation Science, Vol. 12, No. 1, pp. 1-15.

13. Halfin, S. "On Finding the Absolute and Vertex Centers of a Tree with Distances," Transportation Science, Vol. 7, pp. 75-77 (1974).

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

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

16. Kariv, O. and S.L. Hakimi. "An Algorithmic Approach to Network Location Problems. Part I: The p-Centers," forthcoming, SIAM Journal of Applied Mathematics.

17. Megiddo, N. "Combinatorial Optimization with Rational Objective Functions," to appear in Mathematics of Operations Research.

18. Minieka, E. "The m-Center Problem," SIAM Review, Vol. 12, pp. 138-139 (1970).

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

20. Slater, P.J. "R-Domination in Graphs," Journal of the Association for Computing Machinery, Vol. 23, No. 3, July 1976, pp. 446-450.

21. Tamir, A. and E. Zemel. "Locating Centers on a Tree with Discontinuous Supply and Demand Regions," Discussion Paper No. 397, Center for Mathematical Studies in Economics and Management Science, Northwestern University, September 1979.