A spanning tree approach to the absolute p-center problem ☆

Burçin Bozkaya [a,*], Barbaros Tansel [b]

[a] *Faculty of Business, University of Alberta, Edmonton AB, Canada T6G 2R6*
[b] *Department of Industrial Engineering, Bilkent University, Bilkent Ankara 06533, Turkey*

Abstract

We consider the absolute p-center problem on a general network and propose a spanning tree approach which is motivated by the fact that the problem is NP-hard on general networks but solvable in polynomial time on trees. We first prove that every connected network possesses a spanning tree whose p-center solution is also a solution for the network under consideration. Then we propose two classes of spanning trees that are shortest path trees rooted at certain points of the network. We give an experimental study, based on 1440 instances, to test how often these classes of trees include an optimizing tree. We report our computational results on the performance of both types of trees. © 1999 Elsevier Science Ltd. All rights reserved.

Keywords: Facility location; p-center; Spanning tree

1. Introduction

The absolute p-center problem is a model for locating p identical facilities anywhere on a network to minimize the maximum (weighted) distance between each vertex (demand) and its closest facility. The model finds applications in the location of emergency service facilities such as hospitals, ambulance and fire stations, etc. The problem is NP-hard on general networks, but solvable in polynomial time on tree networks; (Kariv and Hakimi, 1979).

For $p = 1$, Dearing and Francis (1974) have shown that the union of shortest paths connecting the optimal 1-center of a network to the vertices forms a spanning

---

tree whose optimal 1-center coincides with that of the network. A natural question to ask is whether this result extends to the case with $p > 1$. That is, does every connected network have a spanning tree whose optimal p-center solution is the same as that of the network, and if it does, what search strategies can be devised to find an *optimal tree* (a spanning tree that supplies an optimal solution to the network)? We explore this question by first proving the existence of an optimal tree (Theorem 1), and then proposing two classes of spanning trees that are suspected of containing an optimal one. It is important to note here that the identification of an optimal tree in polynomial time would mean $P = NP$. Hence, confining the search for an optimal tree to a polynomial-sized subset of all spanning trees is as hard as the p-center problem itself. We implement a computational study and report our results on the success rates of the two proposed classes of trees.

For a brief literature review of the problem, Hakimi (1964) defined and solved the absolute 1-center problem by examining the piecewise linear objective function on each edge and finding the edge-restricted minimum at one of the breakpoints. The smallest among the edge-restricted minima is the absolute 1-center of the network. Hakimi et al. (1978) further reduced the computational effort in Hakimi's algorithm. Hakimi (1965) defined the absolute p-center problem and developed a solution procedure based on solving a sequence of set covering problems. Christofides and Viola (1971) also employed the idea of using the set-covering problem in their algorithm. Minieka (1970), for the unweighted case, and Kariv and Hakimi (1979), for the weighted case, showed that the optimal solution of the problem is restricted to a finite set of points on the network. Hooker et al. (1991) provided later a unified framework for establishing finite dominating sets for rather general classes of network location problems. Hooker et al.'s results include as special cases the dominating set properties of Minieka (1970), and of Kariv and Hakimi (1979). Since the first appearance of this problem, researchers have studied many different versions of the problem, such as the "conditional" 1-center (Minieka, 1980), 2-center (Handler, 1978), unweighted p-center (Handler, 1973; Hedetniemi et al., 1981; Minieka, 1981), vertex-restricted p-center (Toregas et al., 1971; Hooker, 1989), p-center with continuous demand points (Chandrasekaran and Tamir, 1980; Chandrasekaran and Daughety, 1981; Megiddo et al., 1981; Tamir, 1985) and p-center problems in which the weighted distances are replaced by non-linear functions of distances (Tansel et al., 1982; Hooker, 1986, 1989).

The rest of this paper is organized as follows. In Section 2, we define the problem and prove the main theorem. In Section 3, we describe the two classes of spanning trees that are suspected of containing an optimal tree. Section 4 describes the computational study and analyzes the results of assessing the success rates of the proposed classes of trees. The paper ends with concluding remarks in Section 5.

2. Problem and main theorem

Let $N = (V, E)$ be an embedding of a connected network in some space S (e.g. the plane), as defined in Dearing and Francis (1974), where $V = \{v_1, \ldots, v_n\} \subset S$ is the

vertex set consisting of n distinct points in S, and E is the edge set consisting of embedded edges $[v_i, v_j] \subset S$. Each embedded edge $[v_i, v_j]$ is the image, $H_{ij}([0, 1])$, of the unit interval under a one-to-one continuous mapping $H_{ij} : [0, 1] \to S$ where $H_{ij}(0) = v_i$, $H_{ij}(1) = v_j$, and $H_{ij}(a)$ is some point in the interior of $[v_i, v_j]$ for $0 < a < 1$. We take N as the union of its embedded edges and omit the term 'embedded' in the rest of the paper. A point $x \in N$ is either a vertex or a point in the interior of some edge $[v_i, v_j]$ in which case x subdivides the edge into two subedges $[v_i, x]$ and $[x, v_j]$ where $[v_i, x] \cup [x, v_j] = [v_i, v_j]$ and $[v_i, x] \cap [x, v_j] = \{x\}$. The edges are assigned positive lengths. If the length of edge $[v_i, v_j]$ is L_{ij} and if x is a point in this edge with $x = H_{ij}(a)$ for some $a \in [0, 1]$, then the lengths of subedges $[v_i, x]$ and $[x, v_j]$ are aL_{ij} and $(1 - a)L_{ij}$, respectively. Let $X = \{x_1, \ldots, x_p\} \subset N$ be any set of p points at which p facilities (servers) will be located and let $S_p(N)$ be the set of all point sets X with $X \subset N$ and $|X| = p$. Note that $S_p(N)$ is an infinite set. Let $d(x, y)$ be the shortest path distance between any two points $x, y \in N$ and denote the distance of vertex v_i to its closest facility by $D(v_i, X) = \min\{d(v_i, x_j) : x_j \in X\}$. The absolute p-center problem is:

$$\min_{X \in S_p(N)} f(X) \quad \text{where } f(X) = \max_{1 \leqslant i \leqslant n} w_i \cdot D(v_i, X). \tag{1}$$

Here, the w_i are non-negative weights that may reflect the relative importance of each vertex. If X^* solves Eq. (1), we call X^* a *p-center* and call $z_p(N) = f(X^*)$ the *p-radius* of N. If $X \in S_p(N)$, we call X a *feasible solution* or a *candidate p-center*. Each x_j in X will be referred to as a facility or a server. Tansel et al. (1983) and Mirchandani and Francis (1990) provide extensive information on various algorithmic and theoretical aspects of this problem.

A spanning tree of N is any subgraph of N that is connected, has no cycles, and contains all vertices of N. Let T be any spanning tree of N and define $d_T(x, y), D_T(v_i, X), f_T(X)$, and $S_p(T)$ in exactly the same way as $d(x, y), D(v_i, X), f(X)$, and $S_p(N)$, respectively, except that everything is relative to T rather than N. The p-center problem restricted to T is

$$\min_{X \in S_p(T)} f_T(X). \tag{2}$$

Denote by $z_p(T)$ the minimum objective function value in Eq. (2), i.e. the p-radius of T. If $ST(N)$ is the set of all spanning trees of N, it is clear that

$$z_p(N) \leqslant z_p(T) \quad \forall T \in ST(N). \tag{3}$$

The inequality is a consequence of the fact that any p-center for the tree T is a feasible solution for the problem on N.

The next theorem shows that equality is achieved in Eq. (3) by at least one spanning tree.

Theorem 1. *Let N be any connected network. There exists a spanning tree $\overline{T}$ of N such that $z_p(\overline{T}) = z_p(N)$. Consequently, if $\overline{X} \in S_p(\overline{T})$ is a p-center of $\overline{T}$ then $\overline{X}$ is also a p-center of N.*

Proof. Let $X^* = \{x_1^*, \ldots, x_p^*\}$ be a p-center of N. Construct an optimal tree $\overline{T}$ from X^* as follows. Partition V into disjoint subsets $V_1, \ldots, V_p$ where V_j consists of the vertices $v_i \in V$ such that $D(v_i, X^*) = d(v_i, x_j^*)$ and $D(v_i, X^*) < d(v_i, x_{j'}^*)$ for all $j' < j$. That is, each vertex $v_i \in V$ is assigned to the closest facility where ties are broken by selecting the smallest-indexed facility among tied ones, so vertices in V_j are served by x_j^*. We may also assume without loss of generality that each V_j in this partition is non-empty. Otherwise, if $V_j = \phi$, then we may replace x_j^* with an arbitrary vertex, say v_k, so that a re-partitioning of V with respect to the new p-center so obtained ensures that v_k is assigned to the new j th facility.

With $V_j \neq \phi$ $(j = 1, \ldots, p)$, let T_j be a shortest path tree which is rooted at x_j^* and which spans the vertices in V_j. Note that T_j is defined by the union of shortest paths between each $v_i \in T_j$ and x_j^*, and its existence is guaranteed (Busacker and Saaty, 1965). Note also that any tip vertex of T_j is necessarily in V_j. However, it is not obvious that a non-tip vertex v' of T_j is in V_j. But, in what immediately follows, we show that non-tip vertices of T_j are necessarily in V_j which also ensures that $T_1, \ldots, T_p$ are disjoint subtrees. Once this is shown, it is direct to add $(p-1)$ edges from $N \setminus \cup_{j=1}^p T_j$ to the forest $\{T_1, \ldots, T_p\}$ to complete it and form a spanning tree $\overline{T}$.

To prove the claim, let v' be a non-tip vertex of T_j and suppose that v' were on the shortest path connecting x_j^* and a tip vertex $v'' \in T_j$ (see Fig. 1). Then we have $d(v', x_j^*) \leqslant d(v', x_i^*)$ for $i \neq j$, i.e., the path from v' to x_j^* must be a shortest path from v' to its nearest facility(s). (For if not, we could reduce the length of the path from v'' to its nearest facility by serving v'' from the same facility that v' is served from, a contradiction of the definition of T_j.) Now suppose $v' \notin V_j$. Then there must be some facility with an index $i < j$ such that $d(v', x_j^*) = d(v', x_i^*)$. However, all $v \in T_j$ that are "successors" of v' (see circled part of Fig. 1) would also be served by the facility with
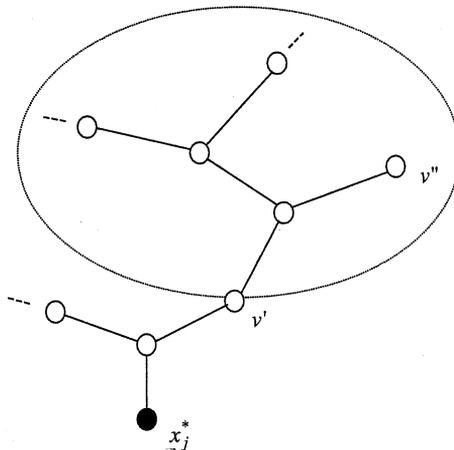

Fig. 1. A non-tip vertex, v', of T_j must be in V_j

the index i since $i < j$ and thus must be in V_i, a contradiction. Therefore any non-tip vertex of T_j must be in V_j, and it follows that the subtrees are disjoint.

At this point, we have p disjoint trees T_j's, each of which spans the vertices (and only those vertices) in the associated V_j. The next step is to combine these trees into a spanning tree by adding $(p-1)$ edges appropriately. Note that, since N is connected, there always exists an edge $e = [v_s, v_t]$ such that $v_s \in T_k$ and $v_t \in T_l$ for some $k \neq l$, and $e \notin T_j$, $\forall j = 1, \ldots, p$. Hence, e can be used to combine T_k and T_l. The T_j's are combined into a spanning tree of N in this way and the tree $\overline{T}$ that we are looking for is constructed. Now we prove that $\overline{T}$ has the same p-radius as that of N.

To show $z_p(\overline{T}) = z_p(N)$, suppose we solve a 1-center problem on each T_j and let $\overline{X} = \{\bar{x}_1, \ldots, \bar{x}_p\}$ be the set of the corresponding 1-centers. Observe that $\overline{X}$ is feasible on both $\overline{T}$ and N. We have

$$z_p(\overline{T}) \leqslant f_{\overline{T}}(\overline{X}). \tag{4}$$

For each T_j, we have

$$z_1(T_j) = \max_{v_i \in V_j}\{w_i \cdot d_{T_j}(v_i, \bar{x}_j)\} \leqslant \max_{v_i \in V_j}\{w_i \cdot d_{T_j}(v_i, x_j^*)\}$$
$$= \max_{v_i \in V_j}\{w_i \cdot d(v_i, x_j^*)\} \leqslant z_p(N)$$

which gives

$$z_p(\overline{T}) \leqslant f_{\overline{T}}(\overline{X}) \leqslant \max_{j=1,\ldots,p} z_1(T_j) \leqslant z_p(N).$$

From Eq. (3), we also have $z_p(\overline{T}) \geqslant z_p(N)$, since any spanning tree of N is a subgraph, hence a restriction, of N with the same weights and edge lengths. This implies

$$z_p(\overline{T}) = z_p(N)$$

which completes the proof of Theorem 1.

Although Theorem 1 shows the existence of an optimal tree, the proof requires knowledge of a p-center of N to construct such a tree. Thus, the question of how to search for an optimal tree without having knowledge of a p-center of N remains an open question. In the next section, we propose two classes of trees that provide the basis of a search strategy that performs well in many instances.

We remark in passing that the proof of Theorem 1 can be adapted to the p-median problem where the objective function is defined by the sum of weighted distances rather than the maximum weighted distance. Tansel et al. (1983) give extensive information on the p-median problem. Hakimi (1964, 1965) proved that there exists an optimal solution to the p-median problem on the vertices of the network. Hence, we may focus on the vertex-restricted problem without loss of optimality. Let $S_p(V)$ be the set of subsets of V consisting of p distinct vertices.

Theorem 2. *Let N be a connected network and $\mu_p = \min_{X \in S_p(V)} \sum_{i=1}^n w_i\, D(X, v_i)$. Define $\mu_p(T)$ similarly relative to any spanning tree T of N. There exists a spanning tree $\overline{T}$ of N such that $\mu_p(\overline{T}) = \mu_p(N)$ and every optimal solution for the p-median problem on $\overline{T}$ is also an optimal solution for the p-median problem on N.*

3. Rooted shortest path trees

In general, the number of spanning trees of a network can be excessively large. A complete network of *n* vertices has n^{n-2} distinct spanning trees (Moon, 1967), and this provides an upper bound for any network even though more complicated formulas for the exact count are available for general networks (Riordan, 1958). One such formula is given by Thulasiraman and Swamy (1992), with reference to Kirchoff (1847), which computes the exact count as the value of any cofactor of $D - A$ where *D* is the diagonal degree matrix and *A* is the adjacency matrix of the network.

One can efficiently solve the *p*-center problem on a cyclic network if the number of spanning trees is polynomial. This is usually the case if the network is sparse or has a simple structure, e.g., a network with few cycles, as is the case with many highway networks. For the general case, we introduce two types of trees that are suspected of containing an optimal tree. Both types of trees will be referred to as rooted shortest path trees (RSPTs) as they are constructed by picking certain points of the network as "roots" and forming the union of shortest paths that connect the roots to the vertices. Our motivation for choosing these sets of trees is given in the corresponding section. Later, we give an experimental search for the optimal tree (i.e., a spanning tree that supplies an optimal solution to the network) in these two types of spanning trees.

3.1. Trees rooted at segments (S-RSPTs)

The first class of spanning trees used in the search for an optimal tree includes the trees rooted at edge segments of the network (the detailed description of S-RSPTs is given below). In our early experiments, the S-RSPTs included an optimizing tree in essentially all small-scale examples that we worked out by hand. For this reason, we found it worthwhile to test their performance in large-scale instances. Our test results with this class of trees are given in Section 4.1.

For any vertex v_k and edge $e = [v_p, v_q]$, it is well known that $d(v_k, x)$ as a function of *x* restricted to *e* is piecewise linear concave with one or two pieces (Fig. 2). If $d(v_k, \cdot)$ has two pieces on *e*, then there is a unique point, say $\overline{v}_k$, at which $d(v_k, \cdot)$ attains its maximum value. We call $\overline{v}_k$ an *antipodal* of v_k (Fig. 3). Let *A* be the set of all antipodals of all vertices on all edges. Since a vertex can have at most one antipodal on a given edge, the cardinality of *A* is $O(n|E|)$. Let $U = V \cup A$. We call *U* the *extended vertex set* and refer to each *u* in *U* as a *pseudo vertex* with the understanding that *u* is either a vertex or an antipodal. Two pseudo vertices $u, u' \in U$ are defined to be *adjacent* if they lie on the same edge and the edge segment that connects *u* and *u'* does not contain any other point of *U*.

$d(v_k, \cdot)$

$d(v_k, x)$

v_p x v_q
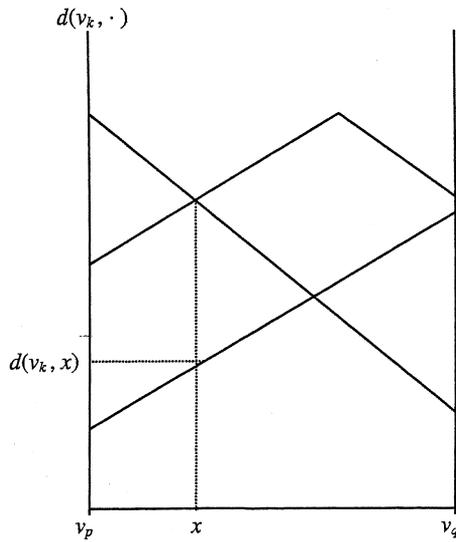
Fig. 2. Plot of $d(v_k, \cdot)$ on edge $e = [v_p, v_q]$.
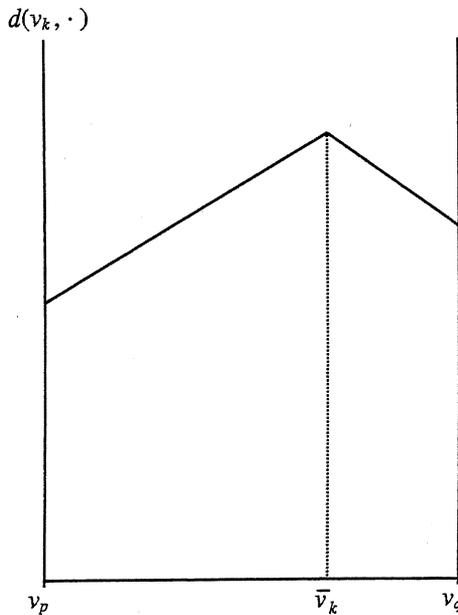
$d(v_k, \cdot)$

v_p $\bar{v}_k$ v_q

Fig. 3. $\bar{v}_k$: The antipodal of v_k on edge $e = [v_p, v_q]$.

Consider two adjacent pseudo vertices a, b and let $[a, b]$ be the edge segment that connects them. Observe that every distance function $d(v_k, \cdot)$ is linear on $[a, b]$; otherwise, $[a, b]$ contains an antipodal in its interior which means that a, b are not

adjacent. As a consequence, $[a, b]$ partitions the vertex set V into two non-empty subsets V_a, V_b as follows:

$$V_a = \{v_k \in V : d(v_k, a) < d(v_k, b)\},$$
$$V_b = \{v_k \in V : d(v_k, b) < d(v_k, a)\}.$$

To construct a spanning tree rooted at $[a, b]$, we first take a as a root vertex and find a set of shortest paths that connect a to every vertex in V_a. Let T_a be any such rooted shortest path tree that is constructed via Dijkstra's shortest path algorithm. We note that there may be alternate shortest paths between the root and any $v_k \in V_a$ but we take the first such path encountered during the path construction phase of Dijkstra's algorithm. Let T_b be constructed similarly. Observe that T_a and T_b are disjoint, as otherwise the existence of a vertex v which is in both T_a and T_b would imply that $d(v, a) = d(v, b)$, which is contradictory. Hence T_a spans V_a while T_b spans V_b with $T_a \cap T_b = \phi$.

Define $T(a, b) = T_a \cup T_b \cup [a, b]$ and call $T(a, b)$ the tree rooted at segment $[a, b]$. In the computational experiments, we refer to such trees as S-RSPTs ("S" for segment). The Dijkstra-based procedure constructs one such tree per segment. Since there are $O(n|E|)$ segments and each requires $O(n^2)$ time for Dijkstra's method, the total effort for the construction of S-RSPTs is $O(n^3|E|)$. We note that the actual number of S-RSPTs may be significantly larger than $O(n|E|)$ since the existence of alternate shortest paths may lead to many distinct rooted shortest path trees.

3.2. Trees rooted at intersection points (I-RSPTs)

The second class of spanning trees in which an optimizing tree can be searched for is the set of shortest path trees rooted at intersection points. The consideration of this set is motivated by the fact that a p-center of a network induces a partitioning of V and the network itself, which is closely related to the intersection points used as facility locations. To clarify this concept further, let $f_k(t) = w_k d(v_k, t)$ be the weighted distance between v_k and a point t on edge e. An *intersection point* on e defined by two distinct vertices v_k and v_l is a point $x \in e$, if it exists, such that $f_k(\cdot)$ and $f_l(\cdot)$ intersect at x, one with a positive, the other with a negative slope (Fig. 4). Kariv and Hakimi (1979) show that the optimal locations of facilities can be restricted to the union of the set of all intersection points and the vertices of N. In fact, given an absolute p-center $X^* = \{x_1^*, \ldots, x_p^*\}$, there is a natural partitioning of V into subsets $V_1, \ldots, V_p$ (as in the proof of Theorem 1) such that each x_j^* serves the vertices in V_j and that x_j^* can be moved without loss of optimality to some intersection point defined by a pair of vertices in V_j.

We use Dijkstra's method to construct a single shortest path tree (referred to as an I-RSPT) by taking each intersection point to be the root and constructing a shortest path tree that connects the root to all vertices in V. This generates $O(n^2|E|)$ I-RSPTs with a total effort of $O(n^4|E|)$. Computational results on the performance of these trees are reported in Section 4.1.
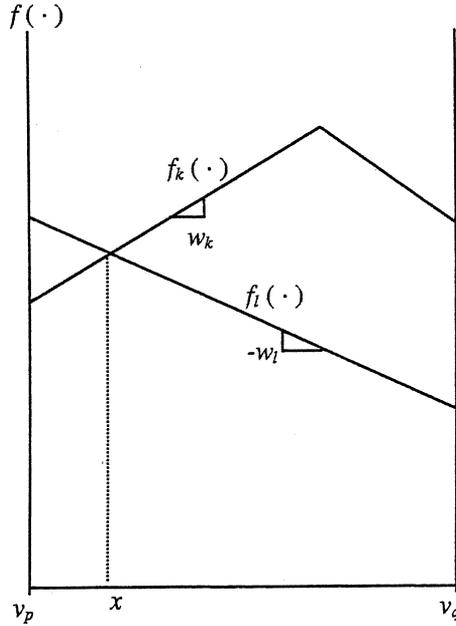
Fig. 4. An example of an intersection point x on $e = [v_p, v_q]$ defined by two distinct vertices v_k and v_l.

4. Computational experiments

In this section, we implement an experiment to test whether an optimal tree is included in the set of S-RSPTs or I-RSPTs. An instance of the problem is defined by the following factors. n is the number of vertices; d is edge density, the ratio (in percent) $|E|/\binom{n}{2}$; w is vertex weights; l is edge lengths; p is number of facilities. These factors are assigned the levels of values in Table 1.

Table 1
Factors and their levels

Factor	Number of levels	Levels
n	4	10, 20, 30, 40
d	3	25%, 50%, 75%
w	2	W or U*
l	1	Uniform from {1, 2, 3, 4, 5}
p	3	$\lfloor n/4 \rfloor, n/2, \lceil 3n/4 \rceil$
Total	72	

*W: weighted (uniform from {1,2,3}), U: unweighted ($w_i = 1, \forall i$).

For each combination of the factors (n, d, w, l, p), 10 random instances are generated, for a total of 720 instances. The unweighted instances are generated by simply making the corresponding weighted instance unweighted. They were included

for the purpose of testing whether or not the case of equal weights improves the results of weighted instances. The experiment is carried out in two stages that are described below:

Stage 1: First, for an instance N (generated by the NETGEN module coded by the authors, which randomly adds and then deletes edges until the connectivity and density requirements are met), the absolute p-center problem is solved on N. The algorithm used for solving the problem is based on the results of Kariv and Hakimi (1979) and Minieka (1970). This exact algorithm first identifies all the intersection points of the network together with the associated candidate p-radii. It then solves a sequence of set-covering problems using the intersection points found as candidate facility locations that "cover" the demand points within the p-radius used.

After the problem is solved on N, all the S-RSPTs (I-RSPTs) of N are constructed and the problem is solved on each (using the results of Tansel et al., 1982 and Tansel et al. 1990). The algorithm used for this purpose again solves a sequence of set-covering problems. Each set-covering problem is solved by starting at the tip vertices of the tree, and then locating facilities as needed while moving towards the "interior" of the tree. The best p-radius obtained from the S-RSPTs (I-RSPTs) is then compared with the p-radius of N and the gap between the two is recorded. In this stage, during the construction of the S-RSPTs (I-RSPTs), only one S-RSPT (I-RSPT) is constructed for each segment (intersection point). That is, ties between alternate shortest paths are broken arbitrarily. Note that this may cause the experimenter to miss an optimal tree which is an S-RSPT (I-RSPT) (if all the ties are broken inappropriately). This potential problem is addressed in Stage 2.

Stage 2: This stage is performed only on those instances for which an optimal tree could not be found in Stage 1. Given such an instance of N, the RSPTs are constructed exhaustively, i.e., all the alternative shortest paths are enumerated. The best p-radius among the RSPTs is again compared with the p-radius of N and the gap is recorded. This stage runs in exponential time since the trees so constructed involve all possible combinations of the individual alternative shortest paths. Because of this and also because solving the problem on N runs in exponential time, the maximum problem size in both stages was limited to 40 vertices due to system resource restrictions. We were also unable to find larger problem instances of the absolute p-center problem with *known* optimal solutions from the OR library (Beasley, 1990) or from other prominent researchers who have done computational work on this problem. Larger solved instances of the vertex-restricted problem are available but do not help with the absolute version of the problem.

The values of edge lengths and weights were initially designed to come uniformly from sets $\{1, 2, 3, 4, 5\}$ and $\{1, 2, 3\}$, respectively. However, these values might restrict the networks that are tested in this study to a narrow subset of the entire population of networks. This may result in ignoring some instances that do not conform to the results regarding the instances actually tested. To avoid this, the edge lengths and weights were allowed to take values, again uniformly, from a wider set of values, namely the set $\{1, 2, \ldots, 20\}$. Again, 720 instances of the problem were solved as with the previous choice of weights and edge lengths. We refer to the first set of 720 instances (with the restricted set of values for edge lengths and weights) as

P_1 instances and refer to the second set of 720 instances (with the wider set of values) as the P_2 instances. For the P_2 instances, the exhaustive stage (Stage 2) was NOT performed. In fact, the first stage applied to the P_2 instances resulted in a higher percentage of finding an optimal tree and lower values of maximum and average p-radii gaps. There were no instances that resulted in gaps larger than 100% (the largest gap in the initial set) and the overall percentage of finding an optimal tree was better than that of the initial set. Therefore, Stage 2 was not performed on this set of instances.

We have also experimented with weights and edge lengths coming from exponential and triangular distributions. In the exponential case, the mean of the edge length (weight) distribution was set equal to 3 (2) which is the same as the mean of the corresponding discrete uniform distribution. In the triangular case, the minimum, the most likely and the maximum values of the edge length (weight) distribution were set equal to 1, 3 and 5 (1, 2 and 3), respectively, which are the same as the corresponding values for the discrete uniform case. In both cases, we have performed Stage 1 and Stage 2 experiments for $n = 10$ and $n = 20$. The computational results based on these values of n indicated no significant deviations from the results that we have obtained for the discrete uniform distribution. For this reason, we report our complete results for the discrete uniform distribution case only.

Table 2 below displays the number of network spanning trees (computed using the formula given in Thulasiraman and Swamy, 1992) vs. the number of S- and I-RSPTs constructed and tested (both in the absolute and the relative sense) for P_1 instances. The figures are the average counts for the 10 instances generated for each pair of n and d. It is clear from this table that the number of the spanning trees of the networks as well as the number of S- and I-RSPTs constructed in Stage 2 grow very quickly as n and d are increased. One striking observation in this table is the ratio of the number of S- and I-RSPTs in both stages to the number of network spanning trees. This ratio is very small (practically zero) except for the $n = 10$, $d = 25\%$ instance group, for which S- and I-RSPTs included practically all spanning trees of the network in both stages. In other words, our approach relies on a very small number of S- and I-RSPTs for finding an optimal tree.

4.1. Results

All the instances tested in the experiment are grouped in the following four major categories so that the results can be analyzed with respect to four different criteria.
1. Weighted vs. unweighted,
2. Sparse vs. dense,
3. The value of p relative to n,
4. The problem size, n.

The summary tables (Tables 3–5) display the results with respect to these four groups. All the groups except the first one are further split into two subgroups as weighted (W) and unweighted (U). The results are reported for three tree classes (S-RSPTs, I-RSPTs, and BOTH) for all instances in both stages, and for a fourth class

(RANDOM) for P_1 instances in Stage 1 only. The first two of these are S-RSPTs and I-RSPTs alone. The third one corresponds to S-RSPTs and I-RSPTs combined (i.e., an optimal tree being either an S-RSPT or an I-RSPT or both). The last one (RANDOM) is the class of randomly generated trees, which we use as a basis for comparison. For this class, we generated as many trees as the number of S-RSPTs constructed in Stage 1 with the corresponding weight and edge length distributions. The random trees are constructed by starting with a complete network, and then randomly deleting one edge at a time, without violating connectivity, until the network reduces to a tree. Finally, each table contains two types of results under each group and subgroup (except for the tree class BOTH) in a particular stage (listed under SUCCESS and GAPS, respectively):

1. the percentage of instances for which an optimal tree *is* found in the set of trees tested.
2. the maximum and average gap for the instances in a particular (sub)group between the *p*-radius of the network and that of the best S-RSPT (I-RSPT).

To define the notation used in the summary tables, first let the term *success* refer to finding an optimal tree in the set of trees tested for a particular instance. The notation used in the summary tables is then defined as follows:

G	Group number,
DESCR	Symbolic description of a particular group,
SG	Subgroups of a particular group (**All**, **W**eighted and **U**nweighted),
#	Total number of instances in a particular (sub)group,
SR_i	Success ratio after Stage i ($i = 1, 2$), i.e., the *cumulative* percentage of the instances in a particular (sub)group for which an optimal tree was found in the set of trees tested,
MG_i	Maximum gap in a (sub)group of instances in Stage i. Let $\mathscr{I}$ denote a (sub)group of instances and $G_i(I)$ denote the Stage i gap (for some $I \in \mathscr{I}$) between the *p*-radius of the best S-RSPT (I-RSPT) of I and the *p*-radius of I itself. Then, MG_i of the (sub)group $\mathscr{I}$ is defined as

$$MG_i = \max_{I \in \mathscr{I}} G_i(I),$$

AG_i	Average gap in a (sub)group of instances in Stage i. Let $\mathscr{I}$ and $G_i(I)$ be defined similarly. Further, let $|\mathscr{I}|$ be the number of instances in the (sub)group $\mathscr{I}$. Then AG_i of $\mathscr{I}$ is defined as

$$AG_i = \frac{\sum_{I \in \mathscr{I}} G_i(I)}{|\mathscr{I}|}.$$

4.1.1. Results for Stage 1

The experimental results for Stage 1 are given in Tables 3 and 4 . These tables correspond to the two sets of parameters for weights and edge lengths, P_1 and P_2, as described above.

Table 2
Average number of network spanning trees and the RSPTs for instances with w and l from $\{1,\ldots,3\}$ and $\{1,\ldots,5\}$

| | | Network's sp. trees | S-RSPTs | | | | I-RSPTs – Weighted | | | | I-RSPTs – Unweighted | | | |
| | | | Stage 1 | | Stage 2 | | Stage 1 | | Stage 2 | | Stage 1 | | Stage 2 | |
n	d	#	#	%	#	%	#	%	#	%	#	%	#	%
10	25	14	13.5	98.54	13.7	100.00	13.4	97.81	13.7	100.00	13.5	98.54	13.7	100.00
	50	89,972	79	0.09	175	0.19	194	0.22	581	0.65	136.5	0.15	421	0.47
	75	5,802,924	128	0.00	481	0.01	297	0.01	984	0.02	196.4	0.00	684	0.01
20	25	4.7E+10	209	0.00	811	0.00	660	0.00	1534	0.00	312.3	0.00	723	0.00
	50	1.9E+17	439	0.00	4156	0.00	1215	0.00	7132	0.00	587	0.00	3612	0.00
	75	8.6E+20	600	0.00	29846	0.00	1568	0.00	46067	0.00	794.7	0.00	24160	0.00
30	25	4.0E+22	560	0.00	5389	0.00	1891	0.00	8923	0.00	726	0.00	3391	0.00
	50	1.6E+32	969	0.00	75017	0.00	2939	0.00	135020	0.00	1248	0.00	58019	0.00
	75	3.8E+37	1346	0.00	843104	0.00	3765	0.00	1165916	0.00	1716	0.00	529801	0.00
40	25	9.6E+35	969	0.00	52017	0.00	3325	0.00	108493	0.00	1236	0.00	40498	0.00
	50	4.9E+48	1739	0.00	667921	0.00	5471	0.00	1036925	0.00	2185	0.00	407850	0.00
	75	7.8E+55	2342	0.00	1852006	0.00	6654	0.00	2813765	0.00	2983	0.00	1290516	0.00

Table 3
Stage 1 results for S-RSPTs, I-RSPTs and random trees with w and l from $\{1,\ldots,3\}$ and $\{1,\ldots,5\}$

G	DESCR	SG	#	S-RSPTs Success SR_1	Gaps MG_1	AG_1	I-RSPTs Success SR_1	Gaps MG_1	AG_1	Both Success SR_1	Random Success SR_1	Gaps MG_1	AG_1
	Total		720	83.75	100.0	42.7	81.94	100.0	43.4	85.28	36.94	300.0	59.3
1	W		360	81.67	77.8	24.9	80.28	77.8	25.8	83.06	29.44	212.5	55.8
	U		360	85.83	100.0	64.1	83.61	100.0	64.6	87.50	44.44	300.0	62.7
2	25%	All	240	87.92	100.0	35.6	86.25	100.0	34.6	88.75	50.00	133.3	31.4
		W	120	87.50	77.8	28.7	85.83	77.8	27.3	88.33	48.33	108.3	30.2
		U	120	88.33	100.0	43.1	86.67	100.0	42.4	89.17	51.67	133.3	32.6
	50%	All	240	83.33	100.0	42.6	81.67	100.0	44.6	85.42	33.33	300.0	60.9
		W	120	80.83	50.0	24.4	80.00	66.7	26.6	82.50	25.00	212.5	61.2
		U	120	85.83	100.0	67.1	83.33	100.0	66.2	88.33	41.67	300.0	60.6
	75%	All	240	80.00	100.0	46.5	77.92	100.0	48.5	81.67	27.50	300.0	85.4
		W	120	76.67	50.0	25.8	75.00	50.0	24.7	78.33	15.00	212.5	76.1
		U	120	83.33	100.0	76.2	80.83	100.0	78.6	85.00	40.00	300.0	94.8
3	n/4	All	240	70.00	100.0	36.1	67.50	100.0	36.2	72.92	11.25	300.0	100.9
		W	120	66.67	50.0	25.7	65.00	50.0	25.4	69.17	12.50	212.5	89.2
		U	120	73.33	100.0	49.1	70.00	100.0	48.9	76.67	10.00	300.0	112.5
	n/2	All	240	84.58	100.0	57.2	81.67	100.0	58.2	86.25	24.17	200.0	68.2
		W	120	84.17	66.7	27.1	81.67	66.7	27.8	85.83	23.33	150.0	62.5
		U	120	85.00	100.0	88.9	81.67	100.0	88.6	86.67	25.00	200.0	73.9
	3n/4	All	240	96.67	100.0	31.6	96.67	100.0	31.6	96.67	75.42	100.0	8.7
		W	120	94.17	77.8	21.8	94.17	77.8	21.8	94.17	52.50	77.3	15.8
		U	120	99.17	100.0	100.0	99.17	100.0	100.0	99.17	98.33	100.0	1.7
4	n = 10	All	180	95.56	100.0	41.6	95.00	100.0	42.0	96.11	72.22	100.0	12.2
		W	90	94.44	77.8	31.6	95.56	77.8	38.2	95.56	72.22	68.0	10.3
		U	90	96.67	100.0	58.3	94.44	100.0	45.0	96.67	72.22	100.0	14.0
	n = 20	All	180	86.67	100.0	29.5	85.00	100.0	33.4	88.33	36.67	250.0	44.0
		W	90	85.56	33.3	19.6	83.33	33.3	20.5	86.67	32.22	150.0	43.0
		U	90	87.78	100.0	44.2	86.67	100.0	49.6	90.00	41.11	250.0	45.0

Table 3 (continued)

G	DESCR	SG	#	S-RSPTs			I-RSPTs			Both	Random		
				Success	Gaps		Success	Gaps		Success	Success	Gaps	
				SR_1	MG_1	AG_1	SR_1	MG_1	AG_1	SR_1	SR_1	MG_1	AG_1
$n=30$	All	180		81.11	100.0	43.9	78.33	100.0	43.8	82.78	22.22	300.0	77.9
	W	90		81.11	60.0	24.2	77.78	50.0	24.2	82.22	11.11	205.4	70.8
	U	90		81.11	100.0	64.2	78.89	100.0	64.5	83.33	33.33	300.0	85.0
$n=40$	All	180		71.67	100.0	47.1	69.44	100.0	48.2	73.89	16.67	300.0	103.0
	W	90		65.56	66.7	28.6	64.44	66.7	27.7	67.78	2.22	212.5	99.2
	U	90		77.78	100.0	75.8	74.44	100.0	75.8	80.00	31.11	300.0	106.8

Table 4
Stage 1 results for S-RSPTs and I-RSPTs with w and l from {1,…, 20}

G	DESCR	SG	#	S-RSPTs Success SR$_1$	S-RSPTs Gaps MG$_1$	S-RSPTs AG$_1$	I-RSPTs Success SR$_1$	I-RSPTs Gaps MG$_1$	I-RSPTs AG$_1$	Both Success SR$_1$
	Total		720	87.78	50.0	18.0	86.67	66.7	18.2	88.89
1	W		360	88.06	30.6	13.3	86.94	30.6	12.6	88.89
	U		360	87.50	50.0	22.3	86.39	66.7	23.3	88.89
2	25%	All	240	91.25	33.3	13.2	90.83	33.3	13.6	92.50
		W	120	90.83	27.0	12.5	90.00	27.0	11.7	93.33
		U	120	91.67	33.3	13.9	91.67	33.3	15.9	91.67
	50%	All	240	85.42	50.0	18.4	84.17	50.0	18.4	86.25
		W	120	86.67	30.6	15.5	85.83	30.6	16.0	86.67
		U	120	84.17	50.0	20.8	82.50	50.0	20.4	85.83
	75%	All	240	86.67	50.0	20.6	85.00	66.7	20.7	87.92
		W	120	86.67	26.6	11.7	85.00	26.6	10.0	86.67
		U	120	86.67	50.0	28.6	85.00	66.7	31.3	89.17
3	n/4	All	240	70.00	50.0	16.9	68.33	66.7	17.1	72.50
		W	120	72.50	26.6	12.4	70.83	27.4	11.9	75.00
		U	120	67.50	50.0	20.7	65.83	66.7	21.4	70.00
	n/2	All	240	93.75	50.0	23.3	92.08	50.0	23.1	94.58
		W	120	92.50	30.6	17.1	90.83	22.8	14.9	92.50
		U	120	95.00	50.0	30.3	93.33	50.0	35.9	96.67
	3n/4	All	240	99.58	12.5	12.5	99.58	12.5	12.5	99.58
		W	120	99.17	12.5	12.5	99.17	12.5	12.5	99.17
		U	120	100.00	–	–	100.00	–	–	100.00
4	n = 10	All	180	96.11	13.0	7.7	96.11	13.0	8.6	96.11
		W	90	98.89	2.9	2.9	98.89	2.9	2.9	98.89
		U	90	93.33	13.0	8.6	93.33	13.0	9.5	93.33
	n = 20	All	180	90.56	33.3	13.4	88.89	33.3	14.9	90.56
		W	90	91.11	26.6	8.6	91.11	26.6	8.6	91.11
		U	90	90.00	33.3	17.8	86.67	33.3	19.1	90.00
	n = 30	All	180	86.67	50.0	20.1	85.56	33.3	18.4	90.00
		W	90	87.78	25.9	15.4	85.56	25.3	14.3	91.11

Table 4 (continued)

G	DESCR	SG	#	S-RSPTs			I-RSPTs			Both
				Success	Gaps		Success	Gaps		Success
				SR_1	MG_1	AG_1	SR_1	MG_1	AG_1	SR_1
	$n=40$	U	90	85.56	50.0	23.8	85.56	33.3	22.4	88.89
		All	180	77.78	50.0	20.4	76.11	66.7	21.1	78.89
		W	90	74.44	30.6	14.5	72.22	30.6	13.4	74.44
		U	90	81.11	50.0	28.0	80.00	66.7	31.9	83.33

Table 5
Stage 2 results for S-RSPTs and I-RSPTs with w and l from $\{1,\ldots,3\}$ and $\{1,\ldots,5\}$

G	DESCR	SG	#	S-RSPTs			I-RSPTs			Both
				Success SR_2	Gaps MG_2	AG_2	Success SR_2	Gaps MG_2	AG_2	Success SR_2
1	Total		720	96.11	100.0	7.8	93.89	100.0	9.7	96.67
	W		360	95.56	33.3	4.7	93.61	33.3	5.6	96.39
	U		360	96.67	100.0	11.7	94.17	100.0	14.6	96.94
2	25%	All	240	97.92	25.0	3.4	95.42	25.0	5.8	97.92
		W	120	97.50	11.1	1.8	95.00	20.0	4.2	97.50
		U	120	98.33	25.0	5.0	95.83	25.0	7.5	98.33
	50%	All	240	94.58	100.0	10.7	92.92	100.0	11.1	95.42
		W	120	94.17	25.0	5.9	93.33	25.0	5.5	95.83
		U	120	95.00	100.0	18.2	92.50	100.0	17.9	95.00
	75%	All	240	95.83	100.0	7.9	93.33	100.0	10.9	96.67
		W	120	95.00	33.3	5.3	92.50	33.3	6.5	95.83
		U	120	96.67	100.0	11.7	94.17	100.0	16.7	97.50
3	n/4	All	240	89.58	100.0	10.4	85.00	100.0	13.7	90.83
		W	120	88.33	33.3	6.3	85.00	33.3	7.3	90.83
		U	120	90.83	100.0	15.5	85.00	100.0	21.2	90.83
	n/2	All	240	98.75	100.0	4.3	96.67	100.0	4.4	99.17
		W	120	98.33	33.3	3.1	95.83	33.3	4.2	98.33
		U	120	99.17	100.0	5.6	97.50	100.0	4.5	100.00
	3n/4	All	240	100.00	0.0	0.0	100.00	0.0	0.0	100.00
		W	120	100.00	0.0	0.0	100.00	0.0	0.0	100.00
		U	120	100.00	0.0	0.0	100.00	0.0	0.0	100.00
4	n=10	All	180	100.00	0.0	0.0	100.00	0.0	0.0	100.00
		W	90	100.00	0.0	0.0	100.00	0.0	0.0	100.00
		U	90	100.00	0.0	0.0	100.00	0.0	0.0	100.00
	n=20	All	180	97.78	33.3	4.1	93.78	50.0	9.8	98.33
		W	90	97.78	11.1	1.6	94.44	25.0	5.2	98.89
		U	90	97.78	33.3	7.1	93.33	50.0	15.6	97.78

Table 5 (*continued*)

G	DESCR	SG	#	S-RSPTs Success	Gaps		I-RSPTs Success	Gaps		Both Success
				SR_2	MG_2	AG_2	SR_2	MG_2	AG_2	SR_2
	$n=30$	All	180	95.56	50.0	6.3	92.78	50.0	7.7	96.67
		U	90	95.56	50.0	8.3	93.33	50.0	10.5	95.56
	$n=40$	All	180	91.11	100.0	11.6	88.89	100.0	12.6	91.67
		W	90	88.89	33.3	7.0	87.78	33.3	6.8	88.89
		U	90	93.33	100.0	18.8	90.00	100.0	20.7	94.44

In this stage, the S-RSPTs do not necessarily contain an optimal tree in all instances. We observed instances for which the best S-RSPT did not give the p-radius of the corresponding network. However, the computational evidence suggests that the S-RSPTs include an optimal tree in a majority of the cases. With P_1 (Table 3), we observe that the success ratio in Stage 1 is 83.75%. We also observe that, for all of the subgroups, unweighted instances give better figures of SR_1, compared to weighted instances. With P_2 (Table 4), Stage 1 success ratio is 87.78% and unweighted instances perform roughly the same as weighted instances. In terms of the maximum gaps, no gap higher than 100% (50%) of the network p-radius is observed for set $P_1(P_2)$. The AG figures suggest that the instances coming from set P_2 give better results in terms of the performance measures. Contrary to the success ratios, the U instances have worse (higher) maximum and average gaps compared to the W instances with both P_1 and P_2.

The results for S-RSPTs with respect to groups 2, 3 and 4 are summarized as follows:

(2) A general pattern is that the success ratios decrease as the density increases. With P_1, the success ratios are consistently higher for the U subgroups, but with P_2, neither U nor W subgroup outperforms the other. The MG does not change with density at all, but AG apparently increases as the density increases, especially for the U subgroups. The S-RSPTs seem to perform better on relatively sparse and unweighted instances.

(3) The success ratios increase as p gets nearer to n. Again, success ratios are generally higher for the U instances, however, with $p = \lfloor n/4 \rfloor$, SR_1 is well below the overall SR_1. The least improvements in maximum and average gap occur again with this case, which implies that S-RSPTs show relatively poor performance for small values of p/n.

(4) As n increases, all the success ratios decrease and the amount of maximum and average gaps increases with few exceptions. Although the S-RSPTs again perform better for the U instances in terms of success ratios, the distinction is not very clear with P_2.

The Stage 1 experimental results for I-RSPTs are also in Tables 3 and 4. Again, we observe that an optimal tree is not always included in the set of I-RSPTs. However, similar to the S-RSPTs, the observed results suggest that the I-RSPTs include an optimal tree most of the time.

The I-RSPTs give results similar to those of S-RSPTs in the other performance measures. All the major patterns observed with S-RSPTs are also valid for I-RSPTs. However, the performance of I-RSPTs is somewhat worse than that of S-RSPTs in terms of success ratios, and maximum and average gaps. The only apparent performance difference between the two classes of RSPTs is in the fourth group with P_2. In this case, W instances perform better, on the average, than the U instances with I-RSPTs.

When S-RSPTs and I-RSPTs are considered together, i.e., when we search for the optimal trees *either* in the set of S-RSPTs *or* the set of I-RSPTs, the success ratios improve slightly. In this case, the increase in success ratios is up to 3% in some subgroups. Over all, the success ratio increases from 83.75% to 85.28% with P_1, and

from 87.78% to 88.89% with P_2. While there is a slight improvement, we find these figures to be an indicator of the fact that S-RSPTs and I-RSPTs succeed or fail on essentially the same instances most of the time.

The last class of trees tested is the class of randomly generated trees (results reported for P_1 instances only). In our experimental study, we observe that S-RSPTs and I-RSPTs significantly outperform randomly generated trees in all performance measures. An interesting observation with these trees, however, is that the pattern of their performance within each group and subgroup is very similar to that of the S- and I-RSPTs.

4.1.2. Results for Stage 2

The experimental results for Stage 2 are provided in Table 5. As we mentioned earlier, this stage is performed for instances with P_1 set of parameters only. We observe that the second stage increases the overall success ratio from 83.75% to 96.11% for S-RSPTs, from 81.94% to 93.89% for I-RSPTs, and from 85.28% to 96.67% for S- and I-RSPTs combined. In other words, only 3.33% of the P_1 instances have an optimal tree which is neither an S- nor an I-RSPT. The increase in the success ratio results from an exhaustive enumeration of all alternate shortest paths, which runs in exponential time. To achieve higher success ratios in Stage 1, one must develop a better way to break ties between alternate shortest paths.

The second stage decreases the gaps considerably within each group and subgroup for both S- and I-RSPTs. The overall maximum gap is still 100%, but in many subgroups, maximum gap is reduced if not completely eliminated. The overall average gap is also decreased drastically in Stage 2 from 42.7% to 7.8% for S-RSPTs and from 43.4% to 9.7% for I-RSPTs. This reflects a reduction in average gaps within all of the subgroups as well.

The observations from Stage 1 regarding the performance of S- and I-RSPTs within each subgroup are mostly valid in the second stage, too. In general, the performance of the RSPTs decreases as the edge density increases, as p gets closer to n, and as the problem size increases. The U instances give better success ratios, but higher maximum and average gaps compared to the W instances. The S-RSPTs are again slightly better in performance compared to I-RSPTs. When considered together, the two classes of RSPTs have slightly better success ratios than each of them alone, but the positive effect of combining the two classes is less compared to that in Stage 1. We have also experimented with random trees for P_1 instances with $n = 10$ and $n = 20$ using as many random trees as the number of RSPTs in Stage 2. Even though this causes a substantial increase in the tested number of random trees as compared to Stage 1, we observed that the increased number of tests does not at all improve the success ratios for random trees (with a few exceptions).

Note that the set of S- and I-RSPTs tested in both stages is a very small subset of all spanning trees of the associated network instances. Because of this, we find S- and I-RSPTs to be very successful in determining optimal trees, even without the exhaustive Stage 2. Even though Stage 2 runs in exponential time, it is still preferable to implement this stage rather than enumerating all spanning trees of a network.

One major observation we have made so far is that, for either of the sets P_1 and P_2 in both stages, U instances give larger maximum and average gaps than the corresponding W instances. This is possibly due to the following: for U instances, a candidate p-radius value associated with an intersection point either has a fractional part of 0.5 or is integer. This follows from the fact that the edge lengths in the tested instances are integers and each candidate p-radius value, generated by a pair of vertices v_i, v_j, is just half the distance between v_i and v_j. Because of this, whenever a gap occurs between a network p-radius and a tree p-radius, it is a positive integer multiple of 0.5 and this may cause the gap to be high when the network p-radius is small (e.g., if the network p-radius is 0.5 and tree p-radius is the possible smallest value with a gap, which is 1, then the gap is 100% in terms of network p-radius). In short, the candidate p-radius values for the U instances come from a more restricted set of discrete values than those for the W instances. A similar argument may be used to explain why the maximum and average gaps decrease as we move from P_1 to P_2. Since the weights and edge lengths come from a larger set of values, we speculate that the candidate p-radius values associated with intersection points may have more variety in their decimal parts, which would possibly imply smaller difference, on the average, between any two consecutive candidate p-radius values.

The following is a summary of the experimental results:

- The set of optimal trees *need not* have an intersection with S- or I-RSPTs.
- Both types of RSPTs provide an optimal tree in a majority of the cases, but in general S-RSPTs supply an optimal tree in more cases than I-RSPTs. Taking the union of the two improves the results slightly.
- Overall, the S- and I-RSPTs constitute an extremely small subset of all the spanning trees of a network, but supply an optimal tree in about 85% of the cases if ties for shortest paths are arbitrarily broken and in about 96% of the cases if all shortest paths are individually taken into account. This is indicative of a very high success rate for the RSPTs.
- The trees perform better on relatively sparse networks.
- The trees perform better as p/n approaches 1.
- The success ratio decreases as the problem size increases.
- Unweighted instances give better results than weighted instances with respect to success ratios, but they have higher maximum and average gaps.
- Making a weighted instance unweighted does not always help to find an optimal tree. There were some unweighted instances that failed to find an optimal tree whereas the corresponding weighted instance had a success. Note that this may hold regardless of whether or not U instances have lower maximum and average gaps than W instances.

5. Conclusion

In this paper, we have discussed the spanning tree approach to solving the absolute p-center problem on cyclic networks. The problem is known to be NP-hard on

cyclic networks (Kariv and Hakimi, 1979) but polynomial-time solvable on tree networks. The new approach is to identify a spanning tree of a particular network whose *p*-center and *p*-radius are also optimal for the cyclic network.

First, for an arbitrary network *N*, we prove the existence of an optimal tree. This implies that the instances which have a polynomial number of spanning trees can be solved efficiently by constructing all the spanning trees. For arbitrary networks, enumeration is too costly, and it is advisable to search for an optimal tree in a polynomial-size subset of the spanning trees since the entire set is of exponential size. Note that the identification of a subset, in polynomial time, which will always contain an optimal tree would imply P = NP and hence is likely impossible. In this paper, we introduced two types of spanning trees that are suspected of containing an optimal tree. Both types of trees are shortest path trees rooted at some point(s) of the network. The first type, S-RSPTs, are trees rooted at segments defined by adjacent elements of the extended vertex set that includes vertices and antipodals of vertices. The second type, I-RSPTs, are rooted at intersection points of pairs of weighted distance functions whenever these functions have slopes of opposite sign. The total number of S-RSPTs and I-RSPTs that are constructed are $O(n|E|)$ and $O(n^2|E|)$, respectively. However, alternate shortest paths may be encountered in the construction phase which may increase the number of trees exponentially. We keep the number of trees polynomial by picking only one path from alternative shortest paths. Overall, the number of S- and I-RSPTs is very small compared to the number of all spanning trees of a network.

In the last section of the paper, we presented an experimental search for the optimal tree in the two sets of spanning trees, S- and I-RSPTs. A total of 1440 instances of the problem were generated and tested. The results indicate that, even though the S-RSPTs and I-RSPTs do not always include an optimal tree, they do most of the time. Furthermore, we observed that the maximum gap between the *p*-radius of the network and the *p*-radii of the trees do not exceed 100% of the *p*-radius of the network. This finding suggests that the worst case deviation of the proposed classes of trees from optimality may be theoretically bounded above by 100%. The 100% bound has already been reported in the literature for other heuristics, e.g., Hochbaum and Shmoys' (1985) heuristic for the vertex-restricted *p*-center problem which is later extended by Plesnik (1987) to the absolute *p*-center problem, and the approximation algorithms of Hochbaum and Pathria (1997) for vertex-restricted versions of the "set" *p*-center problem. Our approach has nothing in common with these heuristics, but the existence of worst case bounds for these heuristics suggests that the same bound may be valid for the spanning tree approach that we propose in this paper. Also, the S-RSPTs and I-RSPTs are observed to perform better (a) on sparse instances of networks, (b) with larger values of *p* and (c) on smaller-sized instances of networks. We have also compared our results with those of randomly generated trees, and found that the RSPTs significantly outperform the random trees. Since the sets of S-RSPTs and I-RSPTs do not always provide an optimal tree, the problem of finding a polynomial subset of spanning trees that include an optimal tree remains as an unsolved problem.

References

Beasley, J.E., 1990. OR-Library: distributing test problems by electronic mail. J. Oper. Res. Soc. 41 (11), 1069–1072.

Busacker, R.G., Saaty, T.L., 1965. Finite Graphs and Networks. McGraw-Hill, New York.

Chandrasekaran, R., Daughety, A., 1981. Location on tree networks: p-centre and N-dispersion problems. Math. Oper. Res. 6, 50–57.

Chandrasekaran, R., Tamir, A., 1980. An $O((n \log p)^2)$ algorithm for the continuous p-center problem on a tree. SIAM J. Algebraic Discrete Meth. 1, 370–375.

Cristofides, N., Viola, P., 1971. The optimum location of multi-centers on a graph. Oper. Res. Quart. 22, 145–154.

Dearing, P.M., Francis, R.L., 1974. A minimax location problem on a network. Transportation Sci. 8, 333–343.

Hakimi, S.L., 1964. Optimal locations of switching centers and the absolute centers and medians of a graph. Oper. Res. 12, 450–459.

Hakimi, S.L., 1965. Optimum distribution of switching centers in a communication network and some related graph theoretic problems. Oper. Res. 13, 462–475.

Hakimi, S.L., Schmeichel, E.F., Pierce, J.G., 1978. On p-centers in networks. Transportation Sci. 12, 1–15.

Handler, G.Y., 1973. Minimax location of a facility in an undirected tree graph. Transportation Sci. 7, 287–293.

Handler, G.Y., 1978. Finding two-centers of a tree; the continuous case. Transportation Sci. 12, 93–106.

Hedetniemi, S.M., Cockayne, E.J., Hedetniemi, S.T., 1981. Linear algorithms for finding the jordan center and path center of a tree. Transportation Sci. 15, 98–114.

Hochbaum, D.S., Pathria, A., 1997. Generalized p-center problems: Complexity results and approximation algorithms. European J. Oper. Res. 100 (3), 594–607.

Hochbaum, D.S., Shmoys, D.B., 1985. A best possible heuristic for the k-center problem. Math. Oper. Res. 10 (2), 180–184.

Hooker, J., 1986. Solving nonlinear single-facility network location problems. Oper. Res. 34, 732–743.

Hooker, J., 1989. Solving nonlinear multiple-facility network location problems. Networks 19, 117–133.

Hooker, J.N., Garfinkel, R.S., Chen, C.K., 1991. Finite dominating sets for network location problems. Oper. Res. 39 (1), 100–118.

Kariv, O., Hakimi, S.L., 1979. An algorithmic approach to network location problems. Part I: The p-centers. SIAM J. Appl. Math. 37, 513–538.

Kirchoff, G., 1847. Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme gefürt wird. Ann. Phys. Chem. 72, 497–508.

Megiddo, N., Tamir, A., Zemel, E., Chandrasekaran, R., 1981. An $O(n \log^2 n)$ algorithm for the kth longest path in a tree with applications to location problems. SIAM J. Computing 10, 328–337.

Minieka, E., 1970. The m-center problem. SIAM Rev. 12, 138–139.

Minieka, E., 1980. Conditional centers and medians of a graph. Networks 10, 265–272.

Minieka, E., 1981. A polynomial time algorithm for finding the absolute center of a network. Networks 11, 351–355.

Mirchandani, P.B., Francis, R.L. (Eds.), 1990. Discrete Location Theory. Wiley, New York.

Moon, J.W., 1967. Various proofs of Cayley's formula for counting trees. In: Harary, F., Beineke, L. (Eds.), A Seminar on Graph Theory. Holt, Rinehart & Winston, New York.

Plesnik, J., 1987. A heuristic for the p-center problem in graphs. Discrete Appl. Math. 17, 263–268.

Riordan, J., 1958. An Introduction to Combinatorial Analysis. Wiley, New York.

Tamir, A., 1985. A finite algorithm for the continuous p-center location problem on a graph. Math. Programming 31, 298–306.

Tansel, B.C., Francis, R.L., Lowe, T.J., 1983. Location on networks: A survey, Part I: the p-center and p-median problems. Management Sci. 29 (4), 482–497.

Tansel, B.C., Francis, R.L., Lowe, T.J., 1990. Duality: Covering and constraining p-center problems on trees. In: Mirchandani, P.B., Francis, R.L. (Eds.), Discrete Location Theory. Wiley Interscience, New York, pp. 349–386.

Tansel, B.C., Francis, R.L., Lowe, T.J., Chen, M.L., 1982. Duality and distance constraints for the non-linear *p*-center problem and covering problem on a tree network. Oper. Res. 30, 725–743.

Thulasiraman, K., Swamy, M.N.S., 1992. Graphs: Theory and Algorithms. Wiley Interscience, New York.

Toregas, C., Swain, R., ReVelle, C., Bergman, L., 1971. The location of emergency service facilities. Oper. Res. 19, 1363–1373.