

THE P-CENTER AND THE P-MEDIAN PROBLEMS
IN GRAPHS WITH SMALL NUMBER
OF SPANNING TREES

By

MOHAMAD H. HASSAN
Department of Mathematics, Faculty of Science,
University of Qatar, Doha, Qatar

مشكلة إيجاد p - مركز و p - وسط لبيان

يحتوي عدد صغير من الهياكل

محمد حسن حسن

يعرض هذا البحث عدد من الخوارزميات مع حساب تعقيدها وهي :

(١) خوارزمية لإيجاد مجموعة التغطية بنصف قطر r في بيان مقيم الرؤوس بعدد صغير من الهياكل . تعقيد هذه الخوارزمية $O(m.n)$ بالنسبة لبيان وحيد الحلقة .

(٢) خوارزمية لإيجاد p - مركز ، مطلق أو رأسي في بيان مقيم الرؤوس يحتوي عدد صغير من الهياكل . تعقيد هذه الخوارزمية $O(m.n^2 \cdot \lg n)$ بالنسبة لإيجاد p - مركز مطلق و $O(m.n^2 \cdot \lg n)$ بالنسبة لإيجاد p - مركز رأسي في بيان يحتوي حلقة واحدة .

(٣) خوارزمية لإيجاد p - وسط في بيان مقيم الرؤوس بعدد صغير من الهياكل . تعقيد هذه الخوارزمية $O(m.n^2 \cdot p^2)$ في بيان وحيد الحلقة .

Key Words: P. center, P. median, Algorithm

ABSTRACT

In this paper, we will describe some algorithms and give their complexity as following:

- (1) The algorithm for finding a dominating set of radius r in a vertex-weighted graph with small number of spanning trees. The complexity of this algorithm for the unicyclic graph is $O(m.n)$.
- (2) The algorithm for finding an absolute and vertex p -center of a vertex-weighted graph with small number of spanning trees. The complexity of determining the p -center is $O(m.n^2 \lg n)$ for absolute (resp., $O(n^2 \lg n)$ for vertex) p -center in unicyclic graphs.
- (3) The algorithm for finding a p -median in a vertex-weighted graph with small number of spanning trees. The complexity of this algorithm for the class of unicyclic graphs is $O(m.n^2.p^2)$.

1. INTRODUCTION.

The terminology used in this paper is standard and follows that of [1]. A network is a connected finite graph $G = (V, E)$ with a nonnegative number $w(v)$ (called the weight of v) associated with each of its $|V| = n$ vertices, and a positive number $l(e)$ (called the length of e) associated with each of its $|E|$ edges (I. e., $l(e) = c(v_r, v_s)$, where $e = (v_r, v_s)$ and $c(v_r, v_s)$ is the length of each (v_r, v_s)). Let $X_p = \{x_1, x_2, \dots, x_p\}$ be a set of p points on G , where by a point on G we mean a point along any edge of G which may or not be a vertex of G . We

define the distance $d(v, X_p)$ between a vertex v of G and a set X_p on G by

$$(1.1) \quad d(v, X_p) = \min_{1 \leq i \leq p} \{d(v, x_i)\}$$

where $d(v, x_i)$ is the length of a shortest path in G between vertex v and point x_i (x_i can be considered as a new vertex inserted into the edge e). Let

$$(1.2) \quad F_G(X_p) = \max_{v \in V} \{w(v) \cdot d(v, X_p)\}.$$

Let X_p^* be such that

$$(1.3) \quad F_G(X_p^*) = \min_{X_p \text{ on } G} \{F(X_p)\}.$$

Then X_p^* is called an absolute p-center of G and $F_G(X_p^*)$ is called the absolute p-radius of G. If X_p and X_p^* in (1.3) are restricted to be sets of p vertices of G, then X_p^* is called a vertex p-center and $F_G(X_p^*)$ is called the vertex p-radius of G.

Further we define

$$(1.4) \quad H_G(X_p) = \sum_{v \in V} w(v).d(v, X_p).$$

We call $H_G(X_p)$ the distance-sum of the set (X_p) .

If X_p^* on G is such that

$$(1.5) \quad H_G(X_p^*) = \min_{X_p \text{ on } G} \{H(X_p)\},$$

then X_p^* is called a p-median of G [2], [3]. Hakimi [3] has shown that there exists a set of p vertices $V_p^* \subset V$ such that

$$H_G(V_p^*) = H_G(X_p^*).$$

If all the vertices of the graph $G = (V, E)$ have the same weight q, then without loss of generality we shall assume that $q = 1$ and we refer G as a vertex-unweighted graph. Otherwise we say that G is a vertex-weighted graph. We shall assume that $p < n$, since if $n = p$ then $X_p^* = V$, $F_G(X_p^*) = 0$, $H_G(X_p^*) = 0$ while $p > n$ has no mathematical significance. We further assume that the graph G contains neither loops nor multiple edges. Finally, we assume that for each edge $e = (v_r, v_s)$ the length of e is equal to the distance between v_r and v_s (I. e., $l(e) = d(v_r, v_s)$), because otherwise, the edge e could be eliminated without affecting the p-radius of G.

The inverse of the p-center problem is defined as follows: Given a graph $G = (V, E)$ and a positive integer r, find the smallest positive integer p such that the p-radius of G is not greater than r. This number p is called the absolute domination number while a set X_p of p points such that $F_G(X_p) \leq r$ is called an absolute dominating set of radius r. The vertex domination number of radius r and the vertex dominating set of radius r are similarly defined.

The problem of finding a p-center of G was originated by Hakimi [2], [3] and is discussed in papers [4], [5], [6], [7], [8] and [9].

Kariv and Hakimi in [10] described an $O(|E|.n \lg n)$ algorithm for finding an absolute 1-center in a vertex weighted graph, and an $O(|E|.n + n^2 \lg n)$ algorithm for finding an absolute 1-center in a vertex-unweighted graph with the assumption that the distance-matrix of the graph is known. Also Kariv and Hakimi described in [10] an

$O(|E|^p . n^{2p-1} / (p-1)! \lg n)$ algorithm (resp., an $O(|E|^p . n^{2p-1} / (p-1)!)$ algorithm) for finding an absolute p-center ($1 < p < n$) in a vertex weighted (resp., vertex-unweighted) network.

In the case, when the graph G is a tree, Kariv and Hakimi in [10] described an $O(n \lg n)$ algorithm for finding the (vertex or absolute) 1-center of vertex-weighted tree and an $O(n)$ algorithm for finding a dominating set of radius r in tree, also an $O(n^2 \lg n)$ algorithm for finding a (vertex or absolute) p-center ($1 < p < n$) of a vertex-weighted tree.

In the case, G is unicyclic graph, M. H. HASSAN in [11] described an $O(n)$ algorithm for finding a dominating set of radius r, and algorithm $O(n^3 \lg n)$ (resp., $O(n^2 \lg n)$) for finding absolute (resp., vertex) p-center of a vertex-weighted graph G ($1 < p < n$).

Goodman in [12] gave an $O(n)$ algorithm for finding a 1-median of tree, while Kariv and Hakimi in [13] described an $O(n^2 p^2)$ algorithm for finding a p-median of tree, where $p > 1$.

Hassan, M. H. gave in [14] an important results for determining a certain bounds of the p-radius (the minimum distance sum of p-median, resp.) of the class of cactus graphs.

2. A dominating set of radius r in graphs with small number of spanning trees:

Now in this part of this section we will show and prove the more general algorithm for finding the dominating set in arbitrary graph which is very useful in the case of classes of graphs with small number of spanning trees.

Lemma 2.1.

Let $G = (V, E)$ be a connected graph with edge length $l(e) > 0$ and vertex weight $w(v) \geq 0$. Let there exist a dominating set of p points of radius r on G. Then there exists a dominating set $X_p = (x_1, x_2, \dots, x_p)$ of radius r on G such that

(2.1) if x_i for some i is an internal point of some edge (v_{i_1}, v_{i_2}) then

$$d(v_{i_1}, (X_p - \{x_i\})) > d(v_{i_1}, x_i) \text{ and also}$$

$$d(v_{i_2}, (X_p - \{x_i\})) > d(v_{i_2}, x_i).$$

Proof

We prove this lemma by contradiction. Assume that there is no dominating set X_p of radius r on G fulfilling to (2.1) and let $Y_p = (y_1, y_2, \dots, y_p)$ be a dominating set of radius r on G such that this set Y_p has the least number k of points not satisfying condition (2.1) from all the dominating sets of p points of radius r on G.

Let $k \neq 0$ and let $y_s \in Y_p$ be an internal point of edge (v_1, v_2) which does not satisfies (2.1). Without loss of generality we can assume that

$$d(v_1, y_s) \geq d(v_1, (Y_p - \{y_s\})) = d(v_1, y_t), \text{ where } y_t \text{ is some vertex of } (Y_p - y_s).$$

We define $Y'_p = (y'_1, y'_2, \dots, y'_p)$ in such a way, that $y'_s = y_i$ for all $i \neq s$ and $y'_1 = v_2$. We divide $V = V_1 \cup V_2$, $V_1 \cap V_2 = \phi$ so that

$$V_1 = \{v \in V ; d(v, y_s) = d(y_s, v_1) + d(v_1, v) < d(y_s, v_2) + d(v, v_2)\} \text{ and}$$

$$V_2 = \{v \in V ; d(v, y_s) = d(y_s, v_2) + d(v_2, v) \leq d(y_s, v_1) + d(v, v_1)\}$$

(2.2) For all $i \neq s$ and every vertex $v \in V$ it holds

$$d(y'_i, v) = d(y_i, v)$$

(2.3) For every vertex $v \in V_2$ it holds

$$d(y'_s, v) = d(v_2, v) \leq d(v_2, v) + d(v_2, y_s) = d(y_s, v)$$

(2.4) For every vertex $v \in V_1$ it holds

$$d(y'_t, v) \leq d(y'_t, v_1) + d(v_1, v) = d(y_t, v_1) + d(v_1, v) \leq d(y_s, v_1) + d(v, v_1) = d(v, y_s).$$

From (2.2), (2.3) and (2.4) it follows that for all $v \in V$ holds

$$d(v, Y'_p) \leq d(v, Y_p), \text{ i.e. } Y'_p \text{ is a dominating set of radius } r,$$

too, and there are less point in Y'_p not satisfying (2.1) than in

Y_p .

This is a contradiction. Hence there exists a dominating set Y_p of radius r , all points of which satisfy the condition (2.1).

Lemma 2.2

Let $G = (V, E)$ be a connected graph with given weight w of vertices and length l of edges. Let there exist a dominating set of p points of radius r on G . Then there exists a dominating set $X_p = (x_1, x_2, \dots, x_p)$ of radius r on G such that

(2.5) If x_i, x_j for some i, j are internal point of some edges e_i, e_j resp., then e_i and e_j are not adjacent.

Proof

From lemma 2.1 it follows that there exists a dominating set $X_p = (x_1, x_2, \dots, x_p)$ of the radius r on G with the property (2.1).

We shall show that then X_p has the property (2.5), too. We shall prove this fact by contradiction.

Let X_p have not the property (2.5), then there exist internal points $x_s \in e_s, x_t \in e_t$ in G such that e_s and e_t are adjacent, i.e. there exist vertices $v_o, v_s, v_t \in V$ such that $e_s = (v_o, v_s), e_t = (v_o, v_t)$. without loss of generality we can assume that $d(x_s, v_o) \leq d(x_t, v_o)$ and then

$d(v_o, (X_p - \{x_t\})) \leq d(v_o, x_s) \leq d(v_o, x_t)$ and then X_p does not satisfy (2.1) which is a contradiction proving the lemma.

Lemma 2.3

Let $G = (V, E)$ be a connected graph with a vertex weight w and edge length l . Let $X_p = (x_1, x_2, \dots, x_p)$ be a set of p points on G such that for every internal point $x_i \in X_p$ of an edge (v_{i_1}, v_{i_2}) it holds

$$d(v_{i_1}, (X_p - \{x_i\})) > d(v_{i_1}, x_i) \text{ and also}$$

$$d(v_{i_2}, (X_p - \{x_i\})) > d(v_{i_2}, x_i).$$

Then there exists a spanning tree T of G such that all the points of X_p lie on T , too, and that $d_G(v, X_p) = d_T(v, X_p)$ for all $v \in V$.

Proof

We shall construct the spanning tree T using induction in finite number of steps.

Step 0 :

We define $T_0 = (V(T_0), E(T_0))$, where

$$V(T_0) = \{v \in V; \exists x_i \in X_p, x_i \equiv v \text{ or } \exists x_j \in X_p$$

and $v' \in V$ so that x_j is internal point of the edge $(v, v')\}$

$E(T_0) = \{e \in E; \exists x_i \in X_p \text{ such that } x_i \text{ is an internal point of the edge } e\}$

It follow from the proof of lemma 2.2 the T_0 is a forest with maximum vertex degree 1, because no two edges in T_0 are adjacent. It follows from the assumption that for every

$$v \in V(T_0) \text{ holds } d_{T_0}(v, X_p) = d_G(v, X_p)$$

Step k:

Let T_{k-1} be such a forest that for all $v \in V(T_{k-1})$ we have

$$d_{T_{k-1}}(v, X_p) = d_G(v, X_p).$$

Let v' be such a vertex of $V(G) - V(T_{k-1})$ that for all

$$v \in V(G) - V(T_{k-1}) \text{ it holds } d_G(v, X_p) \leq d_G(v, X_p).$$

Let $(x_i = v_1, v_2, \dots, v', v')$ be a path of the length

$$d_G(v', X_p)$$

from the set X_p to V in G . Then we define $V(T_k) = V(T_{k-1}) \cup \{v\}$

$$E(T_k) = E(T_{k-1}) \cup \{(v, v')\}$$

As we have added only an end vertex with it's edge by this operation, the graph T_k remains a forest. From the choice of vertex v it follows that $v \in V(T_{k-1})$ and from the inductual assumption it follows that

$$d_G(v, X_p) = d_{T_{k-1}}(v, X_p) \text{ from } E(T_k) \subset E(G) \text{ follows}$$

$$d_{T_k}(v, X_p) \geq d_G(v, X_p)$$

Further we have

$$d_{T_k}(v', X_p) \leq d_{T_k}(v', v'') + d_{T_k}(v'', X_p) = d_G(v', v'') + d_G(v'', X_p) = d_G(v', X_p)$$

This implies that for all $v \in V(T_k)$ it holds

$$d_{T_k}(v, X_p) \geq d_G(v, X_p) \text{ end of step } k.$$

Now we put $m = |V(G) - V(T_0)|$. After m steps we obtain the forest T_m such that $V(T_m) = V(G)$ and $E(T_m) \subset E(G)$ and for all $v \in V$ it holds

$$d_{T_m}(v, X_p) = d_G(v, X_p)$$

Now in the end we add to the forest T_m arbitrary edges from $E(G)$ to obtain connected graph without cycle. In this way we obtain the spanning tree T of G which we wanted to find.

Theorem 2.1

Let $G = (V, E)$ be a connected graph with given weight w of vertices and length l of edges. Let there exist a dominating set of p points of radius r on G . Then there exists a spanning tree T of G with a dominating set

$$X_p = (x_1, x_2, \dots, x_p) \text{ on } T \text{ of the same radius } r.$$

Proof

According to Lemma 2.1 there exists a dominating set

$X_p = (x_1, x_2, \dots, x_p)$ on G such that $d_G(v, X_p) \leq r$ for every $v \in V$ and that for every internal point $x_i \in X_p$ of an edge (v_{i-1}, v_{i+1}) it holds $d(v_{i-1}, (X_p - \{x_i\})) > d(v_{i-1}, x_i)$ and also

$$d(v_{i+1}, (X_p - \{x_i\})) > d(v_{i+1}, x_i).$$

Then according to Lemma 2.3 there exists a spanning tree T of G such that all the points of X_p lie on T , and that

$$d_T(v, X_p) = d_G(v, X_p) \leq r \text{ holds for all } v \in V.$$

From this directly follows that X_p is a dominating set of radius r on T .

After proving this important theorem we are able to formulate a general algorithm for finding the number p and the dominating set of radius r .

Algorithm 2.1

A dominating set of radius r in a vertex-weighted graph G

Step 1: Find all the spanning trees of G .

Step 2: For every spanning tree T of G use the Hakimi's algorithm [10] determining the dominating set on T .

Step 3: Search all the spanning trees with their dominating set and take that spanning tree which has the smallest number p of points of the dominating set.

From the theorem 2.1 it follows that if we want to find the dominating set on some graph it is sufficient to find the dominating set on all his spanning trees and then to take the best one. This implies that the algorithm 2.1 is correct.

Step 1 in algorithm 2.1 is described only in general but for a concrete class of graphs it is necessary to replace it by a concrete algorithm for generation all the spanning trees of a graph. E.g. such an algorithm in the case of unicyclic graph consists of sequential deleting of one edge from the cycle. Let the complexity of Step 1 be Z , then the complexity of the whole algorithm is $O(Z + K.n)$, where K is the number of spanning trees of G . From this formula it follows that Algorithm 2.1 will work best if the graph has not too many spanning trees.

Corolly 2.1

The complexity of algorithm 2.1 for determining the dominating set of radius r on a unicyclic graphs is $O(m.n) \leq O(n^2)$ where m is the length of the cycle of the graph.

Corolly 2.2

The complexity of algorithm 2.1 for determining the dominating set of radius r in a cactus graph (Where a cactus is a connected graph, every cycle block of which is a cycle [1]), with k cycles of lengths m_1, m_2, \dots, m_k is $O(n.m_1.m_2 \dots m_k) \leq O(n.m^k) \leq O(n^{k+1})$ where $m = \max_{1 \leq i \leq k} \{m_i\}$

3. A p-center ($p > 1$) of vertex-weighted graphs with small number of spanning trees

In this section we describe two general algorithms for finding an (absolute or vertex) p -center of vertex-weighted graphs which are for the classes of graphs with small number of spanning trees more efficient that the Kariv-Hakimi's [10] general algorithm.

The first of these algorithm is based on the previous algorithm 2.1 for finding the (absolute or vertex) domination number of radius r and a corresponding dominating set.

Kariv and Hakimi showed in [10] that there exist $O(|E|n^2)$ ($O(n^2)$, resp.) possible values of the absolute (vertex, resp.) p -radius for the absolute (vertex, resp.) p -center. Denote the set of all the possible values of p -radius by Q . We can find Q in the time $O(|E|n^2)$ ($O(n^2)$, resp.) [10].

Let the desired (absolute or vertex) p -radius be denoted by r_p . Given $r' \in Q$, the (absolute or vertex) domination number p' of radius r' can be found by algorithm 2.1. if $p' \leq p$, then

$r_p \leq r'$ therefore, we obtain:

$$r_p = \min_{r' \in Q} \{r' : p' \leq p\}.$$

Thus, by using algorithm 2.1 one can search the $O(|E|n^2)$ or $O(n^2)$ possible values r' and find the p -radius of the given graph.

Algorithm 2.1 which gives the domination number p' , also

constructs a dominating set of radius r' . Thus, once the p -radius r_p is known, one can construct a dominating set of radius r_p . Let p_1 be the number of points in this set. If we add to the dominating set $p - p_1$ points (arbitrary $p - p_1$ points in the case of absolute p -center, or $p - p_1$ arbitrary vertices in the case of vertex p -center), then we obtain a desired (absolute or vertex) p -center.

In this way the following algorithm for finding the (absolute or vertex) p -center is verified.

Algorithm 3.1 A p -center ($p > 1$) of a vertex-weighted graph.

Step 1: Calculate the $O(|E|n^2)$ ($O(n^2)$, resp.) values r' for the absolute (vertex, resp.) p -center (by Kariv and Hakimi [10]).

Step 2: Arrange the $O(|E|n^2)$ ($O(n^2)$, resp.) values r' in a list L according to a nondecreasing order.

Step 3: By performing a binary search on the list L of the values r' , and by using Algorithm 2.1, find the smallest r' , for which $p' \leq p$ (where p' is the domination number of radius r'). denote this value of r' by r_p , and denote the domination number of radius r_p by p_1 .

Step 4: Let $X_{p_1}^*$ be the dominating set of radius r_p as constructed by Algorithm 2.1. Add any arbitrary $p - p_1$ points to $X_{p_1}^*$ (in the case of the vertex p -center, these points must

be vertices). The resulting set $X_{p_1}^*$ is a p -center of the given graph.

Step 1 of Algorithm 3.1 is carried out in $O(|E|n^2)$ ($O(n^2)$, resp.) operations. Step 2 requires $O(|E|n^2 \lg n)$ ($O(n^2 \lg n)$, resp.)

operations. The complexity of the binary search which is performed in Step 3 is $O(\lg n)$ operations, while Algorithm 2.1 which is carried out in each search case costs $O(Z + m.K)$ operations, where K is the number of all spanning trees of G and Z is the complexity of finding all spanning trees of G . Thus the overall complexity of Algorithm 3.1 in the general case is $O(|E|n^2 \lg n + Z + K.n. \lg n)$ for the absolute p -center or $O(n^2 \lg n + Z + K.n. \lg n)$ for the vertex p -center. These general formulas imply the following corollaries in some special cases.

Corolly 3.1

The complexity of algorithm 3.1 in the case of unicyclic graphs is $O(n^3 \lg n)$ ($O(n^2 \lg n)$, resp.) for the absolute (vertex, resp.) p -center.

Corolly 3.2

The complexity of algorithm 3.1 in the case of cactus graph with k cycles of lengths $m_1 \leq m_2 \leq \dots \leq m_k = m$ is

$$O(n^3 \lg n + m_1 . m_2 . \dots . m_k . n \lg n) \leq O(n^3 \lg n + m^k . n \lg n) \leq O(n^{k+1} \lg n) \text{ for absolute } p\text{-center and}$$

$$O(n^2 \lg n + m_1 . m_2 . \dots . m_k . n \lg n) \leq O(n^2 \lg n + m^k . n \lg n) \leq O(n^{k+1} \lg n) \text{ for vertex } p\text{-center.}$$

For the next part of this paper it will be useful to record the following observation.

Lemma 3.1

Let $G = (V, E)$ be a connected graph with vertex weight w and the edge length l and let $G_1 = (V_1, E_1)$ be a connected subgraph of G , such that $V = V_1, E_1 \subset E$. Then the value of the (absolute or vertex) p -center or the minimum distance sum of the p -median of G_1 is equal or greater than the value of the same parameter of G .

Proof

The proof of this lemma follows immediately from the fact that

$$d_{G_1}(x, v) \geq d_G(x, v) \text{ for arbitrary vertex } v \in V \text{ and arbitrary point } x \text{ on } G_1.$$

The second algorithm which we will describe in this section is based on the following theorem.

Theorem 3.1

Let $G = (V, E)$ be a connected graph with the vertex weight w and with the edge length l . Let the (absolute or vertex) p -radius of G be r_p . Then there exist a spanning tree T of G with the same p -radius r_p .

Proof

The proof of this theory follows directly from theorem 2.1. Let Y_p be an absolute (vertex, resp.) p-center on G with the p-radius r_p , then Y_p is the dominating set on G of the radius r_p . From theorem 2.1 it follows that there exists spanning tree T of G with a dominating set X_p of radius r_p .

From the proof of theorem 2.1 it follows that if $Y_p \subset V$, then $X_p \equiv Y_p$. From this fact and from lemma 3.1 directly follows that X_p is the absolute (vertex, resp.) p-center of T with the radius r_p .

From theorem 3.1 it follows that if we want to find the (absolute or vertex) p-center on G, it is sufficient to search all the spanning trees of G, determine the p-radius for each one and then to choose the spanning tree which has the lowest p-radius. This p-radius will be the p-radius of G and the corresponding p-center of the best spanning tree will be the p-center of G, as well. These considerations allow us to formulate the next general algorithm.

Algorithm 3.2 A p-center (p>1) of vertex weighted graph G

- Step 1: Determine all the spanning trees of the graph.
- Step 2: For each spanning tree use the Kariv and Hakim's algorithm [10] for determining the p-center and p-radius on a tree.
- Step 3: Compare the values of the p-radius of all spanning trees and choose the spanning tree with lowest p-radius. This p-radius will be the p-radius of G and the corresponding p-center will be the p-center of G, too.

Let us assume that K is the number of all the spanning trees and let Z be the complexity of the algorithm finding them. As the complexity of the Kariv-Hakimi's algorithm for finding the p-center on the tree is $O(n^2 \lg n)$, the complexity of Algorithm 3.2 is $O(Z + Kn^2 \lg n)$ both for the absolute and vertex p-center. This general formula gives us special cases following corollaries.

Corollary 3.3

The complexity of algorithm 3.2 in the case of unicyclic graph with the cycle of length m is $O(m \cdot n^2 \lg n)$.

Corollary 3.4

The complexity of algorithm 3.2 in the case of cactus with k cycles of lengths $m_1 \leq m_2 \leq \dots \leq m_k = m$ is $O(m_1 \cdot m_2 \cdot \dots \cdot m_k \cdot n^2 \lg n) \leq O(m^k \cdot n^2 \lg n) \leq O(n^{k+2} \lg n)$.

From corollaries 3.1 - 3.4 we can see that it is better to use Algorithm 3.2 for seeking an absolute p-center of an unicyclic graph (mainly if the cycle is short) and that the Algorithm 3.1 is better for seeking the vertex p-center of all cactus graphs and for seeking the absolute p-center of cactus graphs with two and more cycles.

4. A p-median (P > 1) of a vertex weighted graphs with small number of spanning trees

Kariv and Hakimi in [13] described an $O(n^2 \cdot p^2)$ algorithm for finding a p-median (p > 1) of a tree T as follows:

Begin by converting the given tree into a rooted tree as follows:

We pick an arbitrary vertex $v_0 \in V$ to be the "root" of tree T. Let v be a vertex of the tree. We define the level $Lev(v)$ of v to be the number of edges on the path $p(v_0, v)$ which leads from the root v_0 to v (in particular $Lev(v_0) = 0$).

We denote $L_m = \max_{v \in V} \{Lev(v)\}$. If $v \neq v_0$, then by removing the last edge of the path $p(v_0, v)$ we obtain two connected subtrees. We denote the subtree which contains v by T_v , and we define v as the root of T_v (in particular, if v is a leaf of T, then T_v is the single vertex v). We also define

T_{v_0} to be the original tree T. It is not difficult to see that if $v' \in T_v$ and $v' \neq v$ then $Lev(v') > Lev(v)$. The number of vertices of T_v is denoted by $|T_v|$ (in particular $|T_{v_0}| = n$).

Let $v \in V$ and let $E(v)$ be the set of all edges of T_v which are adjacent to v (in particular, if v is a leaf of T then $E(v) = \emptyset$).

We define an arbitrary order among the edges of $E(v)$, and we denote the l-th edge (according to this order) by $e(v, l)$. If vertex v_s is the other endpoint of the edge $e(v, l)$ (namely, $e(v, l) = (v, v_s)$), then we say that v_s is the l-th son of v, and v is the father $F(v_s)$ of v_s .

As a pre-procedure for the algorithm we compute the distance matrix of the tree (this requires $O(n^2)$ steps).

The algorithm itself is of dynamic-programming type, and it consists of two phases: During the first phase we traverse the edges of the tree "upward", from the vertices of higher levels towards the vertices of lower levels, and we compute certain values to be associated with the edges and the vertices of the tree; these values are in fact the corresponding distance-sum of k-medians ($1 \leq k \leq p$) as calculated over the different subtrees T_v and over other subtrees of the original tree T. In particular, we find the distance sum $H(v_0, p)$ corresponding to a p-median of the whole tree. We use these values throughout the second phase in order to traverse the tree "downward", from lower levels to higher levels, and to locate the points of a p-median at p selected vertices of T.

Our aim in this section is to describe an algorithm finding a p-median in a general network which is efficient mainly for graphs with small number of spanning trees. This algorithm is based on the next theorem.

Theorem 4.1

Let $G = (V, E)$ be a connected graph with the vertex weight w and the edge length l. Let $X_p = (x_1, x_2, \dots, x_p)$, $X_p \subset V$ be a vertex p-median of G with the distance sum $H(X_p)$. Then there exists spanning tree T of G such that X_p is a p-median of T with the same distance sum, i.e

$$H_T(X_p) = H_G(X_p)$$

Proof

As the set of vertices $X_p \subset V$ satisfies the conditions of Lemma 2.3 it follows that there exists spanning tree T of G such that

$d_G(v, X_p) = d_T(v, X_p)$ for all $v \in V$. From this immediately follows that

$$H_T(X_p) = \sum_{v \in V} w(v) \cdot d_T(v, X_p) = \sum_{v \in V} w(v) \cdot d_G(v, X_p) = H_G(X_p)$$

From this and from Lemma 3.1 we have that X_p is a p -median of T with the same distance sum as the graph G .

After proving theorem 4.1 we know that if we want to find a p -median of general graph G , it is sufficient to search all the spanning trees of G , determine the p -median and the minimum distance sum for each spanning tree and then to choose that spanning tree which has the lowest minimum distance sum. This distance sum will be the minimum distance sum for the graph G and the corresponding p -median of the best spanning tree will be the p -median of G , too. After this consideration we are able to formulate the next general algorithm.

Algorithm 4.1 A p -median ($P > 1$) of a vertex-weighted graph

Step 1: Determine all the spanning trees of the graph.

Step 2: For each spanning tree use Kariv-Hakimi's algorithm [13] for determining the p -median and the minimum distance sum on a tree.

Step 3: Compare the values of minimum distance sums of all spanning trees and choose the spanning tree with the lowest minimum distance sum. This values will be the minimum distance sum of G and the corresponding p -median will be the p -median of G , too.

Let us assume that K is the number of all the spanning trees and let Z be the complexity of algorithm for finding them. As the complexity of Kariv-Hakimi's algorithm used in step 2 is $O(n^2 \cdot p^2)$ [13].

The complexity of algorithm 4.1 is $O(Z + k \cdot n^2 \cdot p^2)$. This general formula gives us the following corollaries in some special cases. **Corollary 4.1**

The complexity of algorithm 4.1 in the case of unicyclic graph with the cycle of length m is $O(m \cdot n^2 \cdot p^2) \leq O(n^3 \cdot p^2)$.

Corollary 4.2

The complexity of algorithm 4.1 in the case of cactus graph with the k cycles of length $m_1 \leq m_2 \leq \dots \leq m_k = m$ is $O(m_1 \cdot m_2 \dots \cdot m_k \cdot n^2 \cdot p^2) \leq O(m^k \cdot n^2 \cdot p^2) \leq O(n^{k+2} \cdot p^2)$.

Received 20 October, 1996

REFERENCES

- [1] **Behzad, M., Chartrand, G., Lesniak-Foster, L.:** Graphs and Digraphs, Prindle, Boston 1979.
- [2] **Hakimi, S. L.:** Optimum locations of switching centers and absolute centers and medians of a graph, Oper. Res. 12 (1964), 450-459.
- [3] **Hakimi, S. L.:** Optimum distribution of switching centers in a communication network and some related graph-theoretic problems, Oper. Res. 13 (1965), 462-475.
- [4] **Dearing, P. M. and Francis, R. L.:** A minmax location problem on a network, Trans. Sci. 8 (1974), 333-343.
- [5] **Goldman, A. J.:** Minimax location of a facility on a network, Transp. Sci. 6 (1972), 407-418.
- [6] **Helfin, N.:** On finding the absolute and vertex centers of a tree with distances, Transp. Sci. 8 (1974), 75-77.
- [7] **Handler, G. Y.:** Minimax network location theory and algorithms, Technical Rep. No. 107, Oper. Res. Center, Mass, Inst. of Tech, Cambridge, Mass, Nov. 1974.
- [8] **Handler, G. Y.:** Minimax location of facility in an undirected tree graph, Transp. Sci. 7(1973), 287-293.
- [9] **Minieka, E.:** The m -center problem, SIAM Review 12 (1970), 138-139.
- [10] **Kariv, O., Hakimi, S. L.:** An algorithmic approach to network location problems, I: the p -center, SIAM J. Appl. Math. 37 (1979), 513-538.
- [11] **Hassan, M. H.:** The p -center problem in a unicyclic graph, Acta Math. Univ. Come.
- [12] **Goldman, A. J.:** Optimal center location in simple networks, Transp. Sci. 5 (1971), 212-221.
- [13] **Kariv, O., Hakimi, S. L.:** An algorithmic approach to network location problems, II: the p -median, SIAM J. APPL. Math. 37 (1979) 539-560.
- [14] **Hassan, M. H.:** The p -center and the p -median problems in a cactus graph, Acta Math. Univ. Come.