

The 2-Facility Centdian Network Problem ^{*}

Dionisio Pérez-Brito, José A. Moreno-Pérez,
and Inmaculada Rodríguez-Martín

Dep. Estadística, I.O. y Computación.
Universidad de La Laguna.

^{*}The research has been partially supported by the DGICYT project PB95-1237-C03-02.

The 2-Facility Centdian Network Problem.

abstract

The p -facility centdian network problem consists in finding the p points that minimize a convex combination of the p -center and p -median objective functions. The vertices and local centers constitute a dominating set for the 1-facility centdian; i.e., it contains an optimal solution for all instances of the problem. Hooker, Garfinkel and Chen (1991) give a theoretical result to extend the dominating sets for the 1-facility problems to the corresponding p -facility problems. They claim that the set of vertices and local centers is also a dominating set for the p -facility centdian problem. We give a counterexample and an alternative finite dominating set for $p = 2$. We propose a solution procedure for a network that improves the complexity of the exhaustive search in the dominating set. We also provide a very efficient algorithm that solves the 2-centdian on a tree network with complexity $O(n^2)$.

Keywords: Location, Centdian, Networks, Tree.

1. Introduction

Facility location deals with the problems of locating one or several facilities in order to optimize some criteria with regard to the users. In this paper we consider the problem of selecting several points of a network in order to minimize a function which is distance dependent with respect to given points of the network. The median and the center problems are two well known problems with numerous possible applications. The first is suitable for locating a facility providing a routine service, by minimizing the average distances from the users to it. The second is appropriate for emergency services where the objective is to have the farthest users as near as possible to the facility center.

In many real world problems the objective is a mixture of different, possibly adverse objectives. For example, locating a fire station may call for the minimization of the travelling time to the farthest potential source of call for service and also for being as close as possible to the heavily populated areas. The problem is therefore to minimize both objective functions. Such a goal may be mathematically expressed by minimizing a new objective function that is a convex combination of the objective functions of the center and median problems. This multi-objective approach for locating a facility on a network was introduced by Halpern (1976), who coined the term **centdian** for the point which minimizes the convex combination of the center and median objective functions.

Ever since the seminal paper by Hakimi (1964), a thread running through network location theory is the identification of a finite subset of the network that necessarily contains an optimal solution for all the instances of a particular location problem. Since Hakimi (1964), it is known that the set of vertices is a finite dominating set for the p -median problem. The set of vertices and local centers (points, in the interior of the edges, that are equidistant and balanced with respect to two vertices) is a finite dominating set for the p -center problem; e.g. see Moreno (1985). From Halpern (1978), it is known that the set of vertices and local centers of the network is a finite dominating set for the single facility centdian problem. Hooker, Garfinkel and Chen (1991) consider a theoretical result which extends the finite dominating sets of the single facility problems to the

corresponding p -facility problems, and apply it to the p -facility centdian problem (corollary 9). We give a counterexample showing a p -facility centdian instance where there is not optimal solution consisting of vertices and local centers. We propose an alternative finite dominating set for the 2-facility centdian problem and a solution procedure for a network that improves the complexity of the exhaustive search in the dominating set. We also provide a very efficient algorithm that solves the 2-centdian on a tree network with complexity $O(n^2)$.

The next section provides the basic definitions and notation for the formulation of the p -facility centdian problem on networks, that is derived from the classical p -center and p -median problems. Section 3 includes a very simple counterexample for $p = 2$ in a tree network and section 4 the new finite dominating set of points is provided. Section 5 includes the two algorithms that solve the problem in a general network and in a tree network. In the last section some conclusions are provided. The appendix includes the technical rigorous details of the proof of the main result of the paper.

2. Formulation of the problem.

Let $N = (V, E)$ be a connected network where V is a set of n points called vertices and E is a set of m undirected edges. Each edge is a continuous and linear set of points joining two vertices; the edge $e \in E$ joining vertices i and j is denoted by $e = [i, j]$. The points i and j are the only vertices of every edge $e = [i, j] \in E$. For every $e \in E$ let $l(e) = l(i, j)$ be the positive length of $e = [i, j]$ which is a measure of the edge. In location problems the length of an edge represents the cost or time of going once through the edge to satisfy the demand of one client or user.

For every point x of an edge $[i, j]$, let $[i, x]$ and $[x, j]$ be the edges consisting of the portions of $[i, j]$ between x and i and between x and j , respectively. Any point x of an edge $e = [i, j]$ is determined by the value t of the length of $[i, x]$, $0 \leq t \leq l(e)$; this point x is then denoted by $x = p(e, t) = p([i, j], t)$. The vertices of an edge $e = [i, j]$ are the

points $i = p([i, j], 0)$ and $j = p([i, j], l(i, j))$. The points $x = p(e, t)$, for $0 < t < l(e)$, are interior to the edge. Let $P(N)$ be the set of points of the network. This set consists of, on one hand, the union of the sets of points of each edge and, on the other hand, the disjoint union of the set of vertices and the set of interior points.

A path between two points x and y is a sequence of edges $[x_i, x_{i+1}]$, $i = 0, 1, \dots, k$, such that $x_0 = x$ and $x_{k+1} = y$ with $x_i \in V$ for every $i = 1, \dots, k$. The length of a path is equal to the sum of the lengths of all its edges. The distance $d(x, y)$ between any two points x and y is equal to the length of the shortest path between them.

The distance between two points represents the cost or time of the shortest way of going from one point to the other to supply one client. The set $P(N)$ is a mathematical topological space with the topology induced in N by the metric given by the distance $d(., .)$. This allows us to consider continuity, differentiability and slopes of functions defined for points of $P(N)$ or for vectors of points in $P(N)$.

For every number $p > 0$, let $X = (x_1, x_2, \dots, x_p)$ be a collection of p points of $P(N)$. The distance from X to a point $u \in P(N)$ is given by:

$$d(X, u) = \min\{d(x_i, u) : i = 1, 2, \dots, p\}.$$

Let $U \subset V$ represent the set of vertices where the user of the service is established.

The median problems consist of determining the locations for the facilities that minimize the total or average travel time or distance between the facilities and their users. For a given value of p , the so called p -median problem is to establish p facilities in p of the potential locations and to supply each client with one of the established facilities in order to minimize the sum of the distances to the users. Given the set of user vertices $U \subseteq V$, the p -median problem consists of finding p facility points $X = (x_1, \dots, x_p)$ on N minimizing the function:

$$f_m(U; X) = \sum_{u \in U} d(X, u).$$

The p -center problem consists in finding p facility points $X = (x_1, \dots, x_p)$ on N minimizing the function:

$$f_c(U; X) = \max_{u \in U} d(X, u).$$

This means to open p facilities and to assign each user to just one of them, minimizing the maximum distance from any open facility to any of the users assigned to it.

The centdian locations are those that minimize a convex combination of the objective functions of the center and median problem. Given the number p of facilities to be located and a value λ , $0 \leq \lambda \leq 1$, the p -facility λ -centdian problem (or the p - λ -centdian problem) is to find the p facility points $X = (x_1, \dots, x_p)$ on N that minimize the objective function:

$$f_\lambda(U; X) = \lambda \cdot f_c(U; X) + (1 - \lambda) \cdot f_m(U; X).$$

The value of λ reflects the weight attributed to the maximum distance with respect to the total distance. It is usually difficult for the decision maker to specify exactly the weights that should be assigned to the two objectives. When $\lambda = 0$, the λ -centdian is a median; when $\lambda = 1$, it is a center. For $0 < \lambda < 1$, the λ -centdian can be regarded as a solution to a location problem where both efficiency and equity criteria are taken into account. The 2 - λ -centdian goal is to find two points $x_1^*, x_2^* \in P(N)$ such that

$$f_\lambda(U; (x_1^*, x_2^*)) = \min_{x_1, x_2 \in P(N)} f_\lambda(U; (x_1, x_2)).$$

Two sets of interior points of $P(N)$ will be used to define a finite dominating set for the centdian problem on the network.

- A point $x \in P(N)$ is a **local center** with **range** r associated to vertices $u, v \in V$ (we denote $x \in LC(r; u, v)$) if x is an interior point of an edge $[i, j]$ such that:
 - 1) $r = d(x, u) = l(x, i) + d(i, u) < l(x, j) + d(j, u)$ and
 - 2) $r = d(x, v) = l(x, j) + d(j, v) < l(x, i) + d(i, v)$.
- A point $x \in P(N)$ is an **extreme point** with **range** r associated to vertex $v \in V$ (we denote $x \in EP(r; v)$) if x is an interior point of an edge $[i, j]$ such that:

- 1) $r = d(x, v) = l(x, i) + d(i, v)$ or
- 2) $r = d(x, v) = l(x, j) + d(j, v)$.

3. The counterexample.

A dominating set for a location problem is a set of points that contains an optimal solution for all instances of the problem. The set of vertices is a finite dominating set for the p -median problem (Hakimi, 1964). The set of vertices and local centers is a finite dominating set for the p -center problem (Moreno, 1985). The set of vertices and local centers is also a finite dominating set for the single ($p = 1$) λ -centdian problem (Halpern, 1978). However, it is not a dominating set for the p - λ -centdian problem as was claimed by Hooker, Garfinkel and Chen (1991). They use a theoretical result (lemma 10) which extends the finite dominating sets of the single facility to the p -facility problems, and apply it to the p -facility centdian problem. We provide a counterexample that shows that, even in the unweighted case for $p = 2$ in a tree, the set of vertices and local centers is not a dominating set.

Consider 2- λ -cent-dian problem for $\lambda = 0,8$ in the tree network with six user vertices given in figure 1; the lengths are shown below each edge. Two clusters of vertices are easily found; the first one, with vertices v_1, v_2, v_3 and v_4 , and the second one, with vertices v_5 and v_6 . For the second cluster, the best location is $x_2 = p([v_5, v_6], 5)$ that is the only local center associated to v_5 and v_6 ; $\{x_2\} = LC(5; v_5, v_6)$. For the first cluster, the local centers associated to vertices v_1, v_2, v_3 and v_4 are: $p([v_1, v_2], 3) \in LC(3; v_1, v_2)$, $p([v_1, v_2], 4) \in LC(4; v_1, v_3) = LC(4; v_1, v_4)$, $p([v_2, v_3], 1) \in LC(1; v_2, v_3)$ and $p([v_2, v_4], 1) \in LC(1; v_2, v_4)$. The best of these candidates is $x_1 = p([v_1, v_2], 4)$. Other local centers are in the longest edge and clearly they can not be in an optimal solution. The objective value for $X = \{x_1, x_2\}$ is $f_\lambda(X) = 8,8$;

$$f_\lambda(X) = 0,8 \cdot d(x_2, v_6) + 0,2 \cdot \left[\sum_{i=1}^4 d(x_1, v_i) + \sum_{i=5}^6 d(x_2, v_i) \right] = 0,8 \cdot 5 + 0,2 \cdot (14 + 10) = 8,8$$

However the only optimal solution is $X^* = \{x_1^*, x_2^*\}$, where $x_1^* = p([v_1, v_2], 5)$ and $x_2^* = x_2$,

with objective value $f_\lambda(X^*) = 8,4$;

$$f_\lambda(X^*) = 0,8 \cdot d(x_1^*, v_1) + 0,2 \cdot \left[\sum_{i=1}^4 d(x_1^*, v_i) + \sum_{i=5}^6 d(x_2^*, v_i) \right] = 0,8 \cdot 5 + 0,2 \cdot (12 + 10) = 8,4$$

Note that x_1^* is not a local center.

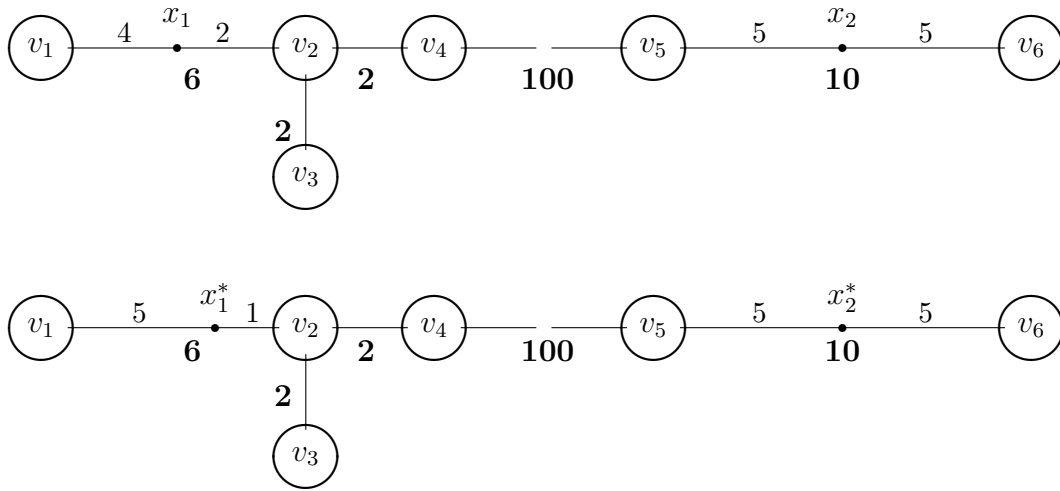


Figura 1: **The counterexample.** For $\lambda = 0,8$, the best solution using only vertices and local centers is (x_1, x_2) . The only optimal solution for $\lambda = 0,8$ consists in (x_1^*, x_2^*) . Point x_2^* is a local center with range 5 and x_1^* is an extreme point with range 5 and it is not a local center.

4. The new finite dominating set.

We propose a finite dominating set for the 2-centdian problem on networks that consists of the vertices, the local centers and the extreme points whose range is equal to the range of a local center or to the distance between two vertices.

For every value $r > 0$, let

$$LC(r) = \bigcup_{u,v \in U} LC(r; u, v) \quad \text{and} \quad EP(r) = \bigcup_{u \in U} EP(r; u).$$

There is at most one local center with respect to each couple of vertices on every edge. Hence there is a finite number of ranges of local centers. Let the distances from vertices to user vertices and the ranges of local centers associated to user vertices be:

$$R = \{r : LC(r) \neq \emptyset\} \cup \{r : r = d(v, u), v \in V, u \in U\}.$$

Let this set be named the canonical set of distances. The extreme points with range in this set are the canonical extreme points. Then, the set of local centers and the set of canonical extreme points associated to user vertices are:

$$LC = \bigcup_{r \in R} LC(r) \quad \text{and} \quad EP = \bigcup_{r \in R} EP(r).$$

The proposed finite dominating set of points is: $D = V \cup LC \cup EP$.

Theorem 1. *The set $D = V \cup LC \cup EP$ consisting of the vertices, the local centers and the canonical extreme points associated to user vertices is a finite dominating set for the 2- λ -centdian problem in the network.*

Proof. Here we outline intuitively the main steps of the proof; rigorous technical details are given in the appendix.

The main idea is to demonstrate that the facility points of an optimal solution can be moved on their edges until a point of D is found without increasing the objective function. Given the optimal solution (x_1, x_2) consisting of the two points selected to open the facilities at them, the user vertices are obviously optimally assigned to the closest one. The set of user vertices is split into three clusters U_0 , U_1 and U_2 . Set U_0 consists of those vertices that can be assigned to x_1 or to x_2 , U_1 consists of those vertices that can be assigned only to x_1 and U_2 consists of those vertices that can be assigned only to x_2 .

If the vertices of U_0 are assigned to x_2 then the 2-center and the 2-median objective functions are given as follows:

$$\begin{aligned} f_c(U; (x_1, x_2)) &= \max\{f_c(U_1; x_1), f_c(U_2 \cup U_0; x_2)\} = \max\{r_1, r_2\} = r \\ f_m(U; (x_1, x_2)) &= f_m(U_1; x_1) + f_m(U_2 \cup U_0; x_2) = s_1 + s_2 \end{aligned}$$

Suppose that $x_2 \notin D$. We consider three cases: $r = r_1 > r_2$, $r_1 < r_2 = r$ and $r_1 = r = r_2$.

Since x_2 is not a vertex, it is an interior point of an edge. In the first two cases, this point x_2 can be moved on its edge without decreasing the objective function $f_\lambda(U; (x_1, x_2))$. In order to see it, rewrite the slope of this objective function as a function of the distance from x_2 to one of the vertices of the edge, when it is moved on its edge. The sum of the two slopes of the movements of x_2 in the two possible directions on its edge is not positive; so one of them is not positive. This one gives a direction to move x_2 while the objective function does not increase.

In the third case, $r_1 = r_2 = r$, the two points x_1 and x_2 are interior points. These points can be shifted the same distance, at the same time, in the direction in which both r_1 and r_2 increase or decrease. Also the sum of the slopes of the objective function corresponding to these two movements is not positive; so one of them gives the direction to move them.

These movements can be done until x_2 (or x_1 , in the third case) reaches the end of its edge (that is a vertex) or a local center associated to two of the farthest vertices. Then the value of r is the distance from x_2 (or x_1) to an user vertex or the range of the local center, and the other facility point is in $E(r)$ with $r \in R$. \diamond

5. The algorithms.

The first direct consequence of getting a finite dominating set for an optimization problem is that it can be solved exactly by an exhaustive search. However, the complexity of the search depends on the size of the set. Let $|V| = n$ and $|E| = m$.

Proposition 1. *The size of the finite dominating set for the 2- λ -centdian problem on a general network is: $|D| = O(m^2n^3)$.*

Proof. The values for R are ranges of local centers and distances between vertices. For every edge, there is at most a local center with respect to each couple of vertices, then $|LC| = O(mn^2)$. Therefore the size of R is $O(mn^2)$. For every $r \in R$, there are at most two extreme points with range r in each edge with respect to every vertex, thus $|E(r)| = O(mn)$. The greater part of D is the union of $O(mn^2)$ sets with size $O(mn)$. Consequently $|D| = O(m^2n^3)$. \diamond

The exhaustive search on pairs of points in D looking for the optimal solution of the 2- λ -centdian takes $O(m^4n^7)$ time because we need $O(n)$ time to compute the objective function for each pair. However the complexity of the search can be reduced because not every pair of points in D constitute a candidate for the 2- λ -centdian. We infer from the proof of the theorem, that the facility point of the optimal solution where the maximum distance is reached has to be a vertex or a local center. This point provides the maximum distance r^* to an user vertice and the other facility point is an extreme point with range r^* or a vertex.

Proposition 2. *Let (x_1^*, x_2^*) be an optimal solution of the 2- λ -centdian on N and $r^* = f_c(U; (x_1^*, x_2^*))$. Then:*

- 1) $x_1^* \in LC(r^*) \cup (V \cap E(r^*))$ and $x_2^* \in V \cup E(r^*)$, or
- 2) $x_2^* \in LC(r^*) \cup (V \cap E(r^*))$ and $x_1^* \in V \cup E(r^*)$.

The following algorithm for solving the 2- λ -centdian problem is based on these results.

Algorithm 2CDN.

Step 0. Set $f^* = \infty$.

Step 1. Compute the set P_1 of pairs (x, r) where $x \in LC(r)$. Compute the set P_2 of pairs (v, r) where $v \in V$ and $r = d(v, u)$ for some $u \in U$. Let L_1 be the list of pairs in $P_1 \cup P_2$.

Step 2. Take a pair (x_1, r_1) of L_1 and compute the set $E(r_1)$. Let L_2 be the list of points in $E(r_1) \cup V$.

Step 3. Take one point x_2 in L_2 and compute $f = f_\lambda(U; x_1, x_2)$. If $f < f^*$ do $f^* = f$, $x_1^* = x_1$ and $x_2^* = x_2$. Delete x_2 from L_2 .

Step 4. If L_2 is not empty go to step 3.

Step 5. Delete (x_1, r_1) from L_1 . If L_1 is not empty go to step 2, otherwise stop.

Proposition 3. *The algorithm 2CDN has complexity $O(m^2n^4)$.*

Proof. First we need the distances between vertices. This takes $O(mn \log n)$ time using, for instance, the well known Dijkstra algorithm. The sets P_1 and P_2 have sizes $O(mn^2)$ and $O(n^2)$ and can be computed in this time. For any r , the set $E(r)$ is computed in $O(mn)$ time by testing every user vertex and every edge. Therefore, the number of pairs of points evaluated in step 3 is $O(m^2n^3)$. Since the objective function is computed in $O(n)$ time, the overall complexity is $O(m^2n^4)$. \diamond

If the underlying graph is a tree network T then $m = n - 1$ and $|LC| = O(n^2)$. Therefore $|R| = O(n^2)$ and $E(r) = O(n^2)$, for every $r \geq 0$. Hence $|D| = O(n^4)$ and the exhaustive search on the pairs of points in D takes $O(n^9)$ time. The algorithm 2CDN applied to a tree network takes $O(n^5)$ time. We propose an algorithm that improves significantly this complexity.

Let (x_1^*, x_2^*) be an optimal solution of the 2- λ -centdian problem in $N = T$ and (T_1^*, T_2^*) denote the two spanning subtrees generated by the user points assigned to x_1^* and x_2^* ; i.e. by the sets of vertices U_1^* and $U_2^* \cup U_0^*$ given above. Assume that $r_1^* \geq r_2^*$, then

$$r^* = f_c(T; (x_1^*, x_2^*)) = \max\{f_c(T_1^*; x_1^*), f_c(T_2^*; x_2^*)\} = f_c(T_1^*; x_1^*) = r_1^*.$$

Therefore:

$$\begin{aligned} f_\lambda(T, (x_1^*, x_2^*)) &= \lambda \cdot f_c(T_1^*; x_1^*) + (1 - \lambda) \cdot (f_m(T_1^*; x_1^*) + f_m(T_2^*; x_2^*)) = \\ &= [\lambda \cdot f_c(T_1^*; x_1^*) + (1 - \lambda) \cdot f_m(T_1^*; x_1^*)] + (1 - \lambda) \cdot f_m(T_2^*; x_2^*) = \\ &= f_\lambda(T_1^*; x_1^*) + (1 - \lambda) \cdot f_m(T_2^*; x_2^*). \end{aligned}$$

Given the optimal partition of T in T_1^* and T_2^* , If $r_1^* > r_2^*$, then x_1^* is the λ -centdian of T_1^* . If x_1^* is not a λ -centdian of T_1^* then replacing it by the λ -centdian we would get a better pair of facility points. Let m_2 be the median of T_2^* , if $f_c(T_2^*; m_2) \leq r^*$ then $x_2^* = m_2$. Otherwise ($f_c(T_2^*; m_2) > r^*$), let u_2 be the farthest user vertex of T_2^* ; i.e. $u_2 \in T_2^*$ such that $d(m_2, u_2) = f_c(T_2^*; m_2)$. Then x_1^* is a local center or a vertex in the path between the λ -centdian and the median of T_1^* and x_2^* is the point in the path from m_2 to u_2 at distance r^* from u_2 . Notice that x_2^* has to be within a distance r^* from u_2 ; i.e. $d(x_2^*, u_2) \leq r^*$. By the convexity of the objective function f_m of the median on trees, the point that minimizes it is the nearest point to x_2' .

Then the following algorithm solves the 2- λ -centdian on a tree network T .

Algorithm 2CDT.

Step 0. Set $f^* = \infty$. Let L be the list of edges in T .

Step 1. Take one edge $[i, j]$ of L and delete it from the tree T . Let T_i and T_j be the resulting subtrees. For $T_1 = T_i$ and $T_2 = T_j$, and for $T_1 = T_j$ and $T_2 = T_i$ take steps 2 to 6.

Step 2. Get the λ -centdian x_1 of T_1 by the Halpern algorithm. Compute $r_1 = f_c(T_1, x_1)$.

Step 3. Get the median x_2' of T_2 by the Goldman algorithm. Compute $r_2' = f_c(T_2, x_2')$.

Step 4. If $r_2' \leq r_1$ then do $x_2 = x_2'$, $r_2 = r_2'$ and compute $s_2 = f_m(T_2, x_2)$. Do step 6 for these two points.

Step 5. Otherwise, take the vertex $u_2 \in T_2$ such that $d(x_2', u_2) = r_2'$. For every local center or vertex x_1 in the path between x_1 to the median of T_1 , compute $s_1 = f_m(T_1, x_1)$ and let x_2 be the point in the path between x_2' and u_2 at distance $r_1 = f_c(T_1, x_1)$ from u_2 and compute $s_2 = f_m(T_2, x_2)$. For every pair (x_1, x_2) do step 6.

Step 6. Compute $f = \lambda \cdot r_1 + (1 - \lambda) \cdot (s_1 + s_2)$. If $f < f^*$ then do $x_1^* = x_1$, $x_2^* = x_2$ and $f^* = f$.

Step 7. Delete the edge $[i, j]$ from L . If L is not empty return to step 1, otherwise stop.

Proposition 4. *The algorithm 2CDT has complexity $O(n^2)$.*

Proof. The number of edges to be deleted at step 1 is $n - 1$; then steps 2 to 6 are performed $O(n)$ times.

The λ -centdian of the tree T_1 is computed in $O(n)$ time using the Halpern algorithm (Halpern, 1976). The median of the tree T_2 is computed in $O(n)$ time using the Goldman algorithm (Goldman, 1971). The distance from a point to a set of vertices takes $O(n)$ time using a simple DFS tree search, see Horowitz & Sahni (1978). The user vertex u_2 in step 4 can also be obtained in $O(n)$ time using the information provided by the DFS. The number of local center in step 5 is $O(n)$. The values s_1 and s_2 are updated in $O(1)$ time using the information provided by Goldman algorithm when x_1 and x_2 are moved on the corresponding paths.

Therefore the overall complexity of the algorithm is $O(n^2)$. ◇

An heuristic improvement of the procedure consists in taking the edges to be tested at step 1 using a BFS (Breadth First search) strategy to traverse the tree T starting at the median as the root of the tree, see Horowitz & Sahni (1978). If, for an edge e taken at step 1 of algorithm 2CDT, the value $f_1 = \lambda r_1 + (1 - \lambda)s_1$ is greater than the current best value f^* , then all the edges of the branch from e can be avoided because they can not provide a better candidate pair of facility points.

This heuristic improvement of the algorithm, named *2CDTH*, was programmed in PASCAL language and run on a *Pentium - 100Mhz*. We tested it solving randomly generated instances with 1000 vertices in 4 seconds (it takes no more than 50 edges to be deleted at step 1).

6. Conclusions.

The existence of a finite dominating set for a location problem is a first step in the search for a procedure to solve it. The paper by Hooker, Garfinkel and Chen (1991) provides them for a large list of location problems on networks. It includes a result for extending the finite dominating sets for a single facility location problem to the corresponding p -facility one. In this paper we show that one must be cautious about using these results. Namely, they claim that, by using these results, the set of vertices and local centers is a finite dominating set for the p -facility centdian problem. The counterexample included in this paper proved that it is not true.

We found that a new kind of points (the r -extremes) have to be considered. This is because when the maximum of the distances from the facility points to the users is part of the objective function then the facility points where this maximum is not reached have to be moved until a new tie in the maximum of the distances is reached. This new kind of points have to be considered when determining finite dominating sets for the p -facility centdian problem for p greater than 2, and also for other multifacility location problems. These results will appear in forthcoming papers.

7. Appendix.

We include in this appendix the technical rigorous details of the proof of theorem 1 outlined in section 4. The theorem is:

Theorem 1. *The set $D = V \cup LC \cup EP$ consisting of the vertices, the local centers and the canonical extreme points associated to user vertices is a finite dominating set for the 2 - λ -centdian problem in the network.*

Proof. Let (x_1, x_2) be an optimal solution of the problem. We show that if $x_1 \notin D$ or $x_2 \notin D$ then there are $x'_1, x'_2 \in D$ such that $f_\lambda(U; (x'_1, x'_2)) \leq f_\lambda(U; (x_1, x_2))$ because they

can be moved on their edges until a point of D is found without increasing the objective function.

Given the facility points x_1 and x_2 , the user vertices are obviously optimally assigned to the closest one. The possible optimal assignments are important to study how the function f_λ changes when one or both of these points are moved on their edges. Every user vertex is in one of the following sets:

$$U_1 = \{u \in U : d(x_1, u) < d(x_2, u)\}, U_2 = \{u \in U : d(x_1, u) > d(x_2, u)\} \text{ and} \\ U_0 = \{u \in U : d(x_1, u) = d(x_2, u)\}.$$

The vertices of U_0 can be assigned to x_1 or to x_2 . If they are assigned to x_2 then the 2-center and the 2-median objective functions are given as follows:

$$f_c(U; (x_1, x_2)) = \max\{f_c(U_1; x_1), f_c(U_2 \cup U_0; x_2)\} = \max\{r_1, r_2\} = r. \\ f_m(U; (x_1, x_2)) = f_m(U_1; x_1) + f_m(U_2 \cup U_0; x_2) = s_1 + s_2 = s.$$

Suposse that $x_2 \notin D$. We consider three cases: case 1) when $r = r_1 > r_2$, case 2) when $r_1 < r_2 = r$ and case 3) when $r_1 = r = r_2$.

Case 1). Let $r_2 < r_1 = r$.

Since $x_2 \notin D$ then $x_2 \notin V$; therefore, it is an interior point of an edge $[i, j]$. We show how to move this point x_2 on its edge without decreasing the objective function $f_\lambda(U; (x_1, x_2))$. In order to do it, we analyze the slope of the objective function as a function of the distances from x_2 to the vertex i , when it is moved on its edge.

The set of user vertices such that the distance to $x \in [i, j]$ is reached by a path passing through the vertex j , is denoted by $U^j(x)$.

$$U^j(x) = \{v \in U : d(x, v) = l(x, j) + d(j, v) \leq l(x, i) + d(i, v)\}.$$

Let $x(\xi)$ denote the point x when it is moved on its edge a small amount ξ in the direction to j ; i.e. if $x = p([i, j], t)$ then $x(\xi) = p([i, j], t + \xi)$. Therefore, if $u \in U^j(x)$ then $d(x(\xi), u) = d(x, u) - \xi$ and if $u \notin U^j(x)$ then $d(x(\xi), u) = d(x, u) + \xi$.

Given the facility points x_1 and x_2 , the vertices of U_1 have to be assigned to x_1 , those of U_2 have to be assigned to x_2 and the vertices of U_0 can be assigned to x_1 or to x_2 . However, when $x_2 \in [i, j]$ is moved on its edge, the user vertices that have to be assigned to $x_2(\xi)$ are those of $U_2 \cup (U_0 \cap U^j(x_2))$, and the user vertices that have to be assigned to x_1 are those of $U_1 \cup (U_0 - U^j(x_2))$. For simplicity we rename these sets by $U'_1 = U_1 \cup (U_0 - U^j(x_2))$ and $U'_2 = U_2 \cup (U_0 \cap U^j(x_2))$.

The objective function $f_\lambda(U; (x_1, x_2(\xi)))$ is reformulated by computing $f_c(U; (x_1, x_2(\xi)))$ and $f_m(U; (x_1, x_2(\xi)))$ as follows:

$$f_c(U; (x_1, x_2(\xi))) = \max\{\max_{v \in U'_1} d(x_1, v), \max_{v \in U'_2} d(x_2(\xi), v)\} = \max\{r_1, r_2(\xi)\} = r_1.$$

$$f_m(U; (x_1, x_2(\xi))) = \sum_{v \in U'_1} d(x_1, v) + \sum_{v \in U'_2} d(x_2(\xi), v) = s_1 + s_2(\xi).$$

The slope of $f_m(U; (x_1, x_2(\xi)))$, as function of ξ , when the movement is towards j , is:

$$m_2^j(\xi) = |U_2 - U^j(x_2)| - |U_0 \cap U^j(x_2)| - |U_2 \cap U^j(x_2)|.$$

If the movement is towards i , then the slope of f_m is

$$m_2^i(\xi) = |U_2 - U^i(x_2)| - |U_0 \cap U^i(x_2)| - |U_2 \cap U^i(x_2)|.$$

The slope of the function $f_c(U; (x_1, x_2(\xi)))$ is zero because the radius r_1 does not change while $r_2(\xi) \leq r_1$. Thus the slope of f_λ is $s_2^j(\xi) = (1 - \lambda)m_2^j(\xi)$, if the movement is towards j , and $s_2^i(\xi) = (1 - \lambda)m_2^i(\xi)$, if it is towards i . The sum of these slopes is

$$s_2^i(\xi) + s_2^j(\xi) = -2(1 - \lambda) \cdot |(U_0 \cup U_2) \cap U^i(x_2) \cap U^j(x_2)| \leq 0.$$

Therefore, one of the slopes $s_2^i(\xi)$ or $s_2^j(\xi)$ is not positive.

If we assume that $s_2^j(\xi) \leq 0$ (the other case is similar), then $m_2^j(\xi) \leq 0$. In order to see that x_2 can be moved until a point of D is found we describe below in which conditions $m_2^j(\xi)$ can be increased when the point $x_2(\xi)$ goes towards to j .

Note that $m_2^j(\xi) = |U_2 - U^j(x_2)| - |U_0 \cap U^j(x_2)| - |U_2 \cap U^j(x_2)|$ can be increased only when $U_2 - U^j(x_2(\xi))$ gets an user vertex or when $U_0 \cap U^j(x_2(\xi))$ or $U_2 \cap U^j(x_2(\xi))$ loses

a vertex. Set $U_2 - U^j(x_2(\xi))$ gets a vertex because a vertex that is not in U_2 neither in $U^j(x_2(\xi))$ comes to U_2 (this is not possible because the distances to $x_2(\xi)$ are increasing and the distances to x_1 do not change) or because a vertex that belongs to U_2 and to $U^j(x_2(\xi))$ goes out of $U^j(x_2(\xi))$ (this is not possible either because the point $x_2(\xi)$ is going towards j). If $U_0 \cap U^j(x_2(\xi))$ loses a vertex then this vertex goes to $U_2 \cap U^j(x_2(\xi))$ so that the slope does not change. Finally, $U_2 \cap U^j(x_2(\xi))$ never loses a vertex when the movement towards j because it makes the distance shorter, except when $r_2(\xi) = r_1$. If this occurs we will be in case 3.

Case 2). Let $r_1 < r_2 = r$.

Like in case 1, $x_2 \notin D$ then $x_2 \notin V$ so it is an interior point of an edge $[i, j]$. We are also going to show how moving this point x_2 on its edge, the function $f_\lambda(U; (x_1, x_2))$ does not decrease by analysing its slope as a function of the distance from x_2 to the vertex i .

Let U_2^* be the set of farthest vertices assigned to x_2 ; i.e. $U_2^* = \{u \in U_2 : r_2 = d(x_2, u)\}$. Two cases must be considered, 2a) $U_2^* \subseteq U^j(x_2)$ or 2b) $U_2^* - U^j(x_2) \neq \emptyset$. In case 2a) is $r_2(\xi) < r_2$ and, since $r_1 < r_2(\xi)$, the function $f_c(U; (x_1, x_2(\xi))) = r_2(\xi)$ has slope -1 , for small ξ . In case 2b) is $r_2 \leq r_2(\xi)$ and, since $f_c(U; (x_1, x_2)) = r_2(\xi)$, the function $f_c(U; (x_1, x_2(\xi)))$ has slope $+1$. Therefore the slopes $s_2^j(\xi)$ (if the movement is towards j) and $s_2^i(\xi)$ (if the movement is towards i) of the function f_λ when x_2 is moved on the edge are obtained in these cases as follows. In case 2a) they are $s_2^j(\xi) = +\lambda + (1 - \lambda) \cdot m_2^j(\xi)$ and $s_2^i(\xi) = -\lambda + (1 - \lambda) \cdot m_2^i(\xi)$. In case 2b) they are $s_2^j(\xi) = -\lambda + (1 - \lambda) \cdot m_2^j(\xi)$ and $s_2^i(\xi) = +\lambda + (1 - \lambda) \cdot m_2^i(\xi)$. Here $m_2^j(\xi)$ and $m_2^i(\xi)$ are the slopes of the median function when x_2 is moved on the edge, computed like in case 1. In both cases 2a) and 2b) the sum of both slopes is $s_2^j(\xi) + s_2^i(\xi) = (1 - \lambda)(m_2^i(\xi) + m_2^j(\xi)) \leq 0$. Therefore one of the slopes, $s_2^j(\xi)$ or $s_2^i(\xi)$, is not positive. Notice that the slope $m_2^j(\xi)$ can increase in the same conditions as in the case above; i.e. if points $x_2(\xi)$ reaches a vertex or $r_1 = r_2(\xi)$.

Case 3). Let $r_1 = r_2 = r$.

In this case the two points x_1 and x_2 will be shifted the same distance ξ in the direction in which both $r_1(\xi)$ and $r_2(\xi)$ increase or decrease.

Since $x_2 \notin D$ then $x_2 \notin E(R)$, therefore $r \notin R$ because $x_2 \in E(r)$. Then $x_1 \notin LC(r)$ and $x_1 \notin V$ because otherwise $r = r_1 \in R$ and $x_2 \in E(r_1)$. Therefore x_1 is an interior point and is not in $LC(r_1)$; then there is a direction to move it on its edge in which the length of the shortest paths to all the user vertices at distance r_1 decreases. Let $[i_1, j_1]$ denote the edge containing x_1 where the j_1 is chosen in such a way that, when $x_1(\xi)$ is moved towards j_1 , $r_1(\xi) = f_c(U_1; x_1(\xi))$ decreases. Similarly for x_2 there is a direction to move it on its edge in which the length of the shortest paths to the user vertices at distance r_2 decreases and let $[i_2, j_2]$ be the edge containing x_2 such that, when $x_2(\xi)$ is moved towards j_2 , $r_2(\xi) = f_c(U_2; x_2(\xi))$ decreases. Thus, the slopes of $r_1(\xi)$ and $r_2(\xi)$ are equal to -1 when x_1 and x_2 are moved towards j_1 and j_2 , and they have slopes $+1$ when x_1 and x_2 are moved towards i_1 and i_2 , respectively.

Therefore the slope of the objective function f_λ when both $x_1(\xi)$ and $x_2(\xi)$ are simultaneously moved the same distance ξ on their edges is $s_\lambda^j(\xi) = -\lambda + (1 - \lambda)(m_1^{j_1}(\xi) + m_2^{j_2}(\xi))$, when x_1 and x_2 are moved towards j_1 and j_2 , and $s_\lambda^i(\xi) = +\lambda + (1 - \lambda)(m_1^{i_1}(\xi) + m_2^{i_2}(\xi))$, when x_1 and x_2 are moved towards i_1 and i_2 . One of these two slopes is not positive, because the sum of them is:

$$s_\lambda^i(\xi) + s_\lambda^j(\xi) = (1 - \lambda) \cdot \underbrace{(m_1^{i_1}(\xi) + m_1^{j_1}(\xi))}_{\leq 0} + \underbrace{(m_2^{i_2}(\xi) + m_2^{j_2}(\xi))}_{\leq 0} \leq 0.$$

Notice that these simultaneous movements can be done until one of the two facility points, $x_1(\xi)$ or $x_2(\xi)$, reaches the end of its edge (that is a vertex) or a local center associated to two of the farthest user vertices assigned to it. If it reaches the end of the edge then the value of r is the distance from x_1 or x_2 to an user vertex. If it reaches a local center then the value of r is its range. Moreover in both cases the other facility point is in $E(r)$ with $r \in R$. \diamond

References

- Goldman, A.J. (1971). Optimal Center Location in Simple Network. *Transportation Science*, **5**, 212-221.
- Hakimi, S.L. (1964). Optimum Locations of Switching Center and the absolute Center

and Medians of a Graph. *Operations Research*, **12**, 450-459.

Halpern, J. (1976). The location of a centdian convex combination on an undirected tree. *Journal of Regional Science*, **16**, 237-245.

Halpern, J. (1978). Finding Minimal Center Median Convex Combination (centdian) of a Graph. *Managament Science*, **24**, 5, 535-544.

Hooker, J.N., Garfinkel, R.S., & Chen, C.K. (1991). Finite Dominating Sets For Network Location Problems. *Operation Research*, **39**, 1, 100-118.

Horowitz, E. & Sahni, S. (1978). *Fundamentals of Computer Algorithms*. Computer Science Press, Inc.

Moreno, J.A. (1985). A Correction to the definition of Local Center. *European Journal of Operational Research*, **20**, 382-385.