A generalized model of equality measures in network location problems

M.C. López-de-los-Mozos[a, b, *], Juan A. Mesa[a, b], Justo Puerto[c]

[a]*Departamento de Matemática Aplicada I. Universidad de Sevilla, Spain*
[b]*Departamento de Matemática Aplicada II. Universidad de Sevilla, Spain*
[c]*Departamento de Estadística e Investigación Operativa. Universidad de Sevilla, Spain*

Abstract

In this paper, the concept of the ordered weighted averaging operator is applied to define a model which unifies and generalizes several inequality measures. For a location x, the value of the new objective function is the ordered weighted average of the absolute deviations from the average distance from the facilities to the location x. Several kinds of networks are studied: cyclic, tree and path networks and, for each of them, the properties of the objective function are analyzed in order to identify a finite dominating set for optimal locations. Polynomial-time algorithms are proposed for these problems, and the corresponding complexity is discussed.
© 2006 Elsevier Ltd. All rights reserved.

Keywords: Location; Networks; Equality measures

1. Introduction

Recent studies in network location problems have been focused on designing new objective functions which generalize different location measures. The incorporation of several problems in a globalized model enables the exploration of the incidence of the geometrical structure of the network with respect to different criteria. In this way, several unifying location models have been introduced, the most recent of which is based on the concept of an ordered averaging operator. The global model obtained by means of such a concept generalizes the most important location criteria according to the vector to be sorted. When such a vector is the classic weighted distance vector, the ordered median model is obtained. This model provides a unified framework which includes both the center and the median as special cases, as well as several related measures (such as the cent-dian and the k-centrum). The single ordered median problem has been studied in several types of cyclic networks and tree networks, both from the theoretical and algorithmic points of view, and also recent studies have focused on the multifacility version and other related instances of this problem in several kinds of spaces (see [1–4] and references therein).

However, the ordered median model does not generalize a group of objective functions: the equality measures, whose formulation and behavior present structural differences with respect to the center and median measures. Nevertheless, the equality measures (and more generally, the issue of equity) are relevant in facility location problems in the public

---

* Corresponding author. Tel.: +34 954554387; fax: +34 954557878.

E-mail addresses: mclopez@us.es (M.C. López-de-los-Mozos), jmesa@us.es (J.A. Mesa), puerto@us.es (J. Puerto).

sector and, for this reason, have received growing attention in the last decade, as witnessed by the proliferation of papers which deal with inequality measures in the literature.

Most of the inequality measures proposed by Marsh and Schilling [5] have been formulated in several metric spaces and the corresponding single facility location problems have been solved by means of efficient algorithms. Some of the most frequently used inequality measures are those in which the cost function is an operator on the vector of absolute deviations. The weighted sum operator and the maximum operator give rise to the mean absolute deviation and the maximum absolute deviation measures, respectively. Although both operators are the same as in the classic median and center measures, in this case the objective is a monotone function of the deviations from the average distance, and the behavior and properties of such absolute deviation measures are radically different from the respective properties of the median and center measures. Therefore, the distance function vector considered gives rise to different problems with different behavior and properties and, in the same way as the mean (or the maximum) absolute deviation problem cannot be obtained from the median (or the center) problem, neither can the ordered absolute deviation model be obtained from the ordered median model (nor vice-versa). These arguments endorse the need to consider these two models separately, since they are structurally different.

Following this line of reasoning, the aim of this paper is to study the ordered absolute deviation model, whose objective function is formulated as the ordered weighted average of the absolute deviations from facilities to the median value. The corresponding single facility location problem is studied in cyclic, tree and path networks. The strategy for solving each of these problems is based on identifying a Finite Dominating Set (FDS), that is, a finite set of points which is valid for all instances of the same location problem and which contains an optimal location (see [6] for further details).

The remainder of the paper is organized as follows. Section 2 is devoted to introducing the notation and formulating the general problem. In Section 3, this problem is solved in cyclic, path and tree networks, and some particular cases are studied in Section 4. Finally, the main conclusions and further research are summarized.

2. Formulation of the problem

Let $N(V, E)$ be a connected and undirected network with node set V, $|V| = n$ and edge set E, $|E| = m$. Each edge $e = [u, v]$ is represented by a continuous rectifiable arc with positive length l_{uv}. Let x be a point on an edge of N whose exact location is determined by its distance (along the edge) from a prescribed endpoint of the edge. Thus, any real number $x \in [0, l_{ij}]$ denotes the location in the edge $[u, v]$ for which the length of the subedge $[u, x]$ is x. We let N denote the continuum set of points of the network.

The edge lengths induce a distance function on N. For any two points x, y in N, $d(x, y)$ will denote the length of a shortest path connecting x and y. Thus, N is a metric space with respect the above distance function. Associated with each node v_i, let $w_i \geqslant 0$ denote the demand weight of the node (e.g. the number of customers residing at node v_i). Without any loss of generality it may be assumed that $\sum_{v_i \in V} w_i = 1$, and each w_i can be interpreted as the fraction of the demand originated at v_i.

For any $x \in N$, the median function of x is given by $M(x) = \sum_{v_i \in V} w_i d_i(x)$ (where $d_i(x) = d(v_i, x)$). Hence $f_i(x) = w_i |d_i(x) - M(x)|$ is the absolute deviation from the v_i-customers to a facility located at x. Two different measures are considered to compute a global measure of this deviation over the network. The mean absolute deviation measure from x, defined as $D(x) = \sum_{v_i \in V} f_i(x)$, and the maximum (weighted) absolute deviation from x, given by $Z(x) = \max_{v_i \in V} f_i(x)$.

In a general network, the mean absolute deviation problem was initially solved by Berman and Kaplan [7] in $O(mn^2)$ time. The algorithm proposed by Tamir [8] further reduced this complexity to $O(mn \log n)$. On the other hand, the maximum absolute deviation problem has been solved in $O(mn^2 \log n)$ time by López-de-los-Mozos and Mesa [9], and this complexity has also been reduced to $O(mn^2)$ by Mesa et al. [10]. The same authors [10] give algorithms to solve the mean and the maximum absolute deviation in respectively $O(n^2)$ and $O(n^3)$ time on tree networks.

A first generalization of both measures is provided by means of the k-sum deviation measure, which translates the well-known k-centrum concept to this equality framework. In this way, the k-sum deviation function measures the sum of the k greatest values of the vector $f(x) = (f_1(x), \ldots, f_n(x))$. The mean absolute deviation and the maximum absolute deviation are obtained when $k = n$ and 1, respectively, and the corresponding k-sum deviation minimization problem can be solved in $O(mn^2)$ time in general networks [11].

The new generalization is derived from applying the ordered averaging operator to the vector $f(x)$. Let $\mathscr{P}(n)$ be the set of all permutations of the set $\{1, \ldots, n\}$. For each $x \in N$, let $\tau^* \in \mathscr{P}(n)$ be a permutation which sorts

Table 1
Summary of complexity for the considered cases

	Cyclic network	Tree network	Path tree
General	$O(mn^2 \log n)$	$O(n^3 \log n)$	$O(n^2 \log n)$
Convex	$O(mn^2 \log n)$	$O(n^2 \log^2 n)$	$O(n^2 \log n)$
Concave	$O(mn^2 \log n)$	$O(Hn^2)^a$	$O(n^2)$
Concave and unweighted	$O(mn^2)$	$O(Hn^2)^a$	$O(n^2)$
Discrete	$O(n^2 \log n)$	$O(n^2)$	$O(n^2)$

[a] H is the number of leaves (degree-one vertices) of the tree.

the components of the distance function vector into non-decreasing order, that is $f_{\tau^*}(x) = (f_{(1)}(x), \ldots, f_{(n)}(x))$, with $f_{(1)}(x) \leqslant f_{(2)}(x) \leqslant \cdots \leqslant f_{(n)}(x)$. Given a vector $\lambda = (\lambda_1, \ldots, \lambda_n) \in \mathbb{R}^n$, with $\lambda_i \geqslant 0$, $\forall i = 1, \ldots, n$, the ordered weighted operator $F_\lambda(x)$ is defined as the scalar product between the vector λ and the sorted vector $f_{\tau^*}(x)$, that is, $F_\lambda(x) = \sum_{i=1}^n \lambda_i f_{(i)}(x)$ [4].

When the distance functions $f_i(x)$ are the classic weighted distances $w_i d_i(x)$, $i = 1 \ldots, n$, we obtain the ordered median function $\sum_{i=1}^n \lambda_i w_{(i)} d_{(i)}(x)$. The corresponding single, multifacility, and other related location problems have been studied in several kinds of spaces (see [1–4] and the references cited therein).

When $f_i(x) = w_i |d_i(x) - M(x)|$, $i = 1, \ldots, n$, the ordered absolute deviation function $F_\lambda(x) = \sum_{i=1}^n \lambda_i f_{(i)}(x)$ is obtained, (note that in this function, the elements to sort are the $f_i(x) = w_i |d_i(x) - M(x)|$ and not the $d_i(x)$ nor the $w_i d_i(x)$). Hence, the ordered absolute deviation problem is to solve $\min_{x \in N} F_\lambda(x)$.

Clearly, different selections of the λ-vectors give rise to different absolute deviation measures. In this way, $\lambda = (1, \ldots, 1)$ gives the mean absolute deviation $D(x)$, $\lambda = (0, \ldots, 0, 1)$ gives the maximum absolute deviation $Z(x)$ and, conversely, $\lambda = (1, 0, \ldots, 0)$ gives the minimum absolute deviation $\min\{f_i(x), i = 1, \ldots, n\}$. The sum of k greatest values of $f(x)$ (the k-sum deviation) is obtained when the last k components of λ are equal to one and the rest equal to zero, that is, $\lambda = (0, \ldots, 0, 1, \overset{(k}{.}., 1)$. Conversely, from $\lambda = (1, \overset{(k}{.}., 1, 0, \ldots, 0)$ the sum of k smallest values of $f(x)$ is obtained. Likewise, the convex mixture $\mu D(x) + (1 - \mu)Z(x)$ is obtained from $\lambda = (\mu, \ldots, \mu, 1)$, the maximum pairwise difference $\max_i f_i(x) - \min_j f_j(x)$ is derived from $\lambda = (-1, 0, \ldots, 0, 1)$, and, in the same way, other measures belonging to the same equality family can be obtained.

Table 1 displays the complexity of the cases studied in this paper in order to solve the aforementioned single facility location problem $\min_{x \in N} F_\lambda(x)$.

3. Solving the problem

Our approach to solve the problem is based on identifying a FDS on each edge e of the network. To this end, the behavior of the objective function will be studied in each of the subedges in which the edge can be partitioned.

Let $e = [u, v] \in E$ be a given edge of the network. It is well known that for each $i = 1, \ldots, n$, $d_i(x) = d(v_i, x)$ is a continuous, concave and piecewise linear function with at most two segments with slopes 1 and -1, respectively. Its breakpoint (if any) is reached at an interior point $\bar{x}_i \in e$ for which $d(v_i, \bar{x}_i)$ via vertex u is the same as via vertex v. Such a point $\bar{x}_i$ is called an edge bottleneck point of v_i in the edge e (see [12]). Henceforth for simplicity we will said bottleneck point instead of edge bottleneck point. Therefore, the median function $M(x)$ is concave and piecewise linear on e, and all its breakpoints are bottleneck points. Let $B(e)$ be the set of bottleneck points of e. Clearly, $|B(e)| \leqslant n$. The points of set $B(e) \cup \{u, v\}$ determine a partition of the edge. Each closed subedge delimited by two adjacent points of this partition is called a *primary region*.

On the other hand, Tamir [8] proved that each distance function $d(v_j, x)$ intersects $M(x)$ at (at most) four points in each edge. Although an improvement of this bound does not affect the complexity of the algorithms, proof of Theorem 1 uses that such a bound can be reduced to two points and, for this reason, we will now prove this assertion. Let $\bar{x}_j$ be the bottleneck point associated with v_j, and let $(\bar{x}_{j-1}, \bar{x}_j]$ be the half-open set associated with the jth primary region $[\bar{x}_{j-1}, \bar{x}_j]$. Consider the nodal partition $\{V_j, \overline{V}_j\}$ such that $V_j = \{v_i \in V / d(v_i, u) + x \leqslant d(v_i, v) + l_{uv} - x\}$, and $\overline{V}_j = V \setminus V_j$. When $x \in (\bar{x}_{j-1}, \bar{x}_j]$, this partition is unchanged (see [13]). Let $w(V_j)$ and $w(\overline{V}_j)$ be the total weight of vertices in V_j and $\overline{V}_j$, respectively. The slope of the $M(x)$ function in the jth primary region is given by $w(V_j) - w(\overline{V}_j)$,
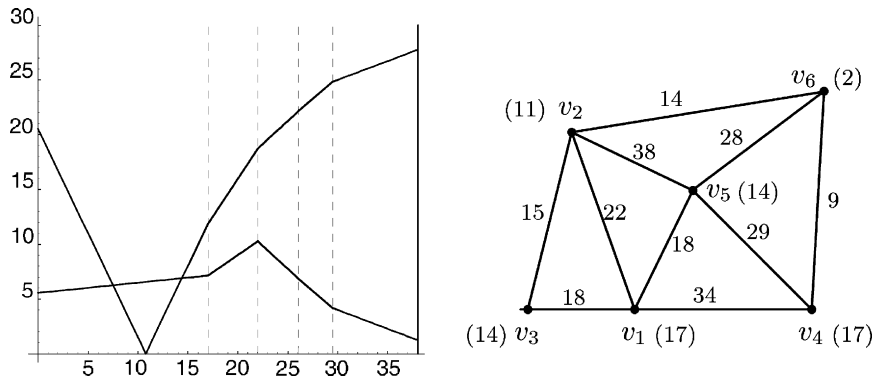
Fig. 1. Intersection points of two f_i-functions in a primary region of this network.

and it holds $|w(V_j) - w(\overline{V}_j)| \leqslant 1$. This fact together with the concavity of both the median and the distance functions implies that there are at most two intersection points of $M(x)$ with each distance function along the edge.

Let $L_1(e)$ be the set of such $2n$ (at most) intersection points. Each closed interval delimited by two consecutive points of the set $B(e) \cup L_1(e) \cup \{u, v\}$ is called a *secondary region*. Clearly, $|B(e) \cup L_1(e) \cup \{u, v\}| \leqslant 3n + 2$. Note that in each secondary region the sign of $d_i(x) - M(x)$ is constant, therefore each $f_i(x) = w_i|d(v_i, x) - M(x)|, i = 1, \ldots, n$, is linear in such a region. However, $f_i(x)$ is neither concave nor convex along the edge, and its breakpoints belong to the set $B(e) \cup L_1(e) \cup \{u, v\}$.

Let $L_2(e) = \{x \in e : f_i(x) = f_j(x), \; i, j = 1, \ldots, n\}$ be the set of intersection points of all pairs of functions $f_i(x)$ and $f_j(x)$ over the edge e, and let $L(e)$ be the union of all these above sets, that is,

$$L(e) = B(e) \cup L_1(e) \cup L_2(e) \cup \{u, v\}.$$

In each subedge $[y_{i-1}, y_i]$ delimited by two consecutive points of $L(e)$, each pair of functions $f_i(x)$ and $f_j(x)$ do not intersect and the order $f_{(1)}(x) \leqslant \cdots \leqslant f_{(n)}(x)$ does not change. Therefore, the objective function $F_\lambda(x)$ is linear in $[y_{i-1}, y_i]$, which implies that its local minimum is reached at an extreme point of this subedge. Consequently, the breakpoints of $F_\lambda(x)$ belong to the set $L(e)$, which implies that $L(e)$ is a FDS over the edge.

To determine the number of points of $L(e)$, we first consider the example shown in Fig. 1, which displays (on the right) a network with six vertices. Edge lengths are shown on the middle of the edges, and vertex weights are the numbers in round brackets. Fig. 1 (on the left) shows the graphs of functions $f_4(x)$ and $f_5(x)$ over the edge $[v_2, v_5]$. The dashed lines delimit the five primary regions. We can observe that these two functions intersect at two points in the first primary region. Therefore, each pair of functions $f_i(x)$ and $f_j(x)$ can intersect $2n$ times along the edge (since there are $O(n)$ primary regions), which implies that $|L_2(e)| \leqslant n^3$. Consequently, the cardinal of the FDS $L(e)$ is upper-bounded by $O(n^3)$.

The following theorem will allow the reduction of the cardinal of $L_2(e)$ from $O(n^3)$ to $O(n^2)$, and, therefore, the cardinal of $L(e)$ will also be reduced to $O(n^2)$.

Theorem 1. *In each edge $e = [u, v]$, each pair of functions $f_i(x)$, $f_j(x)$ intersect at 16 points at most.*

Proof. With each function $f_i(x)$ we associate the set $S_i = \{p_i, q_i, \bar{x}_i\}$, where p_i, q_i are the (at most) two intersection points of $d(v_i, x)$ with the median $M(x)$, and where $\bar{x}_i$ is the bottleneck point associated with v_i. Let m_e be the point at which the median reaches the maximum (if it exists). These four points give (at most) five subedges on edge e (see Fig. 2). In each of such subedges, $f_i(x) = w_i|d_i(x) - M(x)|$ is a piecewise linear and monotone function (increasing or decreasing), and it is either convex (if $d_i(x) > M(x)$) or concave (if $d_i(x) < M(x)$).

Given two functions $f_i(x)$ and $f_j(x)$, we now consider the (at most) seven points of the set $S_i \cup S_j \cup \{m_e\}$, (where $S_j = \{p_j, q_j, \bar{x}_j\}$ is the set associated with f_j). These seven points provide a partition of the edge e in eight subedges
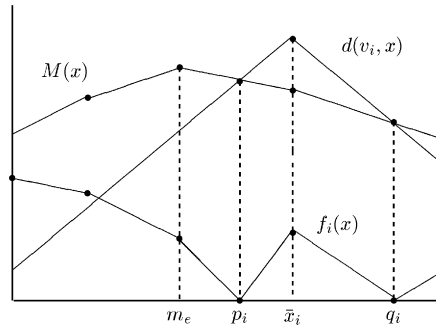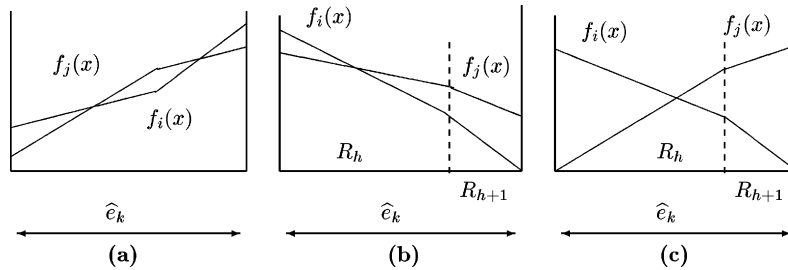
Fig. 2. Partition of an edge.

Fig. 3. Cases in a subedge.

$\widehat{e}_k, k = 1, \ldots, 8$. The demonstration of this theorem is based on proving that in each of these subedges, $f_i(x)$ and $f_j(x)$ intersect at two points at most. To this end several cases in a given subedge $\widehat{e}_k$ are considered:

Case 1: If f_i is convex and f_j is concave (or vice-versa), then clearly they intersect at two points at most (Fig. 3 (a)).

Case 2: If f_i and f_j have the same concavity (suppose, for example, both are concave), they can both have the same monotony or different monotonies.

1. If both have the same monotony and they are, for example, decreasing (as in Fig. 3(b)), then the corresponding distance functions $d_i(x)$ and $d_j(x)$ must be parallel.
   Therefore, in each primary region R_h (totally or partially contained in $\widehat{e}_k$), the slopes of the functions $f_i(x)$ and $f_j(x)$ are $w_i t_h$ and $w_j t_h$, where t_h is the common slope of the absolute value functions $|d_i(x) - M(x)|$ and $|d_j(x) - M(x)|$ in R_h. The subedge $\widehat{e}_k$ can contain more than one primary region. In such a case, if f_i and f_j intersect in R_h, they cannot intersect in the next primary region R_{h+1} since the slope ratio w_i/w_j of $f_i(x)/f_j(x)$ does not vary on the subedge. Consequently, in this case there is (at most) one intersection point.
2. Suppose both functions are concave but with different monotonies (as in Fig. 3 (c), where f_i is decreasing and f_j is increasing). In such a case, if they intersect in R_h, then clearly cannot intersect again in R_{h+1}.

Summarizing, in each subedge $\widehat{e}_k$, both functions intersect at either two points or at one point at most, depending on if they have different or the same concavity, respectively. Since there are eight subedges in the edge, then both functions intersect at 16 points at most, which concludes the proof. $\square$

Consequently, the set $L_2(e)$ of intersection points of all pairs of functions f_i and f_j has $O(n^2)$ points, since $|L_2(e)| = 16\binom{n}{2} = 8n(n-1)$. Therefore, the cardinal of the FDS $L(e)$ is also $O(n^2)$.

From the above results, solving the ordered absolute deviation problem in each edge e takes $O(n^2 \log n)$ time. Indeed, first the points in $L(e)$ are computed and sorted in $O(n^2 \log n)$ time (since the set $L_1(e)$ can be obtained and sorted in $O(n \log n)$ time by applying the procedure of Tamir [8], and also the sorted list of points in $L_2(e)$ can be computed in $O(n^2 \log n)$). On the other hand, given two consecutive points y_{i-1}, y_i in the sorted list $L(e)$, the order $f_{(1)}(x) \leqslant \cdots \leqslant f_{(n)}(x)$ does not change in the interior of the subedge $[y_{i-1}, y_i]$, therefore the objective function at

each point $y_i \in L(e)$ can be recursively computed from the previous point in constant time. Consequently, the effort to compute the objective function at all points in $L(e)$ is $O(n^2)$.

The final complexity for solving the problem over the entire network is $O(mn^2 \log n)$. This effort dominates the preprocessing phase, in which the distance matrix is computed (in $\max\{O(mn \log n), O(n^3)\}$ time).

When the network is a tree the complexity is $O(n^3 \log n)$, since although there are no arc bottleneck points on trees, $L_2(e)$ also has $O(n^2)$ points in each edge. In the case of a path-tree (or chain), there are $O(n^2)$ intersection points of all pairs $f_i(x), f_j(x)$ along the chain, and therefore the complexity is reduced to $O(n^2 \log n)$.

4. Some special cases

In the general model, the components in the λ-vector are non-monotone and the objective function is neither concave nor convex. Therefore, the above algorithm computes the objective function at all points of the FDS. However, when the "lambdas" are monotone (non-increasing or non-decreasing) the objective function presents interesting convexity (or concavity) properties which allow, in some cases, to improve the above complexity. These cases have received specific attention in the literature, and they are known as the convex and the concave case, respectively.

4.1. The convex case

First we consider the case for which the λ-parameters are in ascending order, that is, $0 \leqslant \lambda_1 \leqslant \lambda_2 \leqslant \cdots \leqslant \lambda_n$. By applying the convexity of each distance function in a tree, Kalcsics et al. [1] showed that the ordered median function $\sum_{i=1}^{n} \lambda_i w_{(i)} d_{(i)}(x)$ is convex in each path of a tree. Therefore, the ordered operator $F_\lambda(x) = \sum_{i=1}^{n} \lambda_i f_{(i)}(x)$ is also convex over a real line in which each $f_i(x)$ is convex, $i = 1, \ldots, n$. Since in a primary region each $f_i(x) = w_i |d_i(x) - M(x)|$ $(i = 1, \ldots, n)$ is a piecewise linear and convex function (with at most one breakpoint), then $F_\lambda(x)$ is piecewise linear and convex over each primary region of the network. This implies that the local minimum in each edge e is reached at a point of the FDS, $L(e)$. Consequently, the complexity for the solution of the problem in a network is the same as in the general case.

4.1.1. The convex case in a tree

For tree networks the above complexity reduces to $O(n^2 \log^2 n)$. To obtain this bound we first observe that, although the objective function is neither concave nor convex on any path of the tree, it is convex on each edge $e = \{u, v\}$ and its breakpoints belong to the set $\{u, v\} \cup L_1(e) \cup L_2(e)$. We will use this fact to solve the problem in each edge in $O(n \log^2 n)$ time.

To this end, in a preprocessing phase we recursively compute the median function at all vertices of the tree (in $O(n)$ time). Once these values have been computed, the median function in each edge e is given by $A_e + B_e x$, and these coefficients can be obtained in constant time. Restricting ourselves to the edge $e = \{u, v\}$, we then compute and sort the elements of $L_1(e)$ (intersection points of the median with the distance functions) in $O(n \log n)$ time. Using the convexity of the objective function on the edge, we can apply a binary search on the set $\{u, v\} \cup L_1(e)$ to find a secondary region (a pair of consecutive points of such a set), containing the solution x_e^* of the restricted problem. Since it requires $O(n \log n)$ time to evaluate the objective function at each point of $L_1(e)$, the secondary region $[y_{j-1}, y_j]$ can be identified in $O(n \log^2 n)$ time. We remark that in this region each $f_i(x) = w_i |d(v_i, x) - (A_e + B_e x)|$, $i = 1, \ldots, n$, is a linear function of the parameter x, since $d(v_i, x) = d(v_i, u) + x$, if $d(v_i, u) \leqslant d(v_i, v)$, and $d(v_i, x) = d(v_i, v) + l_{uv} - x$, otherwise.

To find the optimum x_e^* located in $[y_{j-1}, y_j]$, we use the general parametric procedure of Megiddo [14], with the modification in Cole [15] (see these references for a detailed discussion of the parametric approach). The master program that we apply is the sorting of the n linear functions $\{f_i(x), i = 1, \ldots, n\}$ (using x as the parameter), and the test for determining the location of a given point x' with respect to x_e^* is based on calculating the objective function $F_\lambda(x)$ and determining its one-side derivatives at x'. This can be done in $O(n \log n)$ time. We now conclude that with the above test, the parametric approach in [14,15] will find x_e^* in $O(n \log^2 n)$ time.

Finally, when the tree is a path this procedure does not improve the complexity of the exhaustive evaluation of the objective function at all the points of the FDS. Therefore, the complexity of this case is $O(n^2 \log n)$.
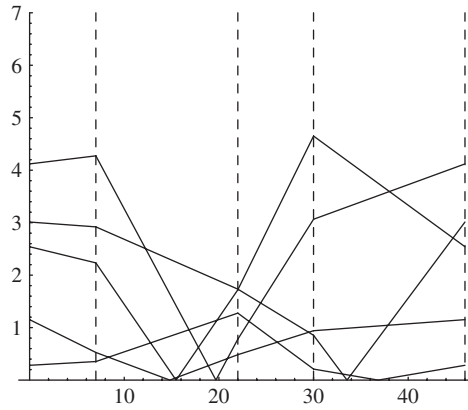
Fig. 4. Intersection points of the f_i-functions in a chain tree.

4.2. The concave case

This case arises when $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \geqslant 0$. Since each $f_i(x) = w_i |d_i(x) - M(x)|$, $i = 1, \ldots, n$ is linear (and therefore concave) in each secondary region, then the objective function is also concave in such a region (see [1]). This implies that in each secondary region the local minimum is reached at one of its extreme points, therefore the set $B(e) \cup L_1(e) \cup \{u, v\}$ is a FDS for the problem in the edge $e = [u, v]$.

This set (with $O(n)$ points) can be obtained and sorted in $O(n \log n)$. However, at each point y_j in this set the objective function cannot be recursively computed in constant time from the previous point y_{j-1}, since the permutation for which $f_1(x) \leqslant \cdots \leqslant f_{(n)}(x)$ is not constant when x varies in the secondary region $[y_{j-1}, y_j]$, even in the case of a path tree. This is due to the fact that the functions $\{f_i(x), i = 1, \ldots, n\}$ can intersect at interior points of such an interval, as can be appreciated in the following example.

Fig. 4 represents a chain tree whose length is 46, with five vertices located, respectively, at the points: $\{0, 7, 22, 30, 46\}$, and whose vertex weights are $\{3, 15, 7, 30, 9\}$. For $i = 1, \ldots, 4$, the value $|v_{i+1} - v_i|$ is the length of the edge $[v_i, v_{i+1}]$ (in Fig. 4 each edge is delimited by dashed lines). Let L_1 be the set of all intersection points of the median with the distance functions along the chain. Since there are no arc bottleneck points on trees, $V \cup L_1$ is a FDS for the concave problem in this path tree.

Fig. 4 displays the five functions $f_i(x) = w_i |d_i(x) - M(x)|$, $i = 1, \ldots, 5$, in the entire chain. We can observe that the third edge $[v_3, v_4]$ is also a secondary region since it does not contain points in L_1 (which correspond to contact points of f_i-functions with the X-axis), and the figure shows several intersection points of the five functions in such an interval.

Consequently, to compute the objective function at each point y_j in the FDS it is necessary to sort the $f_i(y_j)$ $(i = 1, \ldots, n)$ values, which requires $O(n \log n)$ time. Therefore, in the concave case the complexity of this procedure for finding the local minimum in each edge of the network is the same as in the general case, $O(n^2 \log n)$, since although the cardinal of the FDS is lower, the evaluation of the objective function at each point by means of this procedure requires more complexity.

However, when the network is a tree or a path-tree the above complexities can be reduced. Indeed, these improvement are based on applying the techniques described by Tamir [16], which generates all the sorted vectors $f_{\tau^*}(y_j)$, $\forall y_j \in$ FDS.

For the sake of simplicity, first we will apply such a procedure to solve the concave case in a path tree.

4.2.1. The concave case in a path tree

Let $P(v_1, v_n)$ be a path tree with length l in which the n vertices are increasingly located in the line segment $[0, l]$. As it has already been seen, the set $V \cup L_1$ is a FDS for the concave case on a path tree, and its cardinal K is upper-bounded by $3n$, since each distance function intersects the median at (at most) two points along the segment. Such a set can be obtained and sorted in $O(n \log n)$ time. Let $v_1 = y_1 \leqslant y_2 \leqslant \cdots \leqslant y_K = v_n$ be the points of the FDS sorted into ascending order on the real line. The distance between y_j and a vertex v_i is given by $d(v_i, y_j) = |v_i - y_j| = y_j - v_i$

(if $v_i \leqslant y_j$), and $d(v_i, y_j) = v_i - y_j$ (otherwise). To solve the problem it is necessary to obtain all the vectors $f_{\tau^*}(y_j) = (f_{(1)}(y_j), \ldots, f_{(n)}(y_j))$, with $f_{(1)}(y_j) \leqslant f_{(2)}(y_j) \leqslant \cdots \leqslant f_{(n)}(y_j)$.

To this end, we first recursively compute the values $M(y_j) = M_j$ at all points of the FDS (in $O(n)$ time). On the other hand, each $f_{\tau^*}(y_j)$ consists of the largest n elements of the multi-set $\widehat{f}(y_j)$ given by

$$\widehat{f}(y_j) = \{w_i(d(v_i, y_j) - M_j), \ i = 1, \ldots, n\} \cup \{-w_i(d(v_i, y_j) - M_j), \ i = 1, \ldots, n\}.$$

By applying the procedure of Tamir [16], all these multi-sets $\widehat{f}(y_j)$, $j = 1, \ldots, K$ can be (individually) sorted in $O(n^2 + Kn)$ time, and therefore in $O(n^2)$ time (since $K \in O(n)$). Consequently, the concave case in a path tree is solvable in $O(n^2)$ time, since this is the total effort to generate all the vectors $f_{\tau^*}(y_j)$.

4.2.2. The concave case in a tree

The above procedure gives a complexity of $O(n^3)$ if the network is a tree $T(V, E)$. This is the worst case complexity, however, the average complexity of the problem can be improved if we examine $|L_1|$ (the cardinal of L_1, the set of all intersection points of the median with the distance functions along the tree). Let m^* be a median vertex (this is, a vertex which minimizes the median function), let v_h be a leaf (or vertex whose degree is 1), let H be the number of leaves of the tree, and let us consider the path $P(m^*, v_h)$. Since each distance function intersects the median at (at most) two points along $P(m^*, v_h)$, L_1 has at most $2n$ points into this path and, consequently, its cardinal is upper-bounded by $O(Hn)$ (which is also the bound for the cardinal K of $V_1 \cup L_1$, the FDS of the problem). Therefore, the problem can be solved in $O(Hn^2)$ time. To do this, we first compute and sort (in each path) the points of the FDS, and then we recursively compute the median value at all these points. In a second phase, we apply the recursive approach by Tamir [16] based on the centroid decomposition of the tree to generate all the sorted vectors $f_{\tau^*}(y_j)$, $y_j \in$ FDS in $O(Kn)$ time and, therefore, in $O(Hn^2)$ time, which is the overall complexity of the problem.

We point out that this complexity is based on the cardinal of L_1. Clearly, the complexity of the problem is quadratic for trees where $|L_1|$ is linear. From the aforementioned argument, $O(Hn)$ is an upper bound of $|L_1|$, however, it is not clear if such a bound is achieved. In fact, it is possible that $|L_1|$ can be improved to $O(n)$ points, as it can be appreciated from the following computational experience.

We have considered a set of randomly generated trees and, in each of them, we have computed the cardinal of L_1. Two sizes n of trees together with a variable number H of leaves have been considered such that each type of tree is characterized by the input $\{n, H\}$. Table 2 (on the left) presents the results for $n = 100$ vertices and $H = 10, 30, 50, 70, 80, 90$ leaves, and (on the right) for $n = 200$ vertices and $H = 50, 100, 120, 150, 170, 190$ leaves. In all cases, weights and lengths were uniformly generated in the range $[1, 100]$ and $[1, 1000]$, respectively, by means of a random routine, and the experiment was programmed on a PC.

For each type $\{n, H\}$ of tree, 100 instances were generated and, for each instance, the value $|L_1|$ was computed. Table 2 shows, for several p values, the number of instances for which $|L_1| \leqslant pn$ in each 100 cases tested. For example, the column corresponding to $n = 200$ and $H = 170$ indicates that, in 100 instances generated, $|L_1| \leqslant n = 200$ in 0 instances, $|L_1| \leqslant 2n = 400$ in 41 instances, $|L_1| \leqslant 3n = 600$ in 97 instances, and $|L_1| \leqslant 4n = 800$ in all the instances. The results show that $|L_1| \leqslant 4n$ in all the cases considered, which seems to confirm the linear order of $|L_1|$. From above, this bound would give a quadratic time for the problem. However, we remark that it is not yet proved whether we can get $O(n)$ points for $|L_1|$ in a tree and (at least as we are aware), this question still remains open.

4.2.3. The concave and unweighted case

A further improvement can be done when all weights are equal to one (or equivalently, when $w_i = w > 0$, $i = 1, \ldots, n$), as it is explained as follows. Consider $\lambda_1 \geqslant \lambda_2 \geqslant \cdots \geqslant \lambda_n \geqslant 0$ and $w_i = 1$, $i = 1, \ldots, n$. Therefore, $f_i(x) = |d(v_i, x) - M(x)|$, $i = 1, \ldots, n$, where, in order to preserve the scale, $M(x) = 1/n \sum_{i=1}^{n} d(v_i, x)$.

In a path tree, the cardinal of the FDS is the same as in the weighted case ($O(n)$ points), and also the permutation arranging the n f_i-values changes between two consecutive points of the FDS. Therefore, the complexity of the concave and unweighted case in a path tree is the same as in the weighted case: $O(n^2)$. A similar argument in a tree again gives the already established complexity of $O(h_T n^2)$.

We now turn to a general network and consider the concave and unweighted problem in each edge of the network. Next we propose a procedure to solve this restricted problem in $O(n^2)$.

To this end, it is necessary a preprocessing phase in which the distances from each vertex to the remaining vertices are sorted into ascending order (in $O(n^2 \log n)$ time), and also for each edge $e = [u, v]$ the points of the FDS $B(e) \cup$

Table 2
Number of instances for which $|L_1| \leqslant pn$ over 100 cases

p	H					
	10	30	50	70	80	90
$n = 100$						
1	0	0	0	0	0	0
2	100	91	71	73	46	59
3		100	100	100	95	100
4					100	

p	H					
	50	100	120	150	170	190
$n = 200$						
1	0	0	0	1	0	0
2	75	47	51	70	41	46
3	100	100	100	98	97	99
4				100	100	100

$L_1(e) \cup \{u, v\}$ are computed and sorted (in $O(n \log n)$ time). The overall complexity of this phase is $O(mn \log n)$. The following procedure computes the objective function at each point y_j of the FDS in $O(n)$ time.

1. First, the median value $M(y_j) = M_j$ is recursively computed at all points y_j of the FDS (in $O(n)$ time).
2. Suppose y_j belongs to the rth primary region. As it has been pointed out in Section 3, all points in this region determine a nodal partition $\{V_r, \overline{V}_r\}$, such that these sets contain the vertices whose distances to y_j are computed via vertex u or vertex v, respectively. Therefore,

$$d(v_t, y_j) = \begin{cases} d(v_t, u) + y_j = a_t & \forall\, v_t \in V_r, \\ d(v_i, v) + l_{uv} - y_j = b_t & \forall\, v_t \in \overline{V}_r. \end{cases}$$

From the preprocessing phase, the sorted distances $\{a_t, \forall v_t \in V_r\}$ and $\{b_t, \forall v_t \in \overline{V}_r\}$ can be obtained in $O(n)$. Let A_j and B_j be the respective sequences.
3. Insert $M(y_j) = M_j$ in both sorted sequences (in $O(\log n)$), and then construct A'_j and B'_j, which, respectively, contain the sorted differences of M_j and the elements on its left and its right. This can be done in $O(n)$ time. Finally merge (in $O(n)$ time) these two last lists. The resultant list is the sorted vector $f_{\tau^*}(y_j) = (f_{(1)}(y_j), \ldots, f_{(n)}(y_j))$.

Consequently, in each edge the effort to compute the objective function for all points in the FDS is $O(n^2)$, which gives a total complexity of $O(mn^2)$ over the entire network (note that this effort dominates the complexity of computing the distance matrix).

4.3. The discrete case

In this problem, the set of potential optimal locations is restricted to set V. In general, in order to solve the problem it is necessary to evaluate the objective function at each vertex v_j, which requires the vector $(f_1(v_j), \ldots, f_n(v_j))$, to be sorted into ascending order. This can be also done in $O(n^2)$ time in a path-tree and in a tree by, respectively, applying the above procedures in Tamir [16]. For the case of a network, if we suppose the distance matrix is already computed, the total complexity to solve the problem is $O(n^2 \log n)$.

5. Conclusions

In this paper, we have extended the concept of ordered weighted operator to define a model which unifies several equality criteria. Although the resultant model shares the same basic idea as the ordered median model (in the sense

that both deal with flexible constructions which generalize several location problems), the distance function vector considered as well as the equality measures incorporated give rise to a specific objective function clearly distinguished from the ordered median objective, at least in the same way that, in location theory, the center and the median measures are different from the maximum and the median absolute deviation measures.

For algorithmic purposes we have studied and identified the FDS for the problem in several scenarios, and, for each of the cases considered, we have proposed polynomial time algorithms as well as discussed the complexity of each of them.

Further research points out toward a possible improvement on the complexity of some cases (as we have already seen in the previous section), as well as the extension of the model to multifacility problems. In addition, although the equality measures have been slightly studied in other metric spaces (as, for example, $\mathbb{R}^2$), it would also be interesting to formulate this model in these spaces.

Acknowledgment

References

[1] Kalcsics J, Nickel S, Puerto J, Tamir A. Algorithmic results for ordered median problems defined on networks and the plane. Operations Research Letters 2002;30:149–58.
[2] Kalcsics J, Nickel S, Puerto J. Multifacility ordered median problems on networks: a further analysis. Networks 2003;41:1–12.
[3] Nickel S, Puerto J. A unified approach to network location problems. Networks 1999;34:283–90.
[4] Nickel S, Puerto J. Location theory—a unified approach. Springer series in operations research and decision theory. Berlin: Springer; 2005.
[5] Marsh MT, Schilling DA. Equity measurement in facility location analysis: a review and framework. European Journal of Operational Research 1994; 74:1–17.
[6] Hooker JN, Garfinkel RS, Chen CK. Finite dominating sets for network location problems. Operations Research 1991;39:100–18.
[7] Berman O, Kaplan EH. Equity maximizing facility location schemes. Transportation Science 1990;24:137–44.
[8] Tamir A. On the complexity of some class of location problems. Transportation Science 1992;26:352–4.
[9] López-de-los-Mozos MC, Mesa JA. The maximum absolute deviation measure in location problems on networks. European Journal of Operational Research 2001;135:184–94.
[10] Mesa JA, Puerto J, Tamir A. Improved algorithms for several location problems with equality measures. Discrete Applied Mathematics 2003;130:437–48.
[11] López-de-los-Mozos MC, Mesa JA. Certain (related) problems concerning two absolute deviation measures. Communication presented to EWGLA XIV, September 2003, Corfu, Greece, nonpublished paper.
[12] Church RL, Garfinkel RS. Locating an obnoxious facility on a network. Transportation Science 1978;12:107–18.
[13] Hansen P, Zheng M. An algorithm for the minimum variance point of a network. RAIRO *Recherche Opérationnelle/Operations Research* 1991;25:119–26.
[14] Megiddo N. Applying parallel computation algorithms in the design of serial algorithms. Journal of the Association for Computing Machinery 1983;30:852–65.
[15] Cole R. Slowing down sorting networks to obtain faster sorting algorithms. Journal of the Association for Computing Machinery 1987;34: 200–8.
[16] Tamir A. Sorting weighted distances with applications to objective function evaluations in single facility location problems. Operations Research Letters 2004;32:249–57.