ELSEVIER

Discrete Applied Mathematics 118 (2002) 263–278

DISCRETE APPLIED MATHEMATICS

The centdian subtree on tree networks

Arie Tamir[a,*], Justo Puerto[b,1], Dionisio Pérez-Brito[c,2]

[a]*Department of Statistics and Operations Research, School of Mathematical Sciences, Tel-Aviv University, Tel-Aviv 69978, Israel*

[b]*Departamento de Estadística, Investigación Operativa. Universidad de Sevilla, Sevilla, Spain*

[c]*Departamento de Estadística, I.O. y Computación. Universidad de La Laguna, La Laguna, Spain*

Received 30 November 1999; received in revised form 27 September 2000; accepted 22 January 2001

Abstract

This paper describes an $O(n \log n)$ algorithm for finding the optimal location of a tree shaped facility of a specified size in a tree network with n nodes, using the centdian criterion: a convex combination of the weighted average distance and the maximum weighted distance from the facility to the demand points (nodes of the tree). These optimization criteria introduced by Halpern, combine the weighted median and weighted center objective functions. Therefore they capture more real-world problems and provide good ways to trade-off minisum (efficiency) and minimax (equity) approaches. © 2002 Elsevier Science B.V. All rights reserved.

Keywords: Location; Path; Tree; Networks

1. Introduction

In the last years there has been a growing interest in studying the location of connected structures on graphs. Hakimi et al. [8] focused on the complexity of solving 64 versions of that problem. The different versions are derived by considering such elements as locating one or $p > 1$ facilities, whether the facilities are paths or tree shaped, whether the underlying network is a tree or a general network, and the objective function of the problem. The objective functions considered are the most classical, the minimization/maximization of the average distance or the maximum distance to service facilities. The corresponding solution concepts were called median/antimedian and center/anticenter.

* Corresponding author.

E-mail addresses: atamir@math.tau.ac.il (A. Tamir), puerto@cica.es (J. Puerto), dperez@ull.es (D. Pérez-Brito).

[1] Partially supported by Spanish DGICYT grant no. PB97-0707.

[2] Partially supported by the Canary Government through the project PI1999/116.

PII: S0166-218X(01)00199-8

Since the median approach is based on averaging, it often provides a solution in which remote and low-population density areas are discriminated against in terms of accessibility to public facilities, as compared with centrally situated and high-population density areas. For this reason, an alternative approach, involving the maximum distance between any customer and closest facility can be applied. This approach is referred to as the center solution concept. The minimax objective primarily addresses geographical equity issues. It is of particular importance in spatial organization of emergency service systems. On the other hand, locating a facility at the center may cause a large increase in the total distance, thus generating a substantial loss in spatial efficiency. This has led to a search for some compromise solution concept.

Halpern in 1976 [9] has introduced the λ-centdian as a parametric solution concept based on the bicriteria center/median model in a tree network. He has modeled the corresponding trade-off with a convex combination of the unweighted center and weighted median objectives. More recently, Carrizosa et al., in 1994 [3] presented an axiomatic approach justifying the use of the centdian criterion. Tamir et al. in 1998 [26] generalized this convex combination introducing weights in the center function. They presented a polynomial time algorithm for the p-facility case in a tree, where each one of the p facilities is a point.

Halpern in 1978 [10] studied the properties of the λ-centdian in a graph. Pérez-Brito et al. in 1997 [21] presented a finite dominating set for the p-facility centdian in a graph, and studied the generalized case. The reader is referred to [1,2,11,13,22] for more recent studies on location problems involving the centdian objective.

In this paper we consider the location of a single tree shaped facility (subtree), with specified length, on a tree network with demand points at its nodes, using the generalized centdian objective function, i.e., we minimize a convex combination of the weighted average distance and the maximum weighted distance from the subtree to the demand points. There are reasonable applications of this criterion to locate optimal subtrees. For example, consider the case where the subtree models the railways of a light train to be established in the network. This train connects to a fixed transfer point on the network which might be the center or any other distinguished point. The demand points (customers) travel to the light train during rush hours. To simplify, suppose that all customers start moving at the same time, possibly at different constant speeds. The total travel time to the light train is the weighted sum of distances to the light train. The total transportation cost is proportional to the total travel time. In addition there is also the cost of monitoring the traffic during the rush hours, e.g., police cars and helicopters. This cost is proportional to the length of the rush period, which in turn is equal to the maximum travel time. It is now easy to see that the total cost function, consisting of the transportation and the monitoring costs, corresponds to the generalized centdian objective function. Locating subtree facilities on a network has been the subject of many recent papers [16,8,12,20,23–25]. There exists another interesting research line which looks for characterizations of the set of Pareto-optimal paths in networks [1,5,18]. However, as far as we know, there has not been a study, implementing the centdian criterion with tree shaped facilities.

(We note that there are few papers [1,13] which use this objective with path shaped facilities.)

In the context of locating a single connected facility with specified length on a tree network, our work extends and unifies the single subtree facility models discussed in Shioura and Shigeno 1997 [23], and Tamir 1998 [24]. The former presents a linear time algorithm for locating a subtree of a given length, minimizing the (unweighted) maximum distance from the subtree to the nodes, while the latter has a linear time algorithm minimizing the weighted average distance. (We note that efficient algorithms for locating a single path facility with a specified length, minimizing the average distance on a tree network are described in [17,19].)

The paper is organized as follows: In Section 2 we present a formulation of the problem of finding an optimal centdian subtree which is restricted to contain a specified point of the given tree. This formulation reduces the problem to a convex piecewise linear programming problem whose objective function is a linear parametric problem. In Section 3 a greedy algorithm is developed to evaluate in linear time the objective value of the formulation given in Section 2. In Section 4 we develop an $O(n \log n)$ algorithm for solving the above restricted subtree problem. We then show that an optimal unrestricted centdian subtree contains a point centdian of the given tree. Therefore, the unrestricted problem is reduced to a restricted version. The paper ends with some conclusions and the references cited in the text.

2. Formulation of the continuous centdian subtree problem on a tree

Let $T=(V,E)$ be an undirected tree network with node set $V=\{v_1,\dots,v_n\}$ and edge set $E=\{e_2,\dots,e_n\}$. Each edge e_j, $j=2,3,\dots,n$, has a positive length l_j, and is assumed to be rectifiable. In particular, an edge e_j is identified as an interval of length l_j so that we can refer to its interior points. We assume that T is embedded in the Euclidean plane. Let $A(T)$ denote the continuum set of points on the edges of T. We view $A(T)$ as a connected and closed set which is the union of $n-1$ intervals. Let $P[v_i,v_j]$ denote the unique simple path in $A(T)$ connecting v_i and v_j. Suppose that the tree T is rooted at some distinguished node, say v_1. For each node v_j, $j=2,3,\dots,n$, let $p(v_j)$, the parent of v_j, be the node $v\in V$, closest to v_j, $v\neq v_j$ on $P[v_1,v_j]$. v_j is a child of $p(v_j)$. e_j is the edge connecting v_j with its parent $p(v_j)$. A node v_i is a descendant of v_j if v_j is on $P[v_i,v_1]$. V_j will denote the set of all descendants of v_j.

We refer to interior points on an edge by their distances along the edge from the two nodes of the edge. The edge lengths induce a distance function on $A(T)$. For any pair of points $x,y\in A(T)$, we let $d(x,y)$ denote the length of $P[x,y]$, the unique simple path in $A(T)$ connecting x and y. The path $P[x,y]$ is also viewed as a collection of edges and at most two subedges (partial edges). $P(x,y)$ will denote the open path obtained from $P[x,y]$, by deleting the points x,y, and $P(x,y]$, or $P[y,x)$, will denote the half open path obtained from $P[x,y]$, by deleting the point x.

Also, for any subset $Y \subseteq A(T)$, and x in $A(T)$ we define $d(x,Y) = d(Y,x) = \text{Infimum}\{d(x,y) \mid y \in Y\}$. $A(T)$ is a metric space with respect to the above distance function.

A subset $Y \subseteq A(T)$ is called a subtree if it is closed and connected. Y is also viewed as a finite (connected) collection of partial edges (closed subintervals), such that the intersection of any pair of distinct partial edges is empty or is a point in V. We call a subtree discrete when all its (relative) boundary points are nodes of T. If Y is a subtree we define the length or size of Y, $L(Y)$, to be the sum of the lengths of its partial edges.

Suppose that each node $v_i \in V$ is associated with a pair of nonnegative weights, (u_i, w_i).

Restricting ourselves to tree networks, and using the above notation we now define the centdian subtree problem. (Note that the node set V is identified as the set of customers in these problems.)

Let L be a positive number, which is smaller than $L(A(T))$, the length of T.

The (u-weighted) *center subtree problem* is to select a subtree $X \subseteq A(T)$, of maximum length L, to minimize the objective $M(X)$, where

$$M(X) = \max_{v_i \in V}\{u_i d(X, v_i)\}. \tag{1}$$

The (w-weighted) *median subtree problem* is to select a subtree $X \subseteq A(T)$, of maximum length L, to minimize the objective $S(X)$, where

$$S(X) = \sum_{v_i \in V} w_i d(X, v_i). \tag{2}$$

With our notation the *centdian subtree problem* on the tree T is to find a subtree $X \subseteq A(T)$, of maximum length L, minimizing the objective

$$C(X) = M(X) + S(X).$$

We note in passing that the discrete version of the above problem, where the selected subtree must be discrete, is NP-hard even for the median problem. See [26] for approximation algorithms for this discrete version.

When $L = 0$, the selected subtree must be a point $x \in A(T)$. A point x_C, minimizing the centdian function, $C(\{x\})$ is called a point centdian of the tree.

In order to solve this problem we will first consider a restricted version, where the selected subtree must contain the root of the tree, v_1. We will then show that the unrestricted problem can be reduced to a restricted version, since there is an optimal subtree containing a point centdian of the tree.

Consider the restricted version, where the subtree must contain v_1. We propose the following formulation.

For each edge e_j of the rooted tree, connecting v_j to its parent, assign a variable x_j: $0 \leqslant x_j \leqslant 1$, $j = 2, \dots, n$. The interpretation of x_j is as follows: Suppose that $x_j > 0$, and let $e_j(x_j)$ be the point on edge e_j, whose distance from $p(v_j)$, the parent of v_j

is $l_j x_j$. Then the only part of e_j, included in the selected subtree rooted at v_1 is the subedge $P[p(v_j), e_j(x_j)]$.

The goal is to minimize the objective function $g(z)$, defined below. Notice that the first part of the objective is the median function and the second is the center function which we denote by z. We use the notation $W_j = \sum_{v_k \in V_j} w_k$, $j = 1, \ldots, n$,

$$g(z) = \min \sum_{j=2}^{n} W_j l_j (1 - x_j) + \underbrace{\max_{v_t \in V - \{v_1\}} \left\{ u_t \sum_{v_k \in P[v_t, v_1)} l_k (1 - x_k) \right\}}_{=z}$$

s.t.

$$\sum_{j=2}^{n} l_j x_j \leqslant L,$$

$$x_j(1 - x_i) = 0 \quad \text{if } v_i = p(v_j),\ v_i \neq v_1,\ j = 2, \ldots, n,$$

$$0 \leqslant x_j \leqslant 1,\ j = 2, \ldots, n. \tag{3}$$

Note that the set of constraints $x_j(1 - x_i) = 0$, if $v_i = p(v_j)$, $j = 2, 3, \ldots, n$, ensures that the set of subedges induced by the solution define a connected subset of the tree.

Therefore, the formulation of the above problem can be written as

$$\min_{z \geqslant 0} g(z),$$

where

$$g(z) = \min \sum_{j=2}^{n} W_j l_j (1 - x_j) + z$$

s.t.

$$\sum_{j=2}^{n} l_j x_j \leqslant L,$$

$$\sum_{v_k \in P[v_t, v_1)} l_k (1 - x_k) \leqslant \frac{z}{u_t}, \quad t = 2, 3, \ldots, n,$$

$$x_j(1 - x_i) = 0 \quad \text{if } v_i = p(v_j),\ v_i \neq v_1,\ j = 2, \ldots, n, \quad (\triangleleft)$$

$$0 \leqslant x_j \leqslant 1,\ j = 2, \ldots, n,\ z \geqslant 0. \tag{4}$$

Proposition 1. *For each z, the constraint $(\triangleleft)$ can be omitted in the above formulation of $g(z)$.*

Proof. Consider the set of optimal solutions to the relaxed version of (4), obtained by removing the constraint $(\triangleleft)$. Each such solution corresponds to a collection of connected components induced by its positive variables. One of those components contains the root v_1. (If no positive variable is associated with an edge incident to v_1, then $\{v_1\}$ will be considered as a connected component.) Consider now an optimal solution X^* such that the sum of the lengths of the minimal (with respect to the descendant partial ordering) components is minimum. A connected component T_1 is minimal, if there is no other connected component, T_2, such that the path from T_2 to v_1 passes through T_1.

Suppose that X^* does not satisfy condition ($\triangleleft$) and consider a minimal component T_1. Let v_i be the closest point to v_1 in T_1 which does not satisfy ($\triangleleft$). Then $x_i^* < 1$, and there is a child of v_i, say v_j, such that $v_i = p(v_j)$ and $x_j^* > 0$. We contradict the minimality of X^* as follows:

Select ε, satisfying, $0 < \varepsilon \leqslant x_j^*$ and $\varepsilon < l_i(1 - x_i^*)/l_j$.

Define a new solution X^{Δ},

$$x_k^{\Delta} = x_k^* \quad \text{if } k \neq i, j,$$

$$x_j^{\Delta} = x_j^* - \varepsilon,$$

$$x_i^{\Delta} = x_i^* + \frac{l_j}{l_i}\varepsilon.$$

The sum of the lengths of the minimal components with respect to X^{Δ}, is smaller than the sum of the lengths of the minimal components with respect to X^*.

We note that the objective value at the solution X^{Δ}, is not larger than its value at the solution X^*, since the difference is

$$W_j l_j(1 - x_j^{\Delta}) + W_i l_i(1 - x_i^{\Delta}) - W_j l_j(1 - x_j^*) - W_i l_i(1 - x_i^*)$$

$$= W_j l_j \varepsilon - W_i l_i \left(\frac{l_j}{l_i}\varepsilon\right) = \underbrace{(W_j - W_i)}_{W_j \leqslant W_i} l_j \varepsilon \leqslant 0.$$

To contradict the minimality of X^*, it is now sufficient to prove that X^{Δ} is feasible in the formulation of (4). The first constraint is

$$\sum_{k=2}^{n} l_k x_k^{\Delta} = \sum_{k=2}^{n} l_k x_k^* - l_j \varepsilon + l_i \left(\frac{l_j}{l_i}\varepsilon\right) = \sum_{k=2}^{n} l_k x_k^* \leqslant L.$$

Let v_t be a node and consider the constraint

$$\sum_{v_k \in P[v_t, v_1)} l_k(1 - x_k^{\Delta}) \leqslant \frac{z}{u_t}.$$

If v_t is not in V_j, the constraint is definitely satisfied. So suppose that $v_t \in V_j$. Then,

$$\sum_{v_k \in P[v_t, v_1)} l_k(1 - x_k^{\Delta}) = \sum_{v_k \in P[v_t, v_1)} l_k(1 - x_k^*) + l_j \varepsilon - l_i \left(\frac{l_j}{l_i}\varepsilon\right)$$

$$= \sum_{v_k \in P[v_t, v_1)} l_k(1 - x_k^*) \leqslant \frac{z}{u_t}. \qquad \square$$

Using the above proposition the problem of finding the optimal centdian subtree rooted at v_1 can be rewritten as

$$\min_{z \geq 0} g(z) = f(z) + z, \tag{5}$$

where $f(z)$ is defined by

$$f(z) = \min \sum_{j=2}^{n} W_j z_j$$

s.t.

$$\sum_{j=2}^{n} z_j \geq l^1,$$

$$\sum_{v_k \in P[v_t, v_1)} z_k \leq \frac{z}{u_t}, \quad t = 2, \ldots, n,$$

$$0 \leq z_j \leq l_j, \ j = 2, \ldots, n$$

with $l^1 = (\sum_{j=2}^{n} l_j - L)$, and $z_j = l_j(1 - x_j)$. $f(z)$ is a piecewise linear nonincreasing convex function of z. Additional properties of this function are discussed in the following sections.

3. Greedy algorithm for computing $f(z)$ for a given z

We now describe a greedy algorithm to compute $f(z)$. For $t=2, \ldots, n$, define $b_t = z/u_t$.

For each iteration k, we define A_k to be the set of active variables. Initially, for the first iteration, $k = 1$, define $A_1 = \{z_2, \ldots, z_n\}$. Let L_k be the current value of the sum of the active variables i.e., $L_k = \sum_{z_j \in A_k} z_j$. (For the first iteration, $L_1 = l^1$.)

If the current set of active variables is empty, stop. The problem defining $f(z)$ is infeasible. Otherwise, select an active variable say z_t, such that $W_t \leq W_i$, for each active variable z_i. (Initially the variable selected is the one associated with the lightest leaf.)

Assign to z_t the largest possible value satisfying

$$z_t \leq \min\{l_t, b_t, L_k\}.$$

From the above $z_t^* = \min\{l_t, b_t, L_k\}$.

(a) If $z_t^* < l_t$, set $z_i = 0$ for all variables z_i, such that v_i is on $P(v_t, v_1)$. Define A_{k+1} to be the set obtained by removing z_t, and all variables z_i, which are set to 0 from A_k. Set $L_{k+1} = L_k - z_t^*$.

If $L_{k+1} = 0$, stop, and set all remaining active variables to zero. Otherwise, repeat.

(b) If $z_t^* = l_t$, and $v_i \neq v_1$ is the parent of v_t, i.e., $v_i = p(v_t)$, replace the constraint

$$\sum_{v_k \in P[v_i, v_1)} z_k \leq b_i \text{ by } \sum_{v_k \in P[v_i, v_1)} z_k \leq \min\{b_i, b_t - l_t\},$$

(substitute $\min\{b_i, b_t - l_t\}$ for b_i).

Remove z_t^* from A_k to obtain A_{k+1}, and set $L_{k+1} = L_k - z_t^*$. If $L_{k+1} = 0$, stop, and set all remaining active variables to zero. Otherwise, repeat.

It is easy to observe that if the set $\{W_2, \ldots, W_n\}$ is presorted, then for any given value of z, it takes $O(n)$ time to compute $f(z)$.

3.1. Validity of the greedy algorithm

The validity of the greedy algorithm presented above will follow directly from the next result, by an inductive argument on the number of nodes in the tree.

Proposition 2. *Given is a value $z > 0$. Let v_t be a leaf of the tree such that $W_t \leqslant W_i$, $i = 1, 2, \ldots, n$, then there is an optimal solution $(z_2^*, z_3^*, \ldots, z_n^*)$, (resolving $f(z)$), such that $z_t^* = \min\{l_t, l^1, z/u_t\} = \varepsilon$.*

Proof. Consider an optimal solution, $(z_2^*, z_3^*, \ldots, z_n^*)$, such that z_t^* is as large as possible. Suppose $z_t^* < \varepsilon$. There is another variable z_s^*, $s \neq t$, such that $z_s^* > 0$, since $\sum_{k=2}^n z_k^* = l^1$.

We now select a particular positive variable z_i^* as follows.

Case I: If there is $z_s^* > 0, s \neq t$, and $v_s \in P[v_t, v_1)$, select $i = s$.

Case II: If $z_j^* = 0$, for all $v_j \in P[v_t, v_1)$, $j \neq t$, select $i = s$, where $z_s^* > 0$.

Define a new solution $z^\wedge$ by setting

$$z_k^\wedge = z_k^*, \quad k \neq i, t,$$

$$z_t^\wedge = z_t^* + \delta, \quad 0 < \delta \leqslant \min\{z_i^*, \varepsilon - z_t^*\},$$

$$z_i^\wedge = z_i^* - \delta.$$

The objective value at $z^\wedge$ is less than or equal to $f(z^*)$, since

$$\sum_{j=2}^n W_j z_j^\wedge = \sum_{j=2}^n W_j z_j^* + \underbrace{\delta W_t - \delta W_i}_{\leqslant 0} \leqslant \sum_{j=2}^n W_j z_j^* \quad (v_t \text{ is a leaf } \Rightarrow W_t = w_t).$$

We now contradict the maximality of z_t^* by showing that $z^\wedge$ is feasible. Indeed,

$$0 \leqslant z_j^\wedge \leqslant l_j, \quad j = 2, 3, \ldots, n,$$

$$\forall v_j,\ j \neq t, \quad \sum_{v_k \in P[v_j, v_1)} z_k^\wedge \leqslant \sum_{v_k \in P[v_j, v_1)} z_k^* \leqslant \frac{z}{u_j}.$$

For $j = t$:

Case I:

$$\sum_{v_k \in P[v_t, v_1)} z_k^\wedge = \sum_{v_k \in P[v_t, v_1)} z_k^* + \delta - \delta = \sum_{v_k \in P[v_t, v_1)} z_k^* \leqslant \frac{z}{u_t}.$$

Case II:

$$\sum_{v_k \in P[v_t, v_1)} z_k^\wedge = z_t^\wedge = z_t^* + \delta \leqslant \frac{z}{u_t}. \qquad \square$$

4. Finding an optimal centdian subtree

First, we present an $O(n \log n)$ algorithm for finding an optimal centdian subtree of a given length, which is restricted to be rooted at some specified point of the tree. We then prove that there is an optimal (unrestricted) subtree, which contains a point centdian of the tree. Thus, we can find an optimal centdian subtree, by applying the algorithm for the rooted version, by declaring a point centdian to be the root.

We start by identifying a set of polynomial size which contains all the breakpoints of the function $f(z)$ defined in (5).

Theorem 1. *The function $f(z)$, defined in (5), is convex, monotone nonincreasing and piecewise linear. Define the set*

$$R = R_1 \cup R_2,$$

where

$$R_1 = \{u_i(d(v_i, v_1) - d(v_j, v_1)) \mid v_i, v_j \in V,\ v_i \in V_j\},$$
$$R_2 = \{(d(v_i, v_1) - d(v_j, v_1))/(1/u_i - 1/u_j) \mid v_i, v_j \in V\}.$$

Let z' and z'' be two consecutive elements (real numbers) in R. Then $f(z)$ has at most $n-1$ breakpoints in the interval connecting z' and z''.

Proof. $f(z)$ is defined as the solution value of a minimization parametric linear programming problem, where the parameter z appears only on the right-hand side vector of the constraint set. Therefore, the function $f(z)$ is convex, monotone nonincreasing and piecewise linear.

For each node $v_i \in V$, if $d(v_i, v_1) \geqslant z/u_i$, define $v_i(z)$ to be the point on $P[v_i, v_1)$ whose distance from v_i is z/u_i. Otherwise, define $v_i(z) = v_1$.

Let $T(z)$ be the minimal subtree containing the points $v_i(z)$, $i = 1, \ldots, n$.

From the second set of constraints of the program (5) defining $f(z)$, it is clear that the optimal subtree yielding $f(z)$ must contain $T(z)$. Moreover, if $T(z)$ is feasible, i.e., its length is at most L, from the above discussion the optimal subtree is obtained by greedily expanding $T(z)$, following the maximum descent direction.

Consider a value of z, such that each leaf of the subtree $T(z)$ is in the interior of an edge. Suppose that $v_i(z)$ is a leaf of $T(z)$, and it is an interior point of an edge (v_j, v_k), where v_j is a child of v_k. Then we have

$$d(v_j, v_i(z)) = \min\{z/u_t - d(v_t, v_j) \mid v_t \in V_j\}.$$

The distance $d(v_j, v_i(z))$ is a piecewise linear function of z. Each piece is determined by a different descendant of v_j. Therefore, when the parameter z varies within a given piece, the length of the subedge $(v_j, v_i(z))$ varies linearly. Consider a linear decrease in the value of z, starting with an interior point of a piece of the function $d(v_j, v_i(z))$. Suppose that the piece corresponds to the descendant v_t. There are possibly two stopping rules for determining the breakpoint. First, there is the value of z, such that $v_i(z) = v_j$. In this case z is determined by the equation $d(v_j, v_i(z)) = z/u_t - d(v_t, v_j) = 0$.

Since v_t is a descendant of v_j, the breakpoint is given by

$$z = u_t d(v_t, v_j) = u_t(d(v_t, v_1) - d(v_j, v_1)).$$

Thus, the breakpoint is in the set R_1 defined above.

In the second stopping rule we decrease z till we reach a breakpoint of the function $d(v_j, v_i(z))$. In this case there are two descendants of v_j, say v_t and v_s, such that $z/u_t - d(v_t, v_j) = z/u_s - d(v_s, v_j)$.

Since v_t and v_s are descendants of v_j, the breakpoint is given by

$$z = (d(v_t, v_1) - d(v_s, v_1))/(1/u_t - 1/u_s).$$

Thus, the breakpoint is in the set R_2 defined above.

Next consider a pair of consecutive reals z' and z'' in R with $z' < z''$. From the above discussion it follows that $L(T(z))$, the length of $T(z)$ is a decreasing linear function of z in the interval $[z', z'']$. Moreover, in this interval each leaf of $T(z)$ varies linearly within the interior of some edge of the tree. In particular, the edges containing the leaves of $T(z)$ are independent of z. To compute $f(z)$ for some z in $[z', z'']$, we greedily expand $T(z)$, following the maximum descent direction, until the expanded subtree reaches a total length L. Since the edges containing the leaves of $T(z)$ are fixed for all values in $[z', z'']$, it follows that the maximum descent direction is also independent of z. Let the descent direction be represented by the respective sequence of edges, $(e_{k(1)}, e_{k(2)}, \dots, e_{k(t)})$. $T(z)$ is first expanded along $e_{k(1)}$, then, (if the length is still smaller than L), along $e_{k(2)}$, etc. Note that $e_{k(1)}$ contains a leaf of $T(z)$. For each $j = 1, \dots, t$, if $e_{k(j)}$ contains a leaf of $T(z)$, say $v_i(z)$, define $l'_{k(j)}(z)$ to be the distance from $v_i(z)$ to the node of $e_{k(j)}$ which is not in $T(z)$. Otherwise, define $l'_{k(j)}(z) = l_{k(j)}$.

It is now clear from the greedy algorithm that the only possible breakpoints of the function $f(z)$ in the interval $[z', z'']$, are the solutions to any of the following t linear equations:

$$L(T(z)) + \sum_{j=1}^{p} l'_{k(j)}(z) = L, \quad p = 1, \dots, t. \qquad \square$$

It is worth noting that in the unweighted case, i.e. when $u_j = 1$ for all $j = 1, \dots, n$, the set R defined in the theorem above reduces to R_1^-, defined as follows. For each node v_j let $v_{t(j)}$ be a leaf in V_j, which is furthest away from v_j. Then from the proof of the above theorem it follows that z^* is an element in $R_1^- = \{d(v_j, v_{t(j)}) \mid v_j \in V\}$. It is easy to see that the total effort to construct R_1^- is $O(n)$.

From (5) and since Theorem 1 proves that f is piecewise linear and convex, it follows that in order to identify an optimal centdian subtree, it is sufficient to find a breakpoint z^* of the function $f(z)$ such that the directional derivatives $f^+(z)$ and $f^-(z)$ satisfy

$$f^-(z^*) + 1 \leqslant 0 \leqslant f^+(z^*) + 1.$$

Using Theorem 1, we will first identify a pair of consecutive elements in R, which bound z^*. We show how to search for such a pair efficiently in $O(n \log n)$ time, without explicitly generating all the elements in R. (Note that $|R| = O(n^2)$.)

To apply the search we need an efficient procedure to determine for a given value of the parameter z, whether $z = z^*$, $z < z^*$, or $z > z^*$. This can be done by computing $f^+(z)$, and $f^-(z)$. Resolving this question for a given z will be called testing z. We will show that testing takes linear time.

For the sake of brevity, and to simplify the presentation, we assume that we have rational data. Specifically, we assume that each edge length l_j, $j = 2, \ldots, n$, and each weight u_j, $j = 1, \ldots, n$, are rational numbers, where all numerators and denominators are integers bounded above by M. Then, it is easy to see that each element in the set R, defined above, is a rational number where the integer denominator is bounded above by M^{n+1}. Therefore, if z' and z'' are two distinct elements of R, the distance between them is greater than or equal to $1/M^{2n+2}$.

It now follows that in order to compute $f^+(z)$, $f^-(z)$ for a given element $z \in R$, it is sufficient to compute $f(z)$, $f(z + \varepsilon)$, and $f(z - \varepsilon)$, for some $0 < \varepsilon < 1/M^{2n+2}$.

Suppose that the set $\{W_2, \ldots, W_n\}$ is already sorted. Then, testing can be done in $O(n)$ time by the greedy algorithm described above.

Consider first the set R_1. Our task is to identify two consecutive elements in R_1, bracketing z^*. Suppose that the nodes in V are relabelled such that $0 = d(v_1, v_1) \leqslant d(v_2, v_1) \leqslant \cdots \leqslant d(v_n, v_1)$, and define the vector $a = (d(v_1, v_1), d(v_2, v_1), \ldots, d(v_n, v_1))$. Also, let $e = (1, 1, \ldots, 1)$.

For each $i = 1, \ldots, n$, define

$$R_{1,i} = u_i d(v_i, v_1)e - u_i a.$$

Then the set R_1 is a subset of the set $R_1^+ = R_{1,1} \cup R_{1,2} \cup \cdots \cup R_{1,n}$. Note that the components of each vector $R_{1,i}$, $i = 1, \ldots, n$, are already sorted. Therefore, for each k, the kth largest element of $R_{1,i}$ can be computed in constant time.

With the above properties and ingredients we can now directly apply the search procedures over monotone matrices (see [15,6,7]). With this approach it will take $O(n \log n)$ time to find the two consecutive elements of R_1^+ which bracket an optimal solution z^*. If one of the two elements is z^* we stop.

Otherwise we turn to the search over the set R_2. Define the vector $b' = (1/u_1, \ldots, 1/u_n)$. Let $a = (a_1, \ldots, a_n)$ be the vector of distances of the nodes in V from v_1, defined above. Then the set R_2 can be represented as

$$R_2 = \{(a_i - a_j)/(b'_i - b'_j) \mid i, j = 1, \ldots, n\}.$$

With the above linear time testing of an element z, we can now directly apply the search procedure in Megiddo and Tamir (1983) [14], and obtain the two consecutive elements of R_2 which bracket z^* in $O(n \log^2 n)$ time. The running time can be improved to $O(n \log n)$ if we apply the modification and improvements described in Cole (1987) [4]. (See Application (8), p. 206 in [4], where Cole explains how to improve the $O(n \log^2 n)$ search in the appendix of [14] to $O(n \log n)$.)

To summarize, at the end of this phase, we have a pair of consecutive elements of R, say z' and z'', bracketing z^*. Next, we apply the result in the above theorem. Specifically, we use the representation of the O(n) breakpoints of $f(z)$ in $[z', z'']$, (see the proof of the theorem). These breakpoints can be computed in O(n) time, and z^*, which is one of them, will be identified by a binary search in O($n \log n$) time.

To conclude, we have shown that an optimal rooted centdian subtree of a tree can be found in O($n \log n$) time.

It should also be noticed that for the unweighted case we can obtain a simpler O($n \log n$) algorithm using binary search in the set R_1^- defined above.

In the following theorem, we prove that there exists an optimal centdian subtree which contains an optimal centdian point of T. Thus, the problem of finding an optimal not necessarily rooted subtree of given length can be solved by finding the optimal subtree rooted at an optimal centdian point. Since the complexity of finding an optimal centdian point in a tree is O(n), using for example the algorithm of Tamir et al. [26], the overall complexity of the problem does not increase.

Let x_C be a centdian point of T. Recall that from (1) and (2) we denote by $S(T')$ and $M(T')$ the functions sum and maximum of the weighted distances from a subtree T' to the nodes of T. This is,

$$S(T') = \sum_{v_i \in V} w_i d(T', v_i), \tag{6}$$

$$M(T') = \max_{v_i \in V} u_i d(T', v_i). \tag{7}$$

Now, we let

$$C(T') = S(T') + M(T'). \tag{8}$$

It is straightforward that when the subtree T' reduces to a singleton $\{x\}$, the centdian function of x is $C(\{x\})$.

Theorem 2. *Let x_C be a point centdian of T. Then for each length L there is an optimal centdian subtree of length L containing x_C.*

Proof. Let $T(x)$ be an optimal centdian subtree of length L, closest to x_C, where x is the closest point to x_C in $T(x)$. Suppose that $x \neq x_C$. We will contradict this supposition by showing that there is an optimal centdian subtree which is closer than $T(x)$ to x_C.

Let us denote by $P[x_C, x]$ the unique path on T from x_C to x. For any $0 < \delta < d(x, x_C)$, let $x(\delta)$ be the point in $P[x_C, x]$ at a distance δ from x.

Select a positive and sufficiently small Δ, such that no interior point of $P[x, x(\Delta)]$ is a node or a local center. (A point $z \in A(T)$ is called a local center if it is an interior point of some edge, say $e_k = (v_k, p(v_k))$, and there are nodes, say v_i, v_j such that $u_i d(z, v_i) = u_j d(z, v_j)$, and $u_i d(v_k, v_i) \neq u_j d(v_k, v_j)$.)

Let $V^+(x)$ be the set of all nodes v_i such that $x(\Delta) \in P[v_i, x]$, and let $V^-(x) = V \setminus V^+(x)$.

Let v_k be a node such that the maximum weighted distance from any point in $P[x, x(\Delta)]$ to the nodes in $V^-(x)$ is attained at v_k. Define y to be the closest point to v_k in $T(x)$.

Let $y' \neq x$, be a leaf of $T(x)$, such that $y \in P[x, y']$, and let y'' be the closest node to y' in $P[x, y']$. (If there is no such node define $y'' = x$.) Suppose that $\Delta < d(y', y'')$. For any $0 < \delta \leqslant \Delta$, let $y'(\delta)$ be the point on $P[y', y'']$ whose distance from y' is δ. Define $T(x(\delta))$ to be the subtree obtained from $T(x)$ by augmenting $P[x(\delta), x]$ and deleting the half open path $P(y'(\delta)), y']$. $T(x(\Delta))$ will be called the perturbed subtree.

We will contradict the existence of $T(x)$, by showing that the perturbed subtree is also an optimal centdian subtree.

First we note that $T(x(\Delta))$ is also of length L.

We observe that

$$d(v_i, x(\Delta)) - d(v_i, x) = \Delta \geqslant d(v_i, T(x(\Delta))) - d(v_i, T(x)) \quad \text{for any } v_i \in V^-(x). \tag{9}$$

Thus,

$$\begin{aligned}
S(x) - S(x(\Delta)) &= \sum_{v_i \in V^+(x)} w_i(d(x, v_i) - d(x(\Delta), v_i)) \\
&\quad + \sum_{v_i \in V^-(x)} w_i(d(x, v_i) - d(x(\Delta), v_i)) \\
&= \sum_{v_i \in V^+(x)} w_i(d(T(x), v_i) - d(T(x(\Delta)), v_i)) \\
&\quad + \sum_{v_i \in V^-(x)} w_i(d(x, v_i) - d(x(\Delta), v_i)) \\
(\text{by } (9)) &\leqslant \sum_{v_i \in V^+(x)} w_i(d(T(x), v_i) - d(T(x(\Delta)), v_i)) \\
&\quad + \sum_{v_i \in V^-(x)} w_i(d(T(x), v_i) - d(T(x(\Delta)), v_i)) \\
&= S(T(x)) - S(T(x(\Delta))).
\end{aligned}$$

Consider first the case where $M(x) \geqslant M(x(\Delta))$.

In this case the maximal weighted distances from both x and $x(\Delta)$ are attained at some nodes in $V^+(x)$. This is also true for the weighted distances from $T(x)$ and

$T(x(\Delta))$. Thus,

$$M(x) - M(x(\Delta)) = M(T(x)) - M(T(x(\Delta))).$$

Therefore (since $C(x_C) \leqslant C(x(\Delta)) \leqslant C(x)$),

$$0 \leqslant C(x) - C(x(\Delta)) \leqslant C(T(x)) - C(T(x(\Delta))),$$

and $T(x(\Delta))$ is also an optimal centdian subtree.

Suppose now that $M(x) < M(x(\Delta))$.

It is sufficient to prove that

$$M(x(\Delta)) - M(x) \geqslant M(T(x(\Delta))) - M(T(x)).$$

Since there are no local centers in $P(x, x_C)$, both $M(x)$ and $M(x(\Delta))$ are attained at the node v_k in $V^-(x)$, defined above, i.e., $M(x) = u_k d(v_k, x)$, and $M(x(\Delta)) = u_k d(v_k, x(\Delta))$.

Therefore,

$$M(x(\Delta)) - M(x) = \Delta u_k.$$

If $M(T(x(\Delta))) = u_i d(v_i, T(x(\Delta)))$ for some node v_i in $V^+(x)$, i.e., $M(T(x(\Delta)))$ is attained at $V^+(x)$, then it is easy to see that for each node v_j in $V^-(x)$,

$$u_j d(T(x), v_j) \leqslant u_j d(T(x(\Delta)), v_j) \leqslant u_i d(T(x(\Delta)), v_i) < u_i d(T(x), v_i) \leqslant M(T(x)).$$

Thus, $M(T(x))$ is also attained at $V^+(x)$. In such a case $M(T(x(\Delta))) - M(T(x)) < 0$, and therefore

$$M(x(\Delta)) - M(x) \geqslant M(T(x(\Delta))) - M(T(x)).$$

Consider the case where $M(T(x(\Delta)))$ is attained at $V^-(x)$ only.

If $M(T(x))$ is attained at $V^+(x)$, but not at $V^-(x)$, then there exists $0 < \delta < \Delta$, such that $M(T(x(\delta)))$ is attained at $V^+(x)$. From the above discussion we will get the contradiction that $T(x(\delta))$ is an optimal centdian subtree.

To conclude the proof, it is now sufficient to consider the last case where $M(x)$, $M(x(\Delta))$, $M(T(x))$ and $M(T(x(\Delta)))$ are all attained at $V^-(x)$ only. Recall that y is the closest point to v_k in $T(x)$. If $y = x$, then it is easy to see that $M(T(x)) = M(x) = u_k d(x, v_k)$. From the definition of v_k, for each node v_i such that y' is in $P[x, v_i]$, we have

$$u_k d(x, v_k) \geqslant u_i d(x, v_i) = u_i d(y, v_i) = u_i[d(y, y') + d(y', v_i)] > u_i d(y', v_i).$$

The last inequality follows from the fact that $y \neq y'$, when $x = y$. This strict inequality implies that $M(T(x(\Delta))) = u_k d(x, v_k)$, if Δ is sufficiently small. Thus, when $x = y$ we have $M(T(x(\Delta))) - M(T(x)) = 0$, and the result follows.

Thus, suppose that $x \neq y$. Again, if $M(T(x(\Delta))) - M(T(x)) = 0$, the result holds. Suppose otherwise. Then there is a node v_j such that y' is in $P(v_j, x)$, and $M(T(x)) = u_j d(y', v_j)$, $M(T(x(\Delta))) - M(T(x)) = \Delta u_j$.

We need to prove that $u_k \geqslant u_j$. Suppose, on the contrary, that $u_j > u_k$. From the definition $M(x) = u_k d(x, v_k) \geqslant u_j d(x, v_j)$. Thus, $u_k[d(x, y) + d(y, v_k)] \geqslant u_j[d(x, y) + d(y, v_j)]$.

Therefore $u_k d(y, v_k) \geqslant u_j d(y, v_j) + [u_j - u_k] d(x, y) > u_j d(y, v_j)$. (The strict inequality follows from $x \neq y$ and $u_j > u_k$.) The inequality $u_k d(y, v_k) > u_j d(y, v_j)$ implies the following contradiction:

$$M(T(x)) \geqslant u_k d(y, v_k) > u_j d(y, v_j) \geqslant u_j d(y', v_j) = M(T(x)). \qquad \square$$

5. Conclusion

Algorithms which run in linear time for the location of tree shaped facilities in tree networks are known for the unweighted center and the weighted median criteria (see [23,24] for further details).

In this paper, we have presented a new algorithm which runs in $O(n \log n)$ to solve the location of a subtree of given length in a tree network using the centdian objective function. The centdian function is a convex combination of the weighted median and the weighted center objective functions. Therefore, our approach permits, in addition, the resolution of the weighted center case and it also unifies the developments in [23,24].

References

[1] I. Averbakh, O. Berman, Algorithms for path medi-centers of a tree, Comput. Oper. Res. 26 (1999) 1395–1409.

[2] O. Berman, E.K. Yang, Medi-centre location problems, J. Oper. Res. Soc. 42 (1991) 313–322.

[3] E.J. Carrizosa, E. Conde, F.R. Fernández, J. Puerto, An axiomatic approach to the cent-dian criterion, Location Sci. 3 (1994) 1–7.

[4] R. Cole, Slowing down sorting networks to obtain faster sorting algorithms, J. ACM 34 (1987) 200–208.

[5] P. Fernández, B. Pelegrín, J. Fernández, Location of paths on trees with minimal eccentricity and superior section, TOP 6 (1998) 223–246.

[6] G.N. Frederickson, D.B. Johnson, Finding kth paths and p-centers by generating and searching good data structures, J. Algorithms 4 (1983) 61–80.

[7] G.N. Frederickson, D.B. Johnson, Generalized selection and ranking: sorted matrices, SIAM J. Comput. 13 (1984) 14–30.

[8] S.L. Hakimi, E.F. Schmeichel, M. Labbe, On locating path- or tree shaped facilities on networks, Networks 23 (1993) 543–555.

[9] J. Halpern, The location of a cent-dian convex combination on an undirected tree, J. Regional Sci. 16 (1976) 237–245.

[10] J. Halpern, Finding minimal center-median convex combination (cent-dian) of a graph, Management Sci. 24 (1978) 535–544.

[11] G.Y. Handler, Medi-centers of a tree, Transport. Sci. 19 (1985) 246–260.

[12] T.U. Kim, T.J. Lowe, A. Tamir, J.E. Ward, On the location of a tree-shaped facility, Networks 28 (1996) 167–175.

[13] M.C. López De los Mozos, J.A. Mesa Colmenar, Location of cent-dian paths in tree graphs, Proceedings of the VI Meeting of the Euro Working Group on Locational Analysis, 1992, pp. 135–145.

[14] N. Megiddo, A. Tamir, New results on the complexity of p-center problems, SIAM J. Comput. 12 (1983) 751–758.

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

[16] E. Minieka, The optimal location of a path or tree in a tree network, Networks 15 (1985) 309–321.

[17] E. Minieka, N.H. Patel, On finding the core of a tree with a specified length, J. Algorithms 4 (1983) 345–352.

[18] B. Pelegrín, P. Fernández, On the sum-max bicriterion path problem, Comput. Oper. Res. 25 (1998) 1043–1054.

[19] S. Peng, W. Lo, Efficient algorithms for finding a core of a tree with specified length, J. Algorithms 20 (1996) 445–458.

[20] S. Peng, A.B. Stephens, Y. Yesha, Algorithms for a core and k-tree core of a tree, J. Algorithms 15 (1993) 143–159.

[21] D. Pérez-Brito, J.A. Moreno-Pérez, I. Rodríguez-Martín, The finite dominating set for the p-facility centdian network location problem, Stud. Location Anal. 11 (1997) 27–40.

[22] D. Pérez-Brito, J.A. Moreno-Pérez, I. Rodríguez-Martín, The 2-facility centdian network problem, Location Sci. 6 (1998) 369–381.

[23] A. Shioura, M. Shigeno, The tree center problems and the relationship with the bottleneck knapsack problems, Networks 29 (1997) 107–110.

[24] A. Tamir, Fully polynomial approximation schemes for locating a tree-shaped facility: a generalization of the knapsack problem, Discrete Appl. Math. 87 (1998) 229–243.

[25] A. Tamir, T.J. Lowe, The generalized p-forest problem on a tree network, Networks 22 (1992) 217–230.

[26] A. Tamir, D. Pérez-Brito, J.A. Moreno-Pérez, A polynomial algorithm for the p-centdian problem on a tree, Networks 25 (1998) 255–262.