

The Bi-Criteria Doubly Weighted Center-Median Path Problem On A Tree

J. Puerto¹, A.M. Rodríguez-Chía², A. Tamir³, & D. Pérez-Brito⁴

¹ Facultad de Matemáticas. Universidad de Sevilla.

² Facultad de Ciencias. Universidad de Cádiz.

³ School of Mathematical Sciences. Tel Aviv University.

⁴ ETS de Ingeniería Informática. Universidad de La Laguna.

January 5, 2006

Abstract

Given a tree network T with n nodes, let \mathcal{P}_L be the subset of all discrete paths whose length is bounded above by a prespecified value L . We consider the location of a path-shaped facility $P \in \mathcal{P}_L$, where customers are represented by the nodes of the tree. We use a bi-criteria model to represent the total transportation cost of the customers to the facility. Each node is associated with a pair of nonnegative weights: the center-weight and the median-weight. In this doubly weighted model, a path P is assigned a pair of values $(MAX(P), SUM(P))$, which are, respectively, the maximum center-weighted distance and the sum of the median-weighted distances from P to the nodes of the tree. Viewing \mathcal{P}_L and the planar set $\{(MAX(P), SUM(P)) : P \in \mathcal{P}_L\}$ as the decision space and the bi-criteria or outcome space respectively, we focus on finding all the nondominated points of the bi-criteria space. We prove that there are at most $2n$ nondominated outcomes, even though the total number of efficient paths can be $\Omega(n^2)$, and they can all be generated in $O(n \log n)$ optimal time. We apply this result to solve the cent-dian model, whose objective is a convex combination of the weighted center and weighted median functions. We also solve the restricted models, where the

goal is to minimize one of the two functions MAX or SUM , subject to an upper bound on the other one, both with and without a constraint on the length of the path. All these problems are solved in linear time, once the set of nondominated outcomes has been obtained, which in turn results in an overall complexity of $O(n \log n)$. The latter bounds improve upon the best known results by a factor of $O(\log n)$.

Keywords: Location on trees, Bi-criteria location, Center-median paths, Length constrained paths.

1 Introduction

In a typical location problem there is a set of demand points embedded in some metric space and the objective is to locate a specified number of servers optimizing some criterion, which usually depends on the distances between the demand points and their respective servers. Traditionally most papers focus on location problems where a server (facility) is representable by a point in the metric space. However, in recent years there has been a growing interest in studying the location of connected structures, which can not be represented by isolated points in the space. These problems were motivated by concrete decision problems related to routing and network design (for instance, extending or adding lines to an existing road network). These lines can be viewed as new facilities. Studies on location of connected structures (called *extensive* facilities) already appeared in the early eighties [8, 13, 19, 20, 21, 29]. The most common extensive facilities are paths and subtrees of the underlying network. The reader is referred to [12] where the authors focus on the complexity of solving many versions of location problems involving extensive facilities. The need to study more complicated location models and use them to solve real scenarios has also led to the introduction of multicriteria optimization approaches and methods into Location Analysis. (See [22] for an overview of the different models and solution procedures used in multicriteria location problems.) However, almost all studies on multicriteria location models consider only point facilities. To the best of our knowledge there are only very few papers that apply multicriteria analysis to extensive facility location models.

In this paper we concentrate on certain bi-criteria cost minimization problems involving the location of a discrete path facility (a line) on a tree network with n nodes, where these nodes represent the customers. (By a discrete path we mean a path whose endpoints are nodes of the underlying tree, and not interior points of some edges.) We note that in general, when the objective function can be computed in polynomial time for a given path or is given by some oracle model, single discrete path location problems are solvable in polynomial time since there are a quadratic number of different paths in the tree. Thus, for such models the goal is to find low complexity algorithms. In view of the fact that there are a quadratic number of paths in a tree, the major question is whether there are subquadratic algorithms for path location problems. This is still a most challenging open problem for several general path location models on trees. (See the discussion in the last section.) For some important models subquadratic complexity has already been achieved. For example, subquadratic algorithms are known for the weighted path median problem, where the objective is the sum of median-weighted distances, and the weighted path center problem where the objective is the maximum center-weighted distance from the path to the nodes of the considered tree. In both models the selected path is also restricted by a prespecified length constraint. (See e.g., [1, 2, 6, 7, 9, 13, 15, 19, 20, 21, 23, 24, 27, 28, 29, 30, 32, 33, 34, 35, 36], and the references therein.) A model where the objective is to minimize the variance of the distances to the path is considered in [10]. The authors present an $O(n^2 \log n)$ algorithm to solve this model.

Averbakh and Berman [3] consider a bi-criteria path location problem without the length constraint. Specifically, the two criteria that they study are the unweighted center and the weighted median objectives. A linear time algorithm was developed for the following three problems: 1) the minimization of a convex combination of the above pair of criteria (the cent-dian problem), 2) the minimization of the weighted median criterion subject to an upper bound on the maximum unweighted distance from the selected path, and 3) the minimization of the unweighted maximum distance subject to an upper bound on the median objective. Becker et al. [4, 5] study the generalization of the last two problems to the case where the length of the selected path cannot exceed a prespecified value L . They solve this pair of constrained minimization problems in

	Center Unweighted Models	Doubly Weighted Models ¹
Problem	Complexity	Complexity
Bicriteria (MAX(P), SUM(P))	$O(n \log n)$ [3]	$\Theta(n \log n)$
$\min_{s.t.} \begin{array}{l} SUM(P) \\ MAX(P) \leq \alpha \end{array}$	$O(n)$ [3]	$O(n \log n)$
$\min_{s.t.} \begin{array}{l} MAX(P) \\ SUM(P) \leq \alpha \end{array}$	$O(n)$ [3]	$O(n \log n)$
$\min_{s.t.} \begin{array}{l} SUM(P) \\ MAX(P) \leq \alpha \\ L(P) \leq L \end{array}$	$O(n \log^2 n)$ [5]	$\Theta(n \log n)$
$\min_{s.t.} \begin{array}{l} MAX(P) \\ SUM(P) \leq \alpha \\ L(P) \leq L \end{array}$	$O(n \log^2 n)$ [5]	$O(n \log n)$

Table 1: Summary of results.

$O(n \log^2 n)$ time.

In this paper we extend, generalize and improve upon the results in [3, 4, 5]. We also consider tree graphs only, and use a bi-criteria model to represent the total transportation cost of the customers (nodes) to the facility (path). Given a tree network T with n nodes, let \mathcal{P}_L be the subset of all discrete simple paths whose length is bounded above by a prespecified value L . We consider the location of a path-shaped facility $P \in \mathcal{P}_L$. Each node is associated with a pair of nonnegative weights: the center-weight and the median-weight. In this doubly weighted model, each path $P \in \mathcal{P}_L$ is assigned a pair of values $(MAX(P), SUM(P))$, which are, respectively, the maximum center-weighted distance and the sum of the median-weighted distances from P to the nodes of the tree.

Viewing \mathcal{P}_L and the planar set $\{(MAX(P), SUM(P)) : P \in \mathcal{P}_L\}$ as the decision space and the bi-criteria or outcome space respectively, we focus on finding all the nondominated points of the bi-criteria space. We note that in the models in [3, 4,

¹Boldfaced results are new results in the paper.

5], all the center weights are identical. (This uniformity assumption simplifies the combinatorial complexity of the model significantly. For example, the unweighted path center with a length constraint is solved in $O(n)$ time in [35], while the best algorithm known for the weighted version has $O(n \log n)$ complexity [33, Section 3.2].) Studying only this unweighted model without a length constraint on the selected path, Averbakh and Berman [3] proved that there are at most $n - 1$ nondominated outcomes, and gave an $O(n \log n)$ algorithm to generate all of them.

We consider the doubly weighted model with a length constraint, and prove that there are at most $2n$ nondominated points in the bi-criteria space. (Note that the total number of efficient paths can actually be $\Omega(n^2)$; see [3].) We then show how to generate all these nondominated points in $O(n \log n)$ optimal time. We apply this result to solve the cent-dian model, whose objective is a convex combination of the weighted center and weighted median functions. We also solve the restricted models, where the goal is to minimize one of the two functions MAX or SUM , subject to an upper bound on the other one, both with and without a constraint on the length of the path. All these problems are solved in linear time, once the set of nondominated outcomes has been obtained, which in turn results in an overall complexity of $O(n \log n)$. The latter bounds improve upon the best known results by a factor of $O(\log n)$. (See Table 1 for a comparison of results; new results are indicated in boldface in the table.)

In the next section we formally introduce the necessary notation and define the bi-criteria path location problems. In Section 3, we prove that there are at most $2n$ nondominated outcomes, and present an $O(n \log n)$ algorithm to generate the set of different values that the function $MAX(P)$ can take on. In Sections 4-5 we provide $O(n \log n)$ algorithms to compute the set of nondominated outcomes. Section 4 is devoted to the unconstrained case, where there is no length constraint. The algorithm for this case is much easier. It also yields a linear time procedure for solving the path cent-dian problem when the center objective is unweighted (matching the complexity in [3]). Section 5 deals with the case where there is a length constraint on the selected path. More sophisticated tools are needed to generate the set of nondominated outcomes with the same complexity. In the last section we discuss some challenging open problems for path location on trees.

2 Notation and problem definitions

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 $\mathcal{A}(T)$ denote the continuum set of points on the edges of T . In particular, each subgraph of T is also viewed as a subset of $\mathcal{A}(T)$. We refer to an interior point on an edge by its distance along the edge to the nodes of the edge.

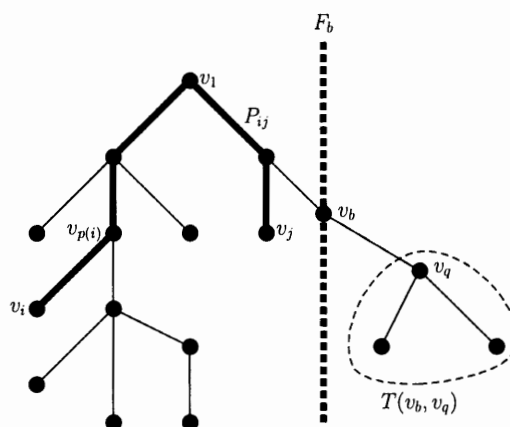


Figure 1: Notation for a tree rooted at v_1 .

Let P_{ij} denote the unique simple path in $\mathcal{A}(T)$ connecting v_i and v_j . Suppose that the tree T is rooted at some specified node, say v_1 . For each node v_j , $j = 2, 3, \dots, n$, let $v_{p(j)}$, the *parent* of v_j , be the node $v \in V$ closest to v_j , $v \neq v_j$, on P_{1j} . Thus v_j is a *child* of $v_{p(j)}$. Let $S(v_j)$ denote the set of children of v_j . Node v_j is a *leaf* if it has no children. A node v_i is a *descendant* of v_j if v_j is on P_{i1} . V_j will denote the set of all descendants of v_j , and $N(v_j)$ will denote the set of all the nodes adjacent to v_j . By definition we have $N(v_1) = S(v_1)$, and $N(v_j) = S(v_j) \cup \{v_{p(j)}\}$, for $v_j \neq v_1$. For any node v_i , let F_i denote the forest obtained from T by removing v_i . For a node $v_j \neq v_i$, let $T(v_i, v_j)$ denote the unique subtree (connected component) of F_i which contains v_j . (See Figure 1.)

The edge lengths induce a distance function on $\mathcal{A}(T)$. For any pair of points

$x, y \in \mathcal{A}(T)$, we let $d(x, y)$ denote the length of the unique simple path in $\mathcal{A}(T)$ connecting x and y . The length of a path P will also be denoted by $L(P)$. $\mathcal{A}(T)$ is a metric space with respect to the above distance function. Also, for any path P and $v_k \in V$ we define $d(v_k, P) = d(P, v_k) = \min_{x \in P} d(x, v_k)$. Each node $v_i \in V$ is associated with a pair of nonnegative weights (u_i, w_i) ; u_i and w_i are the center-weight and median-weight respectively. We assume that $\sum_{i=1}^n u_i > 0$ and $\sum_{i=1}^n w_i > 0$.

Let \mathcal{P}^* be a given subset of simple paths in T . We emphasize that in this paper all the paths in \mathcal{P}^* are assumed to be discrete, i.e., the endpoints of a path are nodes of T and not arbitrary points in $\mathcal{A}(T)$. For any discrete path P , $V(P)$ will denote its node set.

We are interested in the following two criteria:

1. Weighted path center criterion

$$MAX(P) := \max_{v_b \in V} u_b d(v_b, P).$$

2. Weighted path median criterion

$$SUM(P) := \sum_{b=1}^n w_b d(v_b, P).$$

For each path $P \in \mathcal{P}^*$, a node $v_k \in P$ is called *critical* for P if the maximum center-weighted distance to P is attained at v_k , i.e., $\max_{v_b \in V} u_b d(v_b, P) = \max_{v_b \in V \setminus V(P)} u_b d(v_b, v_k)$.

The classical continuous weighted (point) 1-center problem [17] on $\mathcal{A}(T)$ is to find a point x in $\mathcal{A}(T)$ minimizing the objective

$$f(x) = \max_{v_b \in V} u_b d(x, v_b).$$

Let \bar{c}_u be the unique solution to this problem, i.e.,

$$\min_{x \in \mathcal{A}(T)} \max_{v_b \in V} u_b d(x, v_b) = \max_{v_b \in V} u_b d(\bar{c}_u, v_b).$$

The point \bar{c}_u can be obtained in $O(n)$ time by the algorithm in [18]. We augment the node set of T by \bar{c}_u and assume without loss of generality that T is rooted at \bar{c}_u , i.e., $\bar{c}_u = v_1$. We introduce some necessary concepts and notation from multi-criteria

optimization. (For further details the reader is referred to [11].) We first define the function which maps each path in \mathcal{P}^* to a point in the plane:

$$\begin{aligned} \varphi: \mathcal{P}^* &\longrightarrow \mathbb{R}^2 \\ P &\longrightarrow (MAX(P), SUM(P)). \end{aligned}$$

\mathcal{P}^* is called the decision space and the planar set $\varphi(\mathcal{P}^*) = \{(MAX(P), SUM(P)) : P \in \mathcal{P}^*\}$ is the bi-criteria or outcome space. A point $(M, S) \in \varphi(\mathcal{P}^*)$ is a nondominated outcome or a nondominated point if there is no path $P \in \mathcal{P}^*$ such that $MAX(P) \leq M$, $SUM(P) \leq S$, and $MAX(P) + SUM(P) < M + S$. Our main goal in this paper is to construct $NDO(\mathcal{P}^*)$, the *nondominated set*, defined as the subset of all the nondominated points of the planar set $\varphi(\mathcal{P}^*)$.

We will prove that $|NDO(\mathcal{P}^*)| \leq 2n$ for any subset of paths \mathcal{P}^* . We then show how to construct $NDO(\mathcal{P}^*)$ in $O(n \log n)$ optimal time for the case where \mathcal{P}^* is the set of all paths with length bounded above by L . We note that in general a nondominated outcome can correspond to several paths in \mathcal{P}^* . In fact, if T is a star tree with $n \geq 4$ nodes and all edge lengths and node weights are equal to one, all paths of T which connect two leaves are mapped to the unique nondominated outcome $(1, n - 3)$. Therefore, to obtain a compact representation, for each nondominated outcome we will find only one path in \mathcal{P}^* which is mapped into this point. We call such a path a *representative efficient path*.

3 Computing critical values for the weighted path center problem

In this section we compute the set of all the critical values for the weighted path center objective, i.e., all the different values that the function $MAX(P)$ can take on. We first prove that the cardinality of this set is at most $2n$, and then provide an $O(n \log n)$ algorithm for its construction. The result on the cardinality implies that $|NDO(\mathcal{P}^*)| \leq 2n$ for any subset of paths \mathcal{P}^* .

The $O(n)$ algorithm in [18] finds \bar{c}_u , the weighted 1-center of the tree, and also identifies a pair v_s, v_t of “bottleneck” nodes defining \bar{c}_u . (As noted in the previous

section, we add \bar{c}_u to the node set of the tree, and assume without loss of generality that T is rooted at \bar{c}_u , i.e., $\bar{c}_u = v_1$.) Specifically, $v_1 \in P_{st}$ and $u_s d(v_s, v_1) = u_t d(v_t, v_1) = \max_{v_i \in V} u_i d(v_i, v_1)$. Consider any pair of subtrees T^s and T^t such that

$$v_s \in T^s, v_t \in T^t, T^s \cap T^t = \{v_1\} \text{ and } T^s \cup T^t = T.$$

(See Figure 2.)

Throughout this section we assume that \mathcal{P}^* is any arbitrary subset of paths in T . Define \mathcal{P}^s , \mathcal{P}^t and $\bar{\mathcal{P}}$ by

$$\begin{aligned} \mathcal{P}^s &= \{P \in \mathcal{P}^* \text{ such that } P \subseteq T^s\} \\ \mathcal{P}^t &= \{P \in \mathcal{P}^* \text{ such that } P \subseteq T^t\} \\ \bar{\mathcal{P}} &= \{P_{ij} \in \mathcal{P}^* \text{ such that } v_i \in T^s, v_j \in T^t \text{ and } v_i, v_j \neq v_1\}. \end{aligned}$$

Let A denote the set of distinct objective values $MAX(P)$ for all paths $P \in \mathcal{P}^*$.

Theorem 3.1 $|A| \leq 2n$.

Proof:

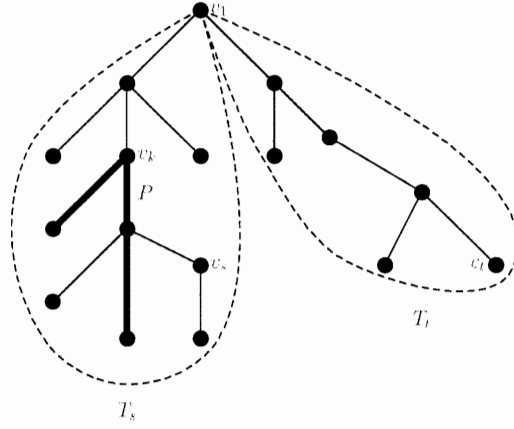
Note that $|\mathcal{P}^*|$ is bounded above by $n(n+1)/2$. Consider first a path $P_{ij} \in \bar{\mathcal{P}}$. Then P_{ij} can be represented as the union of two paths, P_{i1} and P_{1j} . Let $\beta_i^s = \max_{v_k \in T^s} u_k d(v_k, P_{i1})$ and let $\beta_j^t = \max_{v_k \in T^t} u_k d(v_k, P_{1j})$. Then, $MAX(P_{ij}) = \max(\beta_i^s; \beta_j^t)$. Secondly, consider a path $P \in \mathcal{P}^s$ and let v_k be the closest node to v_1 in P . Since v_1 is the continuous weighted center of the tree we have

$$MAX(P) = \max_{v_a \in V} u_a d(v_a, v_k).$$

For $v_k \in T^s$ define $\delta_k = \max_{v_a \in V} u_a d(v_a, v_k)$. Then, for each $P \in \mathcal{P}^s$, $MAX(P)$ is an element in $\{\delta_k : v_k \in T^s\}$. For $v_k \in T^t$ define $\delta_k = \max_{v_a \in V} u_a d(v_a, v_k)$. Using a symmetric argument, we note that for each $P \in \mathcal{P}^t$, $MAX(P)$ is an element in $\{\delta_k : v_k \in T^t\}$. To conclude the proof we note that the set A defined above satisfies

$$A \subseteq \{\beta_i^s : v_i \in T^s\} \cup \{\beta_j^t : v_j \in T^t\} \cup \{\delta_k : v_k \in V\}. \quad \square$$

We note that the above proof demonstrates that if $P \in \mathcal{P}^s \cup \mathcal{P}^t$ and v_k is the closest node to v_1 in P , then v_k is the only critical node of P . (See Figure 2.)


 Figure 2: The case $P \in \mathcal{P}^s$.

Define

$$A^* = \{\beta_i^s : v_i \in T^s\} \cup \{\beta_j^t : v_j \in T^t\} \cup \{\delta_k : v_k \in V\}.$$

The above theorem shows that the set A^* is a superset of A . We next refine the definition of a superset containing A .

Let $\{v_p, v_q\}$, $v_p \in T^s$ and $v_q \in T^t$, be a pair of end nodes defining the optimal weighted path center P_{pq} : i.e., $MAX(P_{pq}) \leq MAX(P_{ij})$ for all pairs $v_i, v_j \in V$. If the optimal path is not unique we let P_{pq} denote the intersection of all the optimal solutions, which is by itself an optimal solution. Clearly, $v_1 = \bar{c}_u \in P_{pq}$. An optimal weighted path center can be obtained in $O(n \log n)$ time by the algorithm in [33].

We use the following definitions in further refining a superset containing A . For each node $v_k \in P_{pq}$, $v_k \neq v_1$, let $v_{a(k)}$ be the child of v_k on P_{pq} . For convenience we define $v_{a(p)} = v_p$ and $v_{a(q)} = v_q$. For v_1 we let $v_{a^s(1)}$ ($v_{a^t(1)}$) be the child of v_1 on P_{p1} (P_{1q}). Also, for $v_k \neq v_1$ set

$$\gamma_k = \max_{v_b \in V_{a(k)}} u_b d(v_b, v_k),$$

and for $v_k = v_1$ set

$$\gamma_1^s = \max_{v_b \in V_{a^s(1)}} u_b d(v_b, v_1),$$

$$\gamma_1^t = \max_{v_b \in V_{a^t(1)}} u_b d(v_b, v_1).$$

We note that v_1 was selected to be the continuous weighted center point of the tree,

and therefore $\gamma_1^s = \gamma_1^t$. Hence, we adopt the notation

$$\gamma_1 = \gamma_1^s = \gamma_1^t.$$

Define

$$\epsilon^s = \max_{v_a \in T^s} u_a d(v_a, P_{pq}),$$

and

$$\epsilon^t = \max_{v_a \in T^t} u_a d(v_a, P_{pq}).$$

In particular, $MAX(P_{pq}) = \max\{\epsilon^t, \epsilon^s\}$.

Properties of P_{pq} :

1. If $v_k \in P_{pq}$ and $v_k \notin \{v_p, v_q, v_1\}$, then

$$\gamma_k = \max_{v_b \in V_{a(k)}} u_b d(v_b, v_k) > \max_{v_b \in V_k \setminus V_{a(k)}} u_b d(v_b, v_k),$$

and $\gamma_k > \max\{\epsilon^s, \epsilon^t\}$.

2. If $v_1 \neq v_p$, then

$$\gamma_1^s = \max_{v_b \in V_{a^s(1)}} u_b d(v_b, v_1) > \max_{v_b \in T^s \setminus V_{a^s(1)}} u_b d(v_b, v_1),$$

and $\gamma_1^s > \max\{\epsilon^s, \epsilon^t\}$.

3. If $v_1 \neq v_q$, then

$$\gamma_1^t = \max_{v_b \in V_{a^t(1)}} u_b d(v_b, v_1) > \max_{v_b \in T^t \setminus V_{a^t(1)}} u_b d(v_b, v_1),$$

and $\gamma_1^t > \max\{\epsilon^s, \epsilon^t\}$.

We next show that the only candidates to be critical nodes of the efficient paths in \bar{P} are the nodes of P_{pq} . (See Figure 3.)

Lemma 3.1 *Let $P \in \bar{P}$. Then any critical node of P must belong to P_{pq} . Moreover, $\max_{v_b \in V} u_b d(v_b, P)$ is an element of the set*

$$A^{st} = \{\gamma_k : v_k \in P_{pq}\} \cup \{\epsilon^s, \epsilon^t\}.$$

Proof:

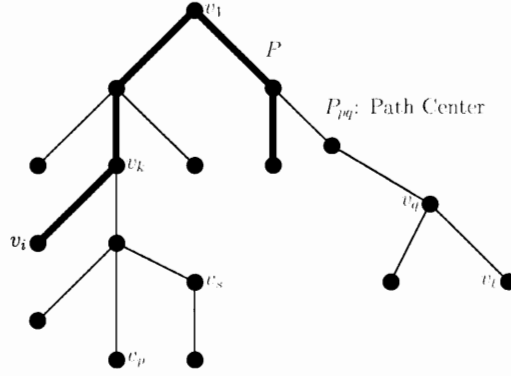


Figure 3: The case $P \in \overline{\mathcal{P}}$.

Consider a path $P \in \overline{\mathcal{P}}$ and let v_r be a critical node of P . Suppose without loss of generality that $v_r \in T^s$ and v_r is not on P_{p1} . Let v_k be the closest node to v_r in P_{p1} . (Throughout this proof, if $v_k = v_1$ or $v_c = v_1$, the set V_1 should be replaced by $V_1 \cap T^s$ and the index $a(1)$ should be replaced by $a^s(1)$.) Let v_i be the leaf node (endpoint) of P in T^s . Let $v_m \in V$ be such that

$$u_m d(v_m, P) = u_m d(v_m, v_r) = \max_{v_a \in T^s} u_a d(v_a, P_{i1}) = MAX(P).$$

Notice that $v_m \in V_r$. Suppose first that $v_k \neq v_p$. Then, since $v_r, v_m \in V_k \setminus V_{a(k)}$ and $v_r \in P_{mk}$, we can apply Property 1 (or Property 2, in case $v_k = v_1$) to conclude that

$$\gamma_k = \max_{v_b \in V_{a(k)}} u_b d(v_b, v_k) > \max_{v_b \in V_k \setminus V_{a(k)}} u_b d(v_b, v_k) \geq u_m d(v_m, v_k) > u_m d(v_m, v_r).$$

Hence, we have contradicted the fact that v_r is a critical node of P .

Next, suppose that $v_k = v_p$, i.e., $v_r \in V_p$. In this case $P_{p1} \subset P_{i1}$ and

$$MAX(P) = u_m d(v_m, v_r) < u_m d(v_m, v_p) \leq \gamma_p \leq MAX(P_{pq}).$$

The inequality $MAX(P) < MAX(P_{pq})$ contradicts the optimality of the path P_{pq} for the weighted center objective.

Therefore, we conclude that any critical node v_k of P must be a node of P_{pq} .

Now we prove that for each $P \in \overline{\mathcal{P}}$, the term $MAX(P)$ is an element of the set A^{st} .

Let $v_k \in P_{pq}$ be a critical node of $P \in \overline{\mathcal{P}}$. Without loss of generality suppose that $v_k \in P_{p1}$. From the above properties of P_{pq} and the fact that $v_{a(p)} = v_p$, it follows that

for each node $v_c \in P_{p1}$,

$$\max_{v_a \in V_c} u_a d(v_a, v_c) = \max_{v_a \in V_{a(c)}} u_a d(v_a, v_c).$$

Suppose first that P does not contain v_p . If v_k is the closest node to v_p on P , then from the above we obtain $MAX(P) = \max_{v_a \in V_{a(k)}} u_a d(v_a, v_k) = \gamma_k$. Next, suppose that $v_c \neq v_k$ is the closest node to v_p on P . Note that in this case, since $v_k, v_c \in P_{p1}$ and $v_k, v_c \neq v_p$, we have

$$\gamma_c \leq \max_{v_a \in V} u_a d(v_a, P) = \max_{v_a \in V_k \setminus V_{a(k)}} u_a d(v_a, v_k) \leq \epsilon^s.$$

However, by Properties 1 and 3, $\gamma_c > \epsilon^s$ and we obtain a contradiction.

Now suppose that P contains v_p , i.e., P_{p1} is a subpath of P . Then

$$MAX(P_{pq}) \leq MAX(P) \leq \max_{v_a \in T^s} u_a d(v_a, P_{p1}) = \epsilon^s \leq MAX(P_{pq}).$$

Therefore, $\max_{v_a \in V} u_a d(v_a, P) = \epsilon^s$ and the proof is complete. \square

There are two important consequences of the above lemma. First, we do not have to consider critical nodes of paths $P \in \overline{\mathcal{P}}$ outside P_{pq} . Secondly, we only need to compute maximum weighted distances to nodes in P_{pq} from nodes in their descendant subtrees.

Define

$$A^{**} = \{\gamma_k : v_k \in P_{pq}\} \cup \{\epsilon^s, \epsilon^t\} \cup \{\delta_k : v_k \in V\}. \quad (1)$$

The above lemma implies the following theorem.

Theorem 3.2 *Let $NDO(\mathcal{P}^*)$ be the subset of all nondominated outcomes of the planar set $\varphi(\mathcal{P}^*) = \{(MAX(P), SUM(P)) : P \in \mathcal{P}^*\}$. Then, for each point (M, S) in $NDO(\mathcal{P}^*)$, $M \in A^{**}$.*

The reader may note that the above theorem implies that our approach can be viewed as an efficient application of the well-known general ϵ -constraint approach (see [11]) since the elements in A^{**} are in fact the different ϵ -values used in that method to generate the set $NDO(\mathcal{P}^*)$. Indeed, we identify the entire set of necessary ϵ -values in linear time and we solve all the corresponding ϵ -constraint problems in $O(n \log n)$.

3.1 Generating A^{**}

We now show how to generate A^{**} in $O(n \log n)$ time.

First we note that δ_k is the value of the weighted (point) 1-center function $f(x)$, defined in Section 2, evaluated at v_k . Using the centroid decomposition we can compute the weighted 1-center objective value at all nodes of a tree network in $O(n \log n)$ time (see Tamir [31]).

Next, recall that the weighted path center P_{pq} can be found in $O(n \log n)$ time by the algorithm in [33]. In particular, this is the effort to find $\{\epsilon^s, \epsilon^t\}$.

Finally, consider the computation of the set $\{\gamma_k : v_k \in P_{p1}, v_k \neq v_1\} \cup \{\gamma_1^s\}$. (A similar procedure is then applied to compute $\{\gamma_k : v_k \in P_{1q}, v_k \neq v_1\} \cup \{\gamma_1^t\}$.) We show how to compute all these terms in $O(n \log n)$ time.

Without loss of generality, suppose that each node $v_k \in P_{p1}$ is a point, say v'_k , on the real line, where $v'_p = 0$ and $v'_k = d(v_p, v_k)$.

For each $v_b \in T^s$, let $v_{i(b)}$ be the closest node on P_{p1} to v_b , and let $g_b(x)$ be defined by $g_b(x) = 0$ if $x < v'_{i(b)}$, and

$$g_b(x) = u_b[d(v_b, v_{i(b)}) + (x - v'_{i(b)})] \text{ when } x \geq v'_{i(b)}.$$

Then

$$G(x) = \max_{v_b \in T^s} g_b(x).$$

$G(x)$ is piecewise linear and it can be generated in $O(n \log n)$ time by the algorithm in [14]. Moreover, it is easy to verify that for each $v_k \in P_{p1}, v_k \neq v_1$, $\gamma_k = G(v'_k)$ and $\gamma_1^s = G(v'_1)$.

We remark that $G(x)$ has at most $2n$ breakpoints. This follows directly from the fact that each pair of functions in the collection $\{g_b(x) : v_b \in T^s\}$ can intersect in at most at two points (see Sharir and Agarwal [26]).

Finally, we comment on the unweighted center function, i.e., the case where the center weights are identical. This is the model studied in [3, 4, 5]. In this case, for each $v_k \in T^s$, $\gamma_k = d(v_k, v_s)$ and $\delta_k = d(v_k, v_t)$. Similarly, for each $v_k \in T^t$, $\gamma_k = d(v_k, v_t)$ and $\delta_k = d(v_k, v_s)$. In particular, the set A^{**} can be computed in $O(n)$ time.

4 Obtaining the weighted center-median nondominated set $NDO(\mathcal{P}^*)$ when \mathcal{P}^* is the set of all paths of T

In this section we focus on the case where \mathcal{P}^* is the set of all paths of T , and show how to obtain $NDO(\mathcal{P}^*)$, the complete set of all doubly weighted center-median nondominated outcomes, in $O(n \log n)$ time.

We note that the complexity of the algorithm presented here is the same as the complexity of the algorithm presented in the next section for solving the more general model with a length constraint. However, the algorithm in this section is much simpler. Moreover, unlike the more general algorithm, its complexity reduces to $O(n)$ in the unweighted case of the center objective. It is obvious that in the case without a length constraint we can assume that \mathcal{P}^* is the set of all paths connecting two leaves of T . From the previous sections we conclude that $|NDO(\mathcal{P}^*)| \leq 2n$. To generate $NDO(\mathcal{P}^*)$ we first compute a planar set W , satisfying $|W| = O(n)$ and

$$NDO(\mathcal{P}^*) \subseteq W \subseteq \varphi(\mathcal{P}^*).$$

We remark that the set W is the set of points that would be obtained after solving the ε -constraint problems taking A^{**} as the admissible ε values. Specifically, we note that the projection of W on the first coordinate (the weighted center criterion) will be the set A^{**} , defined in (1), and the projection on the second coordinate (the weighted median objective) will be the optimal value of each ε -constraint problem. (Note that $NDO(\mathcal{P}^*)$ is then the rectilinear lower envelope of the points in W , and it can be obtained from W in $O(n \log n)$ time [16].)

To complete the analysis it remains to show how to compute the set W .

4.1 Computing the set W

Consider the following terms. Let v_i, v_j be a pair of distinct nodes such that $(v_i, v_j) \in E$:

$$SUM_0(v_i) = \sum_{v_b \in V_i} w_b d(v_b, v_i), \quad (2)$$

$$SUM_1(v_i, v_j) = \sum_{v_b \in T(v_i, v_j)} w_b d(v_b, v_i), \quad (3)$$

$$SUM_2(v_i, v_j) = \min_{v_b \in T(v_i, v_j)} \sum_{v_d \in T(v_i, v_j)} w_d d(v_d, P_{ib}), \quad (4)$$

$$SUM_3(v_i, v_j) = SUM_1(v_i, v_j) - SUM_2(v_i, v_j). \quad (5)$$

Note that for any node v_i the weighted sum of the distances from its descendants satisfies:

$$SUM_0(v_i) = \sum_{v_j \in S(v_i)} SUM_1(v_i, v_j).$$

The values of $SUM_t(v_i, v_j)$, $t = 1, 2, 3$, for all (v_i, v_j) can be computed in $O(n)$ time using a straightforward modification of the algorithm in Morgan and Slater [21].

We will now construct a set W with the above properties. Specifically, for each value $\alpha \in A^{**}$, we will find a path of T which minimizes the weighted sum function among the subset of paths whose weighted center function is equal to α .

Case 1. In this case we consider all points in W , where the value of α is in the set $\{\delta_k : v_k \in V\}$, and denote by W^1 the corresponding elements of W . From the above discussion the corresponding paths are contained either in \mathcal{P}^s or \mathcal{P}^t . In this case, a critical node of a path is the closest node of this path to the weighted 1-center point $\bar{c}_u = v_1$; that is, v_k is a critical point for P if $d(v_k, \bar{c}_u) = \min_{x \in P} d(x, \bar{c}_u)$.

Given a critical node v_k with its respective value δ_k , an efficient path must be such that it extends from v_k in the directions of two distinct descendants of v_k with maximum decrease in the sum objective. (Recall that we have assumed that the tree graph T is rooted at $\bar{c}_u = v_1$.) Let P be a candidate efficient path having v_k as critical node. Then, we have $MAX(P) = \max_{v_b \in V} w_b d(v_b, v_k) = \delta_k$. To find P we identify $v_i, v_j \in S(v_k)$, $v_i \neq v_j$, (i.e., two children of v_k) such that $SUM_3(v_k, v_i) \geq SUM_3(v_k, v_j) \geq SUM_3(v_k, v_b)$ for any $v_b \in S(v_k) \setminus \{v_i, v_j\}$. Then

$$SUM(P) = \sum_{v_b \in N(v_k)} SUM_1(v_k, v_b) - SUM_3(v_k, v_i) - SUM_3(v_k, v_j). \quad (6)$$

Hence, $(MAX(P), SUM(P))$ is a candidate to be a nondominated point and it is added to W^1 . (We assume that all the terms $SUM_t(v_i, v_j)$, $t = 1, 2, 3$, $(v_i, v_j) \in E$, have already been computed in $O(n)$ time.) Hence, for each δ_k the effort to compute $SUM(P)$ above is $O(|N(v_k)|)$.

Given the set $\{\delta_k : v_k \in V\}$, the additional total time spent to generate all the points in W^1 corresponding to Case 1 is $O(n)$, since $\sum_{v_k \in V} |N(v_k)| = 2n - 2$.

Case 2. In this case we consider points in W corresponding to the paths in $\bar{\mathcal{P}}$. Specifically, from the previous section, we focus on instances where the value of α is either in $\{\epsilon^s, \epsilon^t\}$ or in $\{\gamma_k : v_k \in P_{pq}\}$.

To simplify the notation we now use γ_p to denote $\max\{\gamma_p, \epsilon^s\}$ and γ_q to denote $\max\{\gamma_q, \epsilon^t\}$.

We first introduce some notation. For each $v_k \in P_{p1}$, let

$$\bar{\mathcal{P}}_k^s = \{P_{i1} : v_i \in T^s, P_{i1} \cap P_{p1} = P_{k1}\},$$

$$S^s(k) = \min_{P_{i1} \in \bar{\mathcal{P}}_k^s} \sum_{v_r \in T^s} w_r d(v_r, P_{i1}).$$

Similarly, for each $v_k \in P_{1q}$, let

$$\bar{\mathcal{P}}_k^t = \{P_{1i} : v_i \in T^t, P_{1i} \cap P_{1q} = P_{1k}\},$$

$$S^t(k) = \min_{P_{1i} \in \bar{\mathcal{P}}_k^t} \sum_{v_r \in T^t} w_r d(v_r, P_{1i}).$$

We are now ready to generate W^2 , the set of all the points in W corresponding to Case 2.

For $v_k \in P_{p1}$ set

$$b^s(k) = S^s(k) + \min_{v_b \in P_{1q}, \gamma_b \leq \gamma_k} S^t(b).$$

For $v_k \in P_{1q}$ set

$$b^t(k) = S^t(k) + \min_{v_b \in P_{p1}, \gamma_b \leq \gamma_k} S^s(b).$$

Define

$$W^2 = \{(\gamma_k, b^s(k)) : v_k \in P_{p1}\} \cup \{(\gamma_k, b^t(k)) : v_k \in P_{1q}\}.$$

From the above definitions it is clear that the projection of W^2 on the left coordinate is the set A^{st} defined in the previous section. Moreover, for each MAX value γ_k , $v_k \in P_{p1}$ ($\gamma_k, v_k \in P_{1q}$), $b^s(k)$ ($b^t(k)$) is indeed the correct SUM value, i.e., $W^2 \subseteq \varphi(\mathcal{P}^*)$.

We conclude by showing that the total effort to compute the subset W^2 is $O(n)$, when the set A^{st} is already available. We first introduce some additional notation. For each $v_k \in P_{p1}$, let

$$f_k^s = \min_{v_b \in P_{1q}, \gamma_b \leq \gamma_k} S^t(b),$$

and for each $v_k \in P_{1q}$, let

$$f_k^t = \min_{v_b \in P_{p1}, \gamma_b \leq \gamma_k} S^s(b).$$

From the above expressions it will suffice to prove that all the terms $\{S^s(k)\}$, $\{S^t(k)\}$, $\{f_k^s\}$ and $\{f_k^t\}$ can be computed in linear time. The sequence $\{\gamma_k\}$ is decreasing when we move v_k along P_{p1} from v_1 to v_p , and when we move v_k along P_{1q} from v_1 to v_q . Therefore, the sets $\{f_k^s : v_k \in P_{p1}\}$ and $\{f_k^t : v_k \in P_{1q}\}$ can be computed in $O(n)$ time when the terms $\{S^s(k)\}$, $\{S^t(k)\}$, are available. We now show how to compute $\{S^s(k)\}$ and $\{S^t(k)\}$ in $O(n)$ time.

To simplify the presentation we will focus only on the set $\{S^s(k)\}$, and assume that the nodes in P_{p1} are renumbered consecutively with $v_{a^s(1)} = v_2$, $v_{a^s(2)} = v_3$, etc. In particular, if there are m nodes on P_{p1} , $v_p = v_m$. Compute:

$$SUM_4(v_1) = 0,$$

$$SUM_4(v_k) = SUM_4(v_{k-1}) + SUM_0(v_k) - (SUM_0(v_{k+1}) - W_{k+1}d(v_{k+1}, v_k)), \quad k = 2, \dots, m-1,$$

where $W_k = \sum_{v_i \in V_k} w_i$. ($\{W_k\}_{k=1}^m$ and the values $SUM_4(v_k)$ for all v_k , $k = 1, \dots, m-1$, can be computed in $O(n)$ time.)

Notice that $SUM_4(v_k)$, for any $v_k \in P_{p1}$, gives the sum of the weighted distances to P_{k1} from all the nodes v_r such that $d(v_r, P_{p1}) = d(v_r, P_{k1})$, i.e.,

$$SUM_4(v_k) = \sum_{v_r \in T^s \setminus V_{a(k)}} w_r d(v_r, P_{k1}).$$

(If $v_k = v_1$, replace the index $a(1)$ by $a^s(1)$.) It is now easy to verify that for $v_k \in P_{p1}$, $v_k \neq v_1$,

$$S^s(k) = SUM_0(v_k) - \max_{v_r \in S(v_k) \setminus P_{pk}} SUM_3(v_k, v_r) + SUM_4(v_{p(k)}),$$

and

$$S^s(1) = \sum_{v_r \in T^s} w_r d(v_r, v_1) - \max_{v_r \in (T^s \cap S(v_1)) \setminus P_{p1}} SUM_3(v_k, v_r).$$

From the above expressions it is clear that the total time to compute all the terms $\{S^s(k)\}$ is $O(n)$. Hence, given the set A^{st} , the total effort to compute the set W^2 is linear.

To summarize, the total effort needed to compute the superset $W = W^1 \cup W^2$ which contains all the nondominated outcomes is $O(n \log n)$. (Note that with the above scheme we can also record for each point in W a representative path corresponding to the *MAX* and *SUM* values of that point.) Given the planar set W , we can then identify the nondominated set itself in time $O(|W| \log |W|)$; see [16].

Therefore, the overall complexity of identifying $NDO(\mathcal{P}^*)$, the complete set of nondominated outcomes, and the respective representative efficient paths of the doubly weighted center-median path problem on a tree graph is $O(n \log n)$.

Theorem 4.1 *When \mathcal{P}^* is the set of all paths in T , $NDO(\mathcal{P}^*)$, the nondominated set corresponding to the doubly weighted model, can be generated in $O(n \log n)$ time.*

5 Obtaining the weighted center-median nondominated set $NDO(\mathcal{P}^*)$ when \mathcal{P}^* is the set of all paths of T with length bounded above by L

In this section we consider the case where $\mathcal{P}^* = \mathcal{P}_L$, the set of all paths with length less than or equal to L . Again, our goal is to identify the $O(n)$ nondominated outcomes (and the $O(n)$ respective representative efficient paths) for this restricted model. Following the results in Section 3, in order to effectively obtain all the candidate pairs to be nondominated points for the constrained-length *MAX* and *SUM* objectives, we will compute for each value $\alpha \in A^{**}$ a path with the minimum value for the *SUM* objective, among all paths, with *MAX* value equal to α and with length bounded above by L . Again, from Section 3, $|NDO(\mathcal{P}_L)| \leq 2n$. We will construct a superset W_L , with

cardinality $O(n)$, containing all nondominated outcomes for this restricted model, i.e.,

$$NDO(\mathcal{P}_L) \subseteq W_L \subseteq \varphi(\mathcal{P}_L).$$

W_L is the union of the sets W_L^1 and W_L^2 , defined as follows:

The set W_L^1 corresponds to all feasible paths, where the MAX objective is in the set $\{\delta_k : v_k \in V\}$. (From Section 2, these are the feasible paths in $\mathcal{P}^s \cup \mathcal{P}^t$.) Specifically,

$$W_L^1 = \{(MAX_L(v_k), SUM_L(v_k)) : v_k \in V\},$$

where

$$MAX_L(v_k) = \max_{v_j \in V} u_j d(v_k, v_j) = \delta_k,$$

$$SUM_L(v_k) = \min_{\{P:L(P) \leq L, V(P) \subseteq V_k, v_k \in P\}} \sum_{v_i \in V} w_i d(v_i, P).$$

Note that $SUM_L(v_k)$ is the optimal value of the SUM objective among all paths which are contained in V_k , include v_k , and have length bounded above by L .

Similarly, the set W_L^2 corresponds to all feasible paths where the MAX objective is in the set $\{\gamma_k : v_k \in P_{pq}\} \cup \{\epsilon^s, \epsilon^t\}$.

Computing W_L^1

As noted in Section 2, we can use the procedure in Tamir [31] to compute $MAX_L(v_k) = \delta_k$ for all nodes $v_k \in V$ in $O(n \log n)$ time.

The terms $\{SUM_L(v_k) : v_k \in V\}$ can be computed in $O(n \log n)$ time by the algorithm presented in Alstrup et al. [1, 2] using the top trees data structure. (Note that the term $SUM_L(v_k)$ coincides with the term $MinCost(v_k)$ in Alstrup et al. [1].)

Computing W_L^2

In this case we focus on the set of all feasible paths where the MAX objective is in the set $\{\gamma_k : v_k \in P_{pq}\} \cup \{\epsilon^s, \epsilon^t\}$. Using the notation in Section 2, these are the feasible paths in $\overline{\mathcal{P}}$.

Using the same notation as in Section 4, let γ_p denote $\max\{\gamma_p, \epsilon^s\}$ and γ_q denote $\max\{\gamma_q, \epsilon^t\}$.

To explain our algorithmic approach, consider a node $v_i \in T^s$ and the path P_{i1} . Suppose that $P_{i1} \cap P_{p1} = P_{k(i)1}$ for some node $v_{k(i)} \in P_{p1}$. Then, from the above,

$\max_{v_b \in T^s} u_b d(v_b, P_{i1}) = \gamma_{k(i)}$. We now define b_i^s to be the smallest value of the *SUM* objective, over the set of paths $\widehat{\mathcal{P}}_i^s$ defined by

$$\widehat{\mathcal{P}}_i^s = \{P_{ij} : v_j \in T^t, \text{MAX}(P_{ij}) = \gamma_{k(i)}, L(P_{ij}) \leq L\}.$$

Informally, b_i^s is the best value of the *SUM* objective that can be achieved by extending the path P_{i1} towards T^t while maintaining $v_{k(i)}$ as the critical node. It is given by

$$b_i^s = \min_{P_{ij} \in \widehat{\mathcal{P}}_i^s} \sum_{v_b \in V} w_b d(v_b, P_{ij}).$$

Similarly, for each node $v_j \in T^t$, we define the node $v_{k(j)}$, the set $\widehat{\mathcal{P}}_j^t$, and the value b_j^t .

It is now clear that it is sufficient for our purposes to define

$$W_L^2 = \{(\gamma_{k(i)}, b_i^s) : v_i \in T^s\} \cup \{(\gamma_{k(j)}, b_j^t) : v_j \in T^t\}.$$

Next, we show how to construct the set W_L^2 in $O(n \log n)$ time. Due to symmetry, we only show how to compute the set of points $\{(\gamma_{k(i)}, b_i^s) : v_i \in T^s\}$.

To facilitate the discussion, for each node $v_i \in T^s$ define

$$S_i^s = \sum_{v_b \in T^s} w_b d(v_b, P_{i1}).$$

Similarly, for each node $v_j \in T^t$, set

$$S_j^t = \sum_{v_b \in T^t} w_b d(v_b, P_{1j}).$$

(The effort to compute all these values is $O(n)$, if we use a top down approach, starting at the root v_1 .) Then, for $v_i \in T^s$, we have

$$b_i^s = S_i^s + \min_{\{v_j \in T^t : \gamma_{k(j)} \leq \gamma_{k(i)}, d(v_i, v_j) \leq L\}} S_j^t.$$

Recall that the sequence $\{\gamma_k\}$ is decreasing when we move v_k from v_1 to v_p , and is increasing when we move v_k from v_q to v_1 . We compute the terms $\{b_i^s\}$, following the ordering of the nodes $\{v_{k(i)}\}$, along P_{p1} .

To simplify the notation, suppose that the nodes along P_{p1} are numbered consecutively with $v_{a(1)} = v_2$, $v_{a(2)} = v_3$, etc. In particular, if there are m nodes on P_{p1} , $v_p = v_m$.

We begin with the node v_m and consider all nodes $v_i \in T^s$ such that $v_{k(i)} = v_m$, i.e., $v_i \in V_m$. By scanning the path P_{1q} , we find $v_{t(m)}$, the closest node to v_1 on P_{1q} such that $\gamma_{t(m)} \leq \gamma_m$. With each node $v_j \in T^t$, such that v_j is a descendant of $v_{t(m)}$, associate the planar point $(d(v_1, v_j), S_j^t)$.

We maintain a list \mathcal{L}^t whose elements are the above points arranged in nonincreasing order with respect to the first coordinate of its elements. By scanning the list, and eliminating dominated points, we can assume that the remaining points are arranged in increasing order of their first coordinate, and the respective values of the second coordinate form a decreasing sequence. The total effort of this initial phase is $O(n \log n)$.

For each $v_i \in V_m$, we now show how to compute b_i^s in $O(\log n)$ time. Find the point, say $(d(v_1, v_{j(i)}), S_{j(i)}^t)$, in the list \mathcal{L}^t with the largest value of its first coordinate, which is smaller than or equal to $L - d(v_i, v_1) - d(v_1, v_{j(i)})$. It is then clear that $b_i^s = S_i^s + S_{j(i)}^t$.

In the next step we proceed to v_{m-1} , the father of v_m along P_{p1} , and process all the nodes $v_i \in T^s$ such that $v_{k(i)} = v_{m-1}$. By scanning the path P_{1q} from $v_{t(m)}$ towards v_1 , we find $v_{t(m-1)}$, the closest node to v_1 on P_{1q} such that $\gamma_{t(m-1)} \leq \gamma_{m-1}$. With each node $v_j \in V_{t(m-1)} \setminus V_{t(m)}$, associate the planar point $(d(v_1, v_j), S_j^t)$. Insert all these points into the list \mathcal{L}^t , while eliminating dominated points. Again, we assume that the remaining points are arranged in increasing order of their first coordinate, and the respective values of the second coordinate form a decreasing sequence. As above for each $v_i \in V_{m-1} \setminus V_m$, we now compute b_i^s in $O(\log n)$ time. Find the point, say $(d(v_1, v_{j(i)}), S_{j(i)}^t)$, in the updated list \mathcal{L}^t with the largest value of its first coordinate, which is smaller than or equal to $L - d(v_i, v_1) - d(v_1, v_{j(i)})$. It is then clear that $b_i^s = S_i^s + S_{j(i)}^t$.

We then proceed to v_{m-2} , the father of v_{m-1} on P_{1p} , and compute the term b_i^s for all nodes $v_i \in T^s$ such that $v_{k(i)} = v_{m-2}$, etc.

Overall, there are m steps. If there are n' insertions into, and n'' deletions from the list \mathcal{L}^t during a step, the respective effort is $O(n' \log n' + n'')$. Therefore the total effort to compute the set $\{(\gamma_{k(i)}, b_i^s) : v_i \in T^s\}$ is $O(n \log n)$.

To summarize, the total effort needed to compute the set $W_L = W_L^1 \cup W_L^2$, satisfying $NDO(\mathcal{P}_L) \subseteq W_L \subseteq \varphi(\mathcal{P}_L)$, is $O(n \log n)$. (Note that with the above scheme we can

also record for each point in W_L a representative path corresponding to the MAX and SUM values of that point.) Given the planar superset W_L , we can then identify $NDO(\mathcal{P}_L)$, the nondominated set itself, in $O(|W_L| \log |W_L|)$ time; see [16].

Theorem 5.1 *Let \mathcal{P}_L be the set of all paths in T whose length is bounded above by L . Then, $NDO(\mathcal{P}_L)$, the nondominated set corresponding to the doubly weighted model, can be computed in $O(n \log n)$ time.*

We note that if T itself is a path, $NDO(\mathcal{P}_L)$ can be obtained in linear time.

When the set W_L is already available, we can solve a general class of minimization problems in linear time. Consider a general cost function $F(MAX(P), SUM(P))$, which is assumed to be monotone nondecreasing in its two arguments. If F can be evaluated at each planar point in constant time, we conclude that a path minimizing $F(MAX(P), SUM(P))$ can be obtained in $O(n)$ time by evaluating F at each point of W_L . For example, we can solve the minimization problems mentioned below ((7)-(9)) in linear time by enumerating all the points in W_L , improving and extending the results in [3, 4, 5]. When the set $NDO(\mathcal{P}_L)$ is given and ordered according to the MAX coordinates, we can generate the extreme points of its convex hull in $O(n)$ time, and then solve each of these minimization problems in $O(\log n)$ time.

Theorem 5.2 *Let \mathcal{P}_L be the set of all paths in T whose length is bounded above by L . Suppose that $NDO(\mathcal{P}_L)$, the nondominated set corresponding to the doubly weighted model, has already been computed and ordered according to the MAX coordinate. Then the extreme points of the convex hull of $NDO(\mathcal{P}_L)$ can be computed in $O(n)$ time and for any positive real α the following problems can be solved in $O(\log n)$ time:*

The doubly weighted path α -centdian problem, defined by

$$\min\{MAX(P) + \alpha SUM(P) : P \in \mathcal{P}_L\}, \quad (7)$$

the doubly weighted, center α -restricted path median problem, defined by

$$\min\{SUM(P) : P \in \mathcal{P}_L, MAX(P) \leq \alpha\}, \quad (8)$$

and the doubly weighted, median α -restricted path center problem, defined by

$$\min\{MAX(P) : P \in \mathcal{P}_L, SUM(P) \leq \alpha\}. \quad (9)$$

Note that papers [4, 5] consider only the version where the center criterion is unweighted and give $O(n \log^2 n)$ algorithms for solving the last pair of problems. We solve these problems even for the doubly weighted model in $O(n \log n)$ time.

Also, when the set $NDO(\mathcal{P}_L)$ is available, and ordered by its *MAX* or *SUM* coordinates, we can obtain in linear time the partition of the α parameter space associated with each optimal solution of the above problems. This is obvious for the two α -restricted problems of the last theorem. To obtain the partition of the α parameter for the centdian problem, we only need to generate the extreme points on the lower envelope of the convex hull of $NDO(\mathcal{P}_L)$. As noted above, the latter task can be performed in $O(n)$ time, since the set $NDO(\mathcal{P}_L)$ is already ordered by one of its coordinates.

5.1 Lower bounds

We claim that the above $O(n \log n)$ algorithm to construct the set $NDO(\mathcal{P}_L)$ (or even the set W_L) for the doubly weighted model is in fact optimal.

It is shown in [28, 37] that $\Omega(n \log n)$ is a lower bound on the complexity of the problem $\min\{SUM(P) : P \in \mathcal{P}_L\}$. Hence, from the above discussion and the last theorem, we conclude that the $O(n \log n)$ complexity of our algorithm to find $NDO(\mathcal{P}_L)$ and W_L is indeed optimal.

6 Open problems

We have presented above optimal $O(n \log n)$ algorithms to solve path location problems on a tree network, when the objective is a monotone function of the doubly weighted center, median criteria. It is an open question whether there are subquadratic algorithms for more general, or other common objective functions, that have been studied recently in the context of extensive facilities.

For example, in [10], the authors present an $O(n^2 \log n)$ algorithm for locating a path minimizing the variance of distances of the nodes from the path.

Another example is the k -centrum model and its generalization, known as the convex ordered median model, studied recently in [25] with respect to subtree facilities. (In

the k -centrum problem, the objective is to minimize the sum of the k largest weighted distances from the selected facility. See [25] for the definition of the convex ordered median objective.) The question is whether one can locate a path facility with any of these functions in subquadratic time.

Finally, in view of the lower bounds on complexity from the last section, we conclude that the $O(n \log n)$ algorithms to solve the first two problems in Theorem 5.2 can not be improved. We do not know of any lower bound on the complexity of the third problem. However, currently, even the best bound known [33] for solving the unrestricted weighted path problem is still $O(n \log n)$. In addition to this open problem we can also list the following:

Can we solve the length restricted problem $\min\{MAX(P) : P \in \mathcal{P}_L, SUM(P) \leq \alpha\}$ in $O(n)$ time for the unweighted center model?

Can we solve the length unrestricted problem $\min\{SUM(P) : MAX(P) \leq \alpha\}$ in $O(n)$ time for the weighted center model?

Acknowledgements

The research of the first and second authors is partially supported by grants BFM2001-2378, BFM2001-4028, BFM2004-0909 and HA2003-0121.

References

- [1] S. Alstrup, P.W. Lauridsen, P. Sommerlund, and M. Thorup, Finding cores of limited length, Proc 5th International Workshop on Algorithms and Data Structures (WADS), Lecture Notes in Computer Science, Vol. 1272, Springer, 1997, pp. 45–54.
- [2] S. Alstrup, P.W. Lauridsen, P. Sommerlund, and M. Thorup, Finding cores of limited length, IT-C Technical report Series 2000-4, The IT University of Copenhagen, 2001.

- [3] I. Averbakh and O. Berman, Algorithms for path medi-center of a tree, *Computers and Operations Research* 26 (1999), 1395–1409.
- [4] R.I. Becker, Y. Chiang, I. Lari, and A. Scozzari, The cent-dian path problem on tree networks, *ISAAC 2001, Lecture Notes in Computer Science, Vol. 2223*, Springer, 2001, pp. 743–753.
- [5] R.I. Becker, Y. Chiang, I. Lari, and A. Scozzari, Algorithms for central-median paths with bounded length on trees, Technical report, University of Cape Town, South Africa, 2004.
- [6] R.I. Becker, Y. Chiang, I. Lari, A. Scozzari, and G. Storchi, Finding the l -core of a tree, *Discrete Applied Mathematics* 118 (2002), 25–42.
- [7] R.I. Becker, I. Lari, and A. Scozzari, Efficient algorithms for finding the (k, l) -core of tree networks, *Networks* 40 (2003), 208–215.
- [8] R.I. Becker and Y. Perl, Finding the two-core of a tree, *Discrete Applied Mathematics* 11 (1985), 103–113.
- [9] B. Boffey and J.A. Mesa, A review of extensive facility location in networks, *European Journal of Operational Research* 95 (1996), 592–600.
- [10] T. Caceres, M.C. Lopez de-los Mozos, and J.A. Mesa, A path-variance problem on tree networks, *Discrete Applied Mathematics* 145 (2004), 72–79.
- [11] M. Ehrgott and M.M. Wiecek, “Multiobjective programming,” *Multiple criteria decision analysis: State of the art surveys*, International Series in Operations Research and Management Science, J. Figueira, S. Greco and M. Ehrgott (Editors), Springer 2005, Vol. 76, pp. 667–722.
- [12] S.L. Hakimi, E.F. Schmeichel, and M. Labbé, On locating path-or tree shaped facilities on networks, *Networks* 23 (1993), 543–555.
- [13] S.M. Hedetniemi, E.J. Cockayne, and S.T. Hedetniemi, Linear algorithms for finding the Jordan center and path center of a tree, *Transportation Science* 15 (1981), 98–114.

- [14] J. Hershberger, Finding the upper envelope of n line segments in $O(n \log n)$ time, *Information Processing Letters* 33 (1989), 169–174.
- [15] E. Jennings, Distributed algorithms for finding central paths in tree networks, *The Computer Journal* 42:7 (1999), 609–612.
- [16] S. Kapoor, Dynamic maintenance of maxima of 2-d point sets, *SIAM J. Computing* 29 (2000), 1858–1877.
- [17] O. Kariv and S.L. Hakimi, An algorithmic approach to network location problems, Part I. The p -centers, *SIAM J. Applied Mathematics* 37 (1979), 513–538.
- [18] N. Megiddo, Linear-time algorithms for linear programming in R^3 and related problems, *SIAM J. Computing* 12 (1983), 759–776.
- [19] E. Minieka, The optimal location of a path or tree in a tree network, *Networks* 15 (1985), 309–321.
- [20] E. Minieka and N.H. Patel, On finding the core of a tree with a specified length, *J. of Algorithms* 4 (1983), 345–352.
- [21] C.A. Morgan and P.J. Slater, A linear algorithm for a core of a tree, *J. of Algorithms* 1 (1980), 247–258.
- [22] S. Nickel, J. Puerto, and A.M. Rodríguez-Chía, “MCDM location problems,” *Multiple criteria decision analysis: State of the art surveys*, International Series in Operations Research and Management Science, J. Figueira, S. Greco and M. Ehrgott (Editors), Springer, 2005, Vol. 76, pp. 761–795.
- [23] S. Peng and W. Lo, Efficient algorithms for finding a core of a tree with specified length, *J. of Algorithms* 20 (1996), 445–458.
- [24] S. Peng, A.B. Stephens, and Y. Yesha, Algorithms for a core and k -tree core of a tree, *J. of Algorithms* 15 (1993), 143–159.
- [25] J. Puerto and A. Tamir, Locating tree-shaped facilities using the ordered median objective, *Mathematical Programming* 102 (2004), 313–338.

- [26] M. Sharir and P.K. Agarwal, Davenport-Schinzel sequences and their geometric applications, Cambridge University Press, 1995.
- [27] A. Shioura and M. Shigeno, The tree center problems and the relationship with the bottleneck knapsack problems, *Networks* 29 (1997), 107–110.
- [28] A. Shioura and T. Uno, An $O(n \log n)$ time algorithm for an l -core of a tree, Technical report, Tohoku University, Japan, 2002.
- [29] P.J. Slater, Locating central paths in a graph, *Transportation Science* 16 (1982), 1–18.
- [30] A. Tamir, Fully polynomial approximation schemes for locating a tree-shaped facility: A generalization of the knapsack problem, *Discrete Applied Mathematics* 87 (1998), 229–243.
- [31] A. Tamir, Sorting weighted distances with applications to objective function evaluations in single facility location problems, *Operations Research Letters* 32 (2004), 249–257.
- [32] A. Tamir and T.J. Lowe, The generalized p -forest problem on a tree network, *Networks* 22 (1992), 217–230.
- [33] A. Tamir, J. Puerto, J.A. Mesa, and A.M. Rodriguez-Chia, Conditional location of path and tree shaped facilities on trees, *J. of Algorithms* 56 (2005), 50–75.
- [34] A. Tamir, J. Puerto, and D. Perez-Brito, The centdian subtree on tree networks, *Discrete Applied Mathematics* 118 (2002), 263–278.
- [35] B.-F. Wang, Efficient parallel algorithms for optimally locating a path and a tree of a specified length in a weighted tree network, *J. of Algorithms* 34 (2000), 90–108.
- [36] B.-F. Wang, Finding a two-core of a tree in linear time, *SIAM J. on Discrete Mathematics* 15 (2002), 193–210.

- [37] B.-F. Wang, S.-C. Ku, and Y.-H. Hsieh, The conditional location of a median path, Proc 8th Ann International Conference on Computing and Combinatorics (COCOON 2002), Singapore, 2002, pp. 494–503.