The Continuous and Discrete Path Variance Problem on Trees

Justo Puerto[1], Federica Ricca[2], Andrea Scozzari[2]

[1]Universidad de Sevilla
Dep. Estadística e Investigación Operativa
puerto@us.es *
[2]Università di Roma "La Sapienza"
Dip. Statistica, Probabilità e Statistiche Applicate
{federica.ricca, andrea.scozzari}@uniroma1.it †

Abstract

In this paper we consider the problem of locating path-shaped facilities on a tree minimizing the variance objective function. This kind of objective function is generally adopted in location problems which arise in the public sector applications, such as the location of evacuation routes or mass transit routes. We consider the general case in which a positive weight is assigned to each vertex of the tree and positive real lengths are associated to the edges. We study both the case in which the path is continuous, that is, the end points of the optimal path can be either vertices or points along an edge, and the case in which the path is discrete, that is, the end points of the optimal path must lie in some vertex of the tree. Given a tree with n vertices, for both these problems we provide algorithms with $O(n^2)$ time complexity and we extend our results also to the case in which the length of the path is bounded above. Even in this case we provide polynomial algorithms with the same $O(n^2)$ complexity. In particular, our algorithm for the continuous path-variance problem improves upon a $\log n$ term the previous best known algorithm for this problem provided in [5]. Finally, we show that no nestedness property holds for (discrete and continuous) point-variance problem with respect to the corresponding path-variance.

Keywords: Path location, variance criterion, equity measures.

1 Introduction

Network facility location consists of locating facilities in a network in order to supply a set of costumers. During the last two decades there has been increasing attention to *extensive* facilities location models, that is, the location of connected structures such as path-shaped or tree-shaped facilities [6, 17, 19]. For a comprehensive review see for example [9, 18]. In these models the objective function is given either by the sum of the distances from each client to its nearest facility (*median* criterion), or by the maximum of these distances (*center* criterion). However, some authors have considered the problem of finding an optimal location of a path or a tree using either the two criteria simultaneously or a convex combination of them [1, 3, 14, 15, 16].

*The research of this author is partially supported by Spanish research grants number: MTM2004:0909.
†The research of these authors is partially supported by the Italian research grants "Azioni Integrate Italia-Spagna" (number 2.1.4.4.1.15).

Problem [1]	Continuous	Discrete
Point Variance	$O(n)$ [7]	$O(n)$ [10]
Path Variance[2]	$\mathbf{O(n^2)}$	$\mathbf{O(n^2)}$
Path Variance with Bounded Length	$\mathbf{O(n^2)}$	$\mathbf{O(n^2)}$

Table 1: Summary of results.

In recent years some authors started to study location problems with equity measures as objective function. The idea is that some salient features of real problems, like the dispersion of the clients' demand with respect to a facility, are not captured by the traditional objective functions. The dispersion measures are strictly related to the variability of the distribution of the distance from demand points to a facility and they seem to be particularly relevant when locating facilities in the public sector applications, such as the location of evacuation routes or the location of a highway in a road network.

A part of the literature devoted to location problems with equity measures is concerned with providing efficient algorithms for different objective functions. A review of the existing literature about equity measures in location theory can be found in [8]. Actually, almost all the papers focusing on equity measures deal with the location of a single point on a network [7, 10]. However, some papers related to the location of extensive facilities with equity measure appeared in the recent years: [5] provides a polynomial algorithm for locating a continuous path of minimum *variance* on tree networks, that is, the problem of locating a continuous path by minimizing the variance of the distance travelled by the customers to a facility. In addition, in [13] a path problem is studied with respect to the *range* objective function, which is given by the difference between the maximum and the minimum distance from a facility and to the *Hurwicz* objective function, which is given by a convex combination of the maximum and the minimum distance from the points of the network to the facility. Finally, [4] studies the problem of locating a path in a tree minimizing the *coefficient of variation* function, that is, the ratio between the squared root of the variance and the average distance.

In this paper we study the problem of locating a path-shaped facility on a tree minimizing the variance objective function. We consider the general case in which a positive weight is assigned to each vertex of the tree and positive real lengths are associated to the edges. We consider both the case in which the path is continuous, that is, the end points of the optimal path can be either vertices or points along an edge, and the case in which the path is discrete, that is, the end points of the optimal path must lie on some vertex of the tree. Given a tree with n vertices, for both these problems we provide algorithms with $O(n^2)$ time complexity and we extend our results also to the case in which the length of the path is bounded above. Even in this case we provide polynomial algorithms with the same $O(n^2)$ complexity. In particular, our algorithm for the continuous path-variance problem improves upon a $\log n$ term the previous algorithm presented in [5] (see Table 1). Moreover, we show that our algorithms can be applied also for the location of a path with other equity measures related to the variance function, such as the coefficient of variation.

Furthermore, with respect to the point and the vertex location problem with minimum variance, we show that the nestedness property does not hold. In fact, we provide two coun-

[1] Boldfaced results are new results in the paper.
[2] Nestedness property with respect to the point variance does not hold in any case.

terexamples showing that the continuous path that minimizes the variance function on a tree does not necessarily contain the point of minimum variance, and the discrete path that minimizes the variance function on a tree does not necessarily contain the vertex of minimum variance.

The paper is organized as follows. In Section 2 we introduce the continuous path-variance problem with unrestricted length and provide a polynomial $O(n^2)$ time algorithm for solving it. The same is done in Section 3 for the discrete version of this problem, while Section 4 is devoted to the path-variance problems with bounded length. Finally, in Section 5 we draw some conclusions and provide extensions of our results.

2 The Continuous Path-Variance Problem

We are given a tree $T = (V, E)$ with $|V| = n$. To each vertex v is associated a positive weight $w(v)$, while to each edge $e = (v, u)$ is assigned a positive real length ℓ_{vu}. We may assume that $W(T) = \sum_{v \in V} w(v) = 1$ and interpret each $w(v)$ as the fraction of the population that lives in vertex v [7]. Given any two points x and y in T that may be vertices or may belong to the interior of an edge, we denote by $d(x, y)$ the length of the unique path P_{xy} from x to y. Given a path P_{xy}, the sum of the weighted distances from all the vertices of T to P_{xy} is:

$$D(P_{xy}) = \sum_{v \in V} w(v) d(v, P_{xy})$$

where $d(v, P_{xy})$ is the distance from a vertex v to the path P_{xy}. Denote by $VAR(P_{xy})$ the variance of a path P_{xy} which is defined:

$$VAR(P_{xy}) = \sum_{v \in V} w(v)(d(v, P_{xy}) - D(P_{xy}))^2.$$

By referring to the well-known variance decomposition formula [7] we can re-write $VAR(P_{xy})$ as follows

$$VAR(P_{xy}) = \sum_{v \in V} w(v) d(v, P_{xy})^2 - D(P_{xy})^2.$$

If we denote by $Q(P_{xy}) = \sum_{v \in V} w(v) d(v, P_{xy})^2$ the sum of the weighted squared distances, the above formula becomes:

$$VAR(P_{xy}) = Q(P_{xy}) - D(P_{xy})^2.$$

The *Continuous Path-Variance Problem* consists of finding the path $P_{x^*y^*}$ between two points x^* and y^* in T such that $VAR(P_{x^*y^*}) \leq VAR(P_{xy})$, $\forall x, y \in T$. Similarly, the corresponding *Continuous Point-Variance Problem* on T can be formulated as the problem of finding the point x^* in T such that $VAR(x^*) \leq VAR(x)$, $\forall x \in T$. The following result shows that the point in T of minimum variance is not necessarily contained in $P_{x^*y^*}$.

Remark 1. Given a tree T, a point $x \in T$ that minimizes the variance objective function is not necessarily contained in a path of minimum variance.

Let us consider the tree in Figure 1. This is the same tree depicted in [7] but with the weight assigned to the edge $(4, 3)$ equal to 10 in place of 4. We assign weights equal to $\frac{1}{8}$ to each vertex. By applying the linear time algorithm presented in [7], the point x^* of minimum variance in T is located on the edge $(4, 3)$ at a distance $d(x^*, 3) = 8.214$. The value of the objective function at x^* is $VAR(x^*) = 14.214$. The path that minimizes the variance among all the paths containing the point x^* is the one whose end points are located at a distance equal to 2 to vertex 5 and equal to 2 to vertex 3, and such that its variance is equal to 11.750. The path that minimizes the variance in T is the path P_{15} with $VAR(P_{15}) = 10.937$ This path does not contain the edge $(4, 3)$ and, consequently, it does not contain the point x^* of minimum variance.

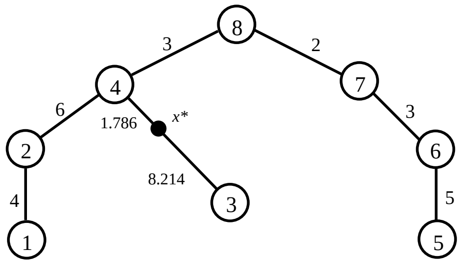

Figure 1: An example of a tree where the path of minimum variance does not contain the minimum variance point.

In our algorithm, for each edge we compute all the paths in T that minimize the variance and having a starting point in the given edge. In order to do this, consider an edge (r, s), and let T_r and T_s be the two subtrees obtained by removing (r, s) from T. First suppose to root T at vertex s and let $T(s) = T_r \cup T_s \cup (r, s)$ be the resulting rooted tree. By visiting top down level-by-level the subtree $T_r \subset T(s)$, for each edge $(v, u) \in T_r$ we find the path P_{xy} that minimizes the variance function with $x \in (r, s)$ and $y \in (v, u)$. Then, we consider r as the new root of T and, by visiting $T_s \subset T(r)$ top down, for each edge $(v, u) \in T_s$ we find the path P_{xy} that minimizes $VAR(P_{xy})$ with $x \in (r, s)$ and $y \in (v, u)$.

Suppose we are given a path $P_{\bar{x}\bar{y}}$ where $\bar{x}$ belongs to the edge (r, s) and $\bar{y}$ to the edge $(v, u) \in T_r$ (see Figure 2). In a rooted tree we may always assume that v is closer to the root s than u in $T(s)$. For convenience, we denote by $x \in (r, s)$ ($y \in (v, u)$) both the point along the edge and the distance of the point from r (from v). Given a path $P_{\bar{x}\bar{y}}$ we have to compute the squared sum of the weighted distances $D(P_{xy})^2$ of a new path $P_{xy} = P_{\bar{y}y} \cup P_{\bar{x}\bar{y}} \cup P_{\bar{x}x}$ with $x \in (r, s)$ and y belonging to an edge $(u, m) \in T_u \subseteq T_r$ adjacent to the edge (v, u). $D(P_{xy})^2$ can be computed as follows:

$$D(P_{xy})^2 = [D(P_{\bar{x}\bar{y}}) - Sav(P_{\bar{y}y}) - Sav(P_{\bar{x}x})]^2$$

where $Sav(P_{\bar{y}y})$ and $Sav(P_{\bar{x}x})$ are the reductions (*savings*) in the sum of the weighted distances $D(P_{\bar{x}\bar{y}})$ obtained by attaching to the path $P_{\bar{x}\bar{y}}$ the two paths $P_{\bar{y}y}$ and $P_{\bar{x}x}$, respectively [11, 12]. In a rooted tree T_r, we denote by $W(T_v) = \sum_{k \in T_v} w(k)$ the sum of the weights of the vertices k belonging to a subtree $T_v \subseteq T_r$ and by $D(P_{\bar{x}\bar{y}}; T_v)$ the sum of the distances of all the vertices $k \in T_v \subseteq T_r$ from the path $P_{\bar{x}\bar{y}}$. We have:

$$Sav(P_{\bar{y}y}) = d(\bar{y}, u)W(T_u \backslash T_m) + d(\bar{y}, y)W(T_m).$$

Since by definition $d(\bar{y}, y) = d(\bar{y}, u) + y$, and $W(T_u \backslash T_m) + W(T_m) = W(T_u)$, $Sav(P_{\bar{y}y})$ becomes:

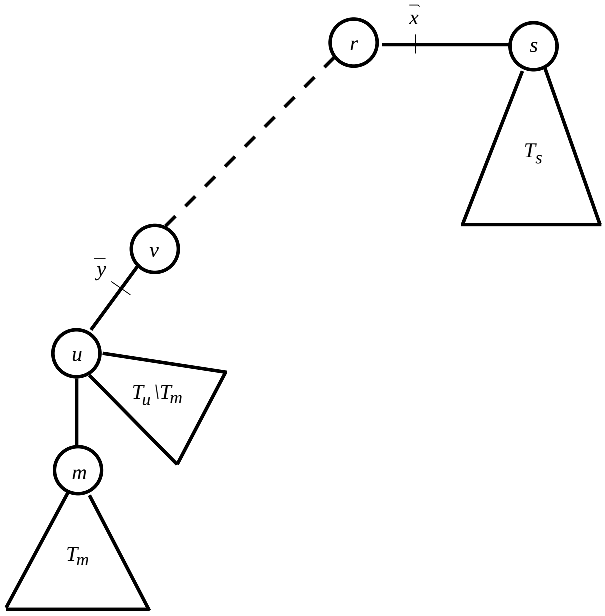

Figure 2: Basic elements in the construction of P_{xy} when visiting $T_r \subset T(s)$.

$$Sav(P_{\bar{y}y}) = d(\bar{y}, u)W(T_u) + yW(T_m). \tag{1}$$

Similarly, to compute $Sav(P_{\bar{x}x})$ we have:

$$Sav(P_{\bar{x}x}) = (x - \bar{x})W(T_s). \tag{2}$$

We note that $Sav(P_{\bar{x}x})$ may be positive or negative depending on the position on the edge (r, s) of the new point x with respect to the old point $\bar{x}$. Hence, the squared sum of the weighted distances $D(P_{xy})^2$ of the path P_{xy} can be expressed as a quadratic function of the two (unknown) points y and x as follows:

$$D(P_{xy})^2 = [D(P_{\bar{x}\bar{y}}) - d(\bar{y}, u)W(T_u) - yW(T_m) - (x - \bar{x})W(T_s)]^2. \tag{3}$$

Note that in (3) $D(P_{\bar{x}\bar{y}}) - d(\bar{y}, u)W(T_u)$ is constant. Moreover, $D(P_{\bar{x}\bar{y}})$ is the current value of the sum of the distances of the path $P_{\bar{x}\bar{y}}$, while $d(\bar{y}, u) = \ell_{vu} - \bar{y}$. The value $W(T_v)$ can be computed for all the vertices $v \in T(s)$ in linear time by visiting bottom up level-by-level the rooted tree $T(s)$ in a preprocessing phase [2].

Let us now consider the savings in the sum of the weighted squared distances $Q(P_{\bar{x}\bar{y}})$ when we consider the new path $P_{xy} = P_{\bar{y}y} \cup P_{\bar{x}\bar{y}} \cup P_{\bar{x}x}$ with y belonging to an edge $(u, m) \in T_u$ adjacent to (v, u), and x belonging to (r, s). As before, only the vertices in the two subtrees $T_u \subseteq T_r$ and T_s are involved in the computation of the savings. In particular, let us consider the saving obtained when the path $P_{\bar{y}y}$ is added to $P_{\bar{x}\bar{y}}$. By considering the vertices in T_u we have:

$$QSav(P_{\bar{y}y}) = \sum_{k \in T_u \setminus T_m} w(k)[d(k, \bar{y})^2 - d(k, u)^2] + \sum_{k \in T_m} w(k)[d(k, \bar{y})^2 - d(k, y)^2].$$

Since $d(k,\bar{y}) = d(k,u) + d(u,\bar{y})$ for all the vertices $k \in T_u \backslash T_m$, and $d(k,\bar{y}) = d(k,y) + d(\bar{y},y)$ for all the vertices $k \in T_m$, by substituting in the above formula we obtain:

$$\begin{aligned} QSav(P_{\bar{y}y}) &= d(u,\bar{y})^2 W(T_u \backslash T_m) + 2d(u,\bar{y}) D(u; T_u \backslash T_m) \\ &\quad + (d(u,\bar{y}) + y)^2 W(T_m) + 2(d(u,\bar{y}) + y) D(y; T_m) \end{aligned} \tag{4}$$

with $D(y;T_m) = D(m;T_m) + (\ell_{um} - y)W(T_m)$.

Let us now consider the saving in the sum of the weighted squared distances of the path $P_{\bar{x}x}$. In this case only the vertices in the subtree T_s are involved in the computation. Then we have

$$QSav(P_{\bar{x}x}) = \sum_{k \in T_s} w(k)[d(k,\bar{x})^2 - d(k,x)^2].$$

Since for all the vertices $k \in T_s$ we have $d(k,\bar{x}) = d(k,x) + (x - \bar{x})$, and $\sum_{k \in T_s} w(k) d(k,\bar{x}) = D(\bar{x}; T_s)$, we have:

$$QSav(P_{\bar{x}x}) = (x - \bar{x})^2 W(T_s) + 2(x - \bar{x}) D(x; T_s). \tag{5}$$

with $D(x;T_s) = D(s;T_s) + (\ell_{rs} - x)W(T_s)$. Even in this case $QSav(P_{\bar{x}x})$ can be positive or negative according to the position of the point $\bar{x}$ with respect to x. The sum of the weighted squared distances of the new path P_{xy} as a function of the two variables x and y is:

$$\begin{aligned} Q(P_{xy}) &= Q(P_{\bar{x}\bar{y}}) - d(u,\bar{y})^2 W(T_u \backslash T_m) - 2d(u,\bar{y}) D(u; T_u \backslash T_m) \\ &\quad - (d(u,\bar{y}) + y)^2 W(T_m) - 2(d(u,\bar{y}) + y) D(y; T_m) \\ &\quad - (x - \bar{x})^2 W(T_s) - 2(x - \bar{x}) D(x; T_s). \end{aligned} \tag{6}$$

From (3) and (6) follows that the variance of the path P_{xy} can be written as a function $\phi(x,y)$ of the two variables x and y:

$$VAR(P_{xy}) = \phi(x,y) = Q(P_{xy}) - D(P_{xy})^2. \tag{7}$$

As proved in [5], the variance is a strictly convex function on the compact set $[0,\ell_{rs}] \times [0,\ell_{um}]$. Thus, the two points $y \in (u,m)$ and $x \in (r,s)$ can be computed by solving the following system of linear equations:

$$\begin{cases} \frac{\partial \phi(x,y)}{\partial x} = 0 \\ \frac{\partial \phi(x,y)}{\partial y} = 0 \end{cases} \tag{8}$$

Note that when the points x and/or y do not belong to the above compact set, they belong to the boundary of the set $[0,\ell_{rs}]\times[0,\ell_{um}]$. Hence, similar cases described in [5] must be considered and, in particular, we have:

$$\begin{cases} x = \ell_{rs} & \text{if } x \geq \ell_{rs} \\ x = 0 & \text{if } x \leq 0 \\ y = \ell_{um} & \text{if } y \geq \ell_{um} \\ y = 0 & \text{if } y \leq 0 \end{cases} \tag{9}$$

For a given pair of edges (r,s) and (v,u) we define a *best* path $P_{\bar{x}\bar{y}}(v,u)$ the path of minimum variance among all the paths P_{xy} with $x \in (r,s)$ and $y \in (v,u)$. Given an edge (r,s), with the previous formulas we are able to compute the best path $P_{\bar{x}\bar{y}}(v,u)$ for each possible edge (v,u) of a given rooted tree.
In order to solve the Continuous Path-Variance problem on a tree T, we must also consider the case in which the optimal path lies within a single edge, including the special case when the optimal path is a point. For this case we can simply refer to the procedure provided in [5].

In the following we sketch a pseudo-code of the algorithm for solving the Continuous Path-Variance problem on a tree T.

CONTINUOUS PATH-VARIANCE

Input: A weighted tree T.
Output: A continuous path P^* of minimum variance $VAR(P^*)$ in T.

1. Solve the path-variance problem for each edge e of T (see [5])
   and let $P(e)$ be the best path on e and $Var(P(e))$ the corresponding variance
2. **for each** edge $e = (a,b) \in T$
3.   Let $P^*_e = P_{\bar{x}\bar{y}}(a,b) = P(e)$, with $\bar{x}, \bar{y} \in e$
4.   **repeat** steps 5-13 for $r = a$ and $s = b$ and for $r = b$ and $s = a$
5.   root the tree T at vertex s and let $T(s)$ be the resulting rooted tree
6.   visit $T(s)$ bottom up and compute all the information at all the vertices $v \in V$
7.   visit top down level-by-level the subtree rooted at r and let T_r be the rooted tree
8.   **for each** edge $(u,m) \in T_r$
9.   find $x \in (r,s)$ and $y \in (u,m)$ by solving (8) and $VAR(P_{xy})$ by using (7)
10.   let $P_{\bar{x}\bar{y}}(u,m) = P_{xy}$
11.   **if** $VAR(P_{\bar{x}\bar{y}}(u,m)) < VAR(P^*_e)$ **then**
12. $P^*_e = P_{\bar{x}\bar{y}}(u,m)$
13. $VAR(P^*_e) = VAR(P_{\bar{x}\bar{y}}(u,m))$
14. Let $P^* = \text{argmin}_e\{VAR(P^*_e)\}$
15. **output** P^* and $VAR(P^*)$

Lemma 1. The CONTINUOUS PATH-VARIANCE algorithm finds the path P^* of minimum variance in T.

Proof. It is clear that an optimal path exists in T. Indeed, it is either included in a single edge or its end points are contained in two different edges of T. In the first case, the optimal path is found at step 1 of the above algorithm by the procedure given in [5]. In the second case, we can compute the variance of a path P_{xy}, $x \in (r,s)$ and $y \in (u,m)$, through formula (7) where the function $\phi(x,y)$ is updated by the saving functions $Sav(\cdot)$ and $QSav(\cdot)$. In both these savings the quantities that are function of x are independent from those that are function of y. Thus, given a path $P_{\bar{x}\bar{y}}$, for $x \in (r,s)$ and $y \in (u,m)$, $Var(P_{xy})$ can be correctly updated from $Var(P_{\bar{x}\bar{y}})$ by computing $Sav(\cdot)$ and $QSav(\cdot)$ with respect to the vertices in the two subtrees $T_u \subseteq T_r$ and T_s, separately. Since the variance function is strictly convex on the compact set $[0,\ell_{rs}]\times[0,\ell_{um}]$ (see [5]), a path that minimizes $\phi(x,y)$ among all the paths P_{xy} with $x \in (r,s)$ and $y \in (u,m)$ can be correctly found by solving (8).

Since our algorithm computes the best path $P_{\bar{x}\bar{y}}$ for all the possible pairs of edges of a given tree T, it correctly finds a path P^* of minimum variance in T. $\square$

Lemma 2. The CONTINUOUS PATH-VARIANCE algorithm finds the path P^* of minimum variance in $O(n^2)$ time.

Proof. Given a tree $T(s)$ rooted at a vertex s, and for each subtree $T_u \subseteq T(s)$, all the quantities we use in our algorithm, that is, $W(T_u)$, $D(u;T_u)$, $D(u;T_u)^2$ can be computed once in $O(n)$ time in a preprocessing phase. The path-variance problem restricted to an edge can be solved in linear time for all the edges in T. Starting from a path $P_{\bar{x}\bar{y}}$, the saving formulas $Sav(\cdot)$ and $QSav(\cdot)$ can be computed in constant time. The system given by (8) is linear in the two variables x and y and then it is solvable in constant time. Since the algorithm must be repeated for all the possible edges $(r,s) \in T$, the overall time complexity is $O(n^2)$. $\square$

3 The Discrete Path-Variance Problem

In this section we consider the *Discrete Path-Variance Problem* on a tree. Let $T = (V,E)$ be a tree with $|V| = n$. As before, we assume that a positive weight $w(v)$ is associated to each vertex $v \in V$ and a positive real length ℓ_{vu} is assigned to each edge $(v,u) \in E$. Here we adopt the same notation as before and consider the same assumptions over the weights $w(v)$, $v \in V$. The Discrete Path-Variance Problem on a tree T consists of finding the path $P_{v^*u^*}$ between two vertices v^* and u^* in V such that $VAR(P_{v^*u^*}) \leq VAR(P_{vu})$, $\forall v,u \in V$. Similarly, the corresponding *Discrete Vertex-Variance Problem* on T can be formulated as the problem of finding the vertex $v^* \in V$ such that $VAR(v^*) \leq VAR(v)$, $\forall v \in V$. In the following, if there is no possibility of confusion we omit the end points of a path and we simply write P in place of P_{vu}.

In order to solve the Discrete Path-Variance Problem, we provide an $O(n^2)$ algorithm based on quantities similar to those used in Section 2.

Recall that, given a path P, the sum of the weighted distances from all the vertices of T to P is denoted by

$$D(P) = \sum_{v \in V} w(v)d(v,P)$$

where $d(v,P)$ is the distance from a vertex v to its closest vertex lying in the path P. According to this, the variance of a path P is given by

$$VAR(P) = \sum_{v \in V} w(v)(d(v,P) - D(P))^2$$

and can always be decomposed into the difference between the weighted sum of the square distances and the square of the sum of the weighted distances as follows:

$$VAR(P) = \sum_{v \in V} w(v) d(v, P)^2 - D(P)^2 = Q(P) - D(P)^2.$$

Remark 2. Given a tree T, a vertex $v \in T$ that minimizes the variance objective function is not necessarily contained in a discrete path of minimum variance.

Let us consider the tree T in Figure 3. This is the same tree of Figure 1 but with an additional vertex (numbered by 9). Suppose that we associate to the vertices 1-8 a weight equal to $\frac{1}{(8+\epsilon)}$ and to the vertex 9 a weight equal to $\frac{\epsilon}{(8+\epsilon)}$, $\epsilon > 0$. If we choose $\epsilon = 0.1$ the vertex that minimizes the variance is the new vertex 9 with value $VAR(9) = 14.861$. The path that minimizes the variance among all the discrete paths that contain vertex 9 is P_{35}, with value $VAR(P_{35}) = 12.888$, but the path that minimizes the variance in T is P_{15} with value $VAR(P_{15}) = 10.806$.

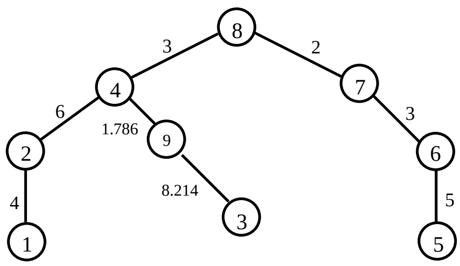

Figure 3: An example of a tree where the discrete path of minimum variance does not contain the vertex of minimum variance.

Remark 3 A discrete path P in T that minimizes $VAR(P)$ does not necessarily have its end vertices in the leaves of T.

Consider the tree in Figure 4 with 9 vertices, with all edge lengths equal to 1 and all the weights of the vertices equal to $\frac{1}{9}$. The path P from vertex 1 to vertex 3 is the one that minimizes the variance objective function.

In our algorithm, given a vertex r, we root the tree at r and compute the variance of all the paths in T having a starting point in r. Then, by rooting T at each vertex $v \in V$ we are able to find a path $P_{v^*u^*}$ of minimum variance among all paths P_{vu}, $v, u \in V$.

Suppose that the tree T is rooted at a vertex $r \in V$, and let T_r be the resulting rooted tree. Let T_v be a subtree of T_r rooted at vertex v. Given a path P starting from r to a vertex v, the squared sum of the weighted distances of a path $P' = P \cup (v, u)$, where u belongs to the set $S(v)$ of the children of v in T_v can be computed as follows:

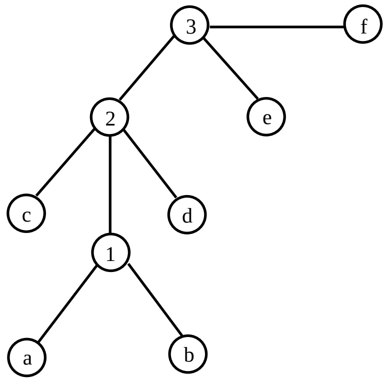

Figure 4: The path that minimizes the variance is P_{13}.

$$D(P')^2 = [D(P) - Sav(v,u)]^2$$

where $Sav(v,u)$ is the saving in the sum of the weighted distances $D(P)$ obtained by attaching to the path P the edge $(v,u) \in T_v$ [11, 12]. We have:

$$Sav(v,u) = \ell_{vu} W(T_u)$$

where $W(T_u)$ is the sum of the weights of all the vertices in T_u. By simple computations, we have:

$$D(P')^2 = [D(P) - Sav(v,u)]^2 = D(P)^2 + [\ell_{vu} W(T_u)]^2 - 2\ell_{vu} W(T_u) D(P). \qquad (10)$$

On the basis of the above formulas, the saving in the squared sum of the distances is given by

$$D(P)^2 - D(P')^2 = 2\ell_{vu} W(T_u) D(P) - [\ell_{vu} W(T_u)]^2. \qquad (11)$$

Let us now consider the saving in the weighted sum of the squared distances when an edge $(v,u) \in T_v$ is attached to the path P obtaining a new path $P' = P \cup (v,u)$. The weighted sum of the squared distances to path P' can always be written as

$$Q(P') = \sum_{k \notin T_v} w(k) d(k,P')^2 + \sum_{k \in T_v \setminus T_u} w(k) d(k,v)^2 + \sum_{k \in T_u} w(k) d(k,u)^2$$

while, for P, it can be written:

$$Q(P) = \sum_{k \notin T_v} w(k) d(k,P)^2 + \sum_{k \in T_v \setminus T_u} w(k) d(k,v)^2 + \sum_{k \in T_u} w(k) d(k,v)^2$$

and $\sum_{k \notin T_v} w(k) d(k,P')^2 = \sum_{k \notin T_v} w(k) d(k,P)^2$. Then, we have:

$$QSav(v,u) = Q(P) - Q(P') = \sum_{k \in T_u} w(k) d(k,v)^2 - \sum_{k \in T_u} w(k) d(k,u)^2.$$

Notice that the quantity $\sum_{k\in T_u} w(k)d(k,v)^2$ can be decomposed as follows

$$\sum_{k\in T_u} w(k)(d(k,u)+\ell_{vu})^2 = \sum_{k\in T_u} w(k)d(k,u)^2 + \sum_{k\in T_u} w(k)\ell_{vu}^2 + 2\ell_{vu}\sum_{k\in T_u} w(k)d(k,u).$$

It follows that the saving in the weighted sum of the squared distances $QSav(v,u)$ is given by

$$QSav(v,u) = Q(P) - Q(P') = \ell_{vu}^2 W(T_u) + 2\ell_{vu} D(u;T_u) \tag{12}$$

Given a path P from the root r to a vertex v, we denote by $TSav(v,u)$ the total saving in $VAR(P)$ obtained by attaching to P the edge $(v,u)\in T_v$. On the basis of formulas (11) and (12), we have:

$$\begin{aligned} TSav(v,u) &= VAR(P) - VAR(P') = (Q(P)-D(P)^2) - (Q(P')-D(P')^2) \\ &= (Q(P)-Q(P')) - (D(P)^2 - D(P')^2) \\ &= \ell_{vu}^2 W(T_u) + 2\ell_{vu}D(u;T_u) - (2\ell_{vu}W(T_u)D(P) - [\ell_{vu}W(T_u)]^2). \end{aligned} \tag{13}$$

Given the rooted tree T_r, start with $P=\{r\}$. The value $VAR(r)$ can be easily computed on the basis of the quantities associated at vertex r in a preprocessing phase. Visit T_r top down level-by-level; at a given vertex v, for each edge $(v,u)\in S(v)$ in T_v compute

$$VAR(P\cup(v,u)) = VAR(P) - TSav(v,u).$$

In this way in a rooted tree T_r we can compute the variance function for all the paths P_{rv} from the root r to each other v in T_r by the algorithm described below.

DISCRETE ROOTED PATH-VARIANCE

Input: A tree T_r rooted at a vertex r.
Output: A discrete path P_r^* of minimum variance among all the paths starting from r.

1. Let $P_r^* = \{r\}$ with $Var(P_r^*) = Var(r)$
2. **visit** T_r top down level-by-level
3. **for each** vertex $v\in T_r$
3. **for each** vertex $u\in S(v)$
4. let $P_{ru} = P_{rv}\cup(v,u)$ and $VAR(P_{ru}) = VAR(P_{rv}) - TSav(v,u)$
5. **if** $VAR(P_{ru}) < VAR(P_r^*)$ **then**
7. $P_r^* = P_{ru}$
6. $VAR(P_r^*) = VAR(P_{ru})$
8. **output** P_r^*

Once we have computed the variance function for all the paths P_{rv} from the root r to each other vertex v in T_r, we choose the one of minimum variance, say P_r^*. Then, among all the P_r^* for all the possible $r\in V$, we choose a path $P^*\in T$ that minimizes the variance function.

Lemma 3. The algorithm finds a discrete path P^* in T that minimizes $VAR(P^*)$ in $O(n^2)$ time.

Proof. Given a rooted tree T_r, we can compute $VAR(r)$ in $O(n)$ time. All the other quantities needed in the computation of the function $TSav(\cdot)$ can be computed once $\forall(v,u) \in T_r$ in linear time in a preprocessing phase. Since the above algorithm must be repeated by rooting the tree to all the vertices of T, the overall time complexity for finding a path of minimum variance in T is $O(n^2)$.

One may think that by a suitable tree decomposition rule, a dynamic programming approach can be adopted to solve the Discrete Path-Variance Problem on a tree in order to obtain a better overall time complexity algorithm. Nevertheless, it seems that here this approach doesn't work. Actually, we were not able to find recursive formulas because, given a tree rooted at a vertex v, it is not possible to obtain a best path passing through or starting from a vertex v by composing a best path starting from one of its sons u with the edge (v,u). For example, consider the tree in Figure 1. Suppose to root T at vertex 8, a best path starting from 8 is the path P_{18}, but the best path starting from vertex 4 is P_{34}.

4 The Path-Variance Problem with Bounded Length

In this section we consider the discrete and continuous path variance problems with an additional constraint on the length of the path. Given a path P we denote by $L(P)$ the length of P. The two problems can be stated as follows:

Continuous Path-Variance Problem with Bounded Length: given a bound $B > 0$, find the path $P_{x^*y^*}$ between two points x^* and y^* in T with $L(P_{x^*y^*}) \leq B$ such that $VAR(P_{x^*y^*}) \leq VAR(P_{xy})$ $\forall\, x, y \in T$ and $L(P_{xy}) \leq B$.
Discrete Path-Variance Problem with Bounded Length: given a bound $B > 0$, find the path $P_{v^*u^*}$ between two vertices v^* and u^* in V with $L(P_{v^*u^*}) \leq B$ such that $VAR(P_{v^*u^*}) \leq VAR(P_{vu})$ $\forall\, v, u \in V$ and $L(P_{vu}) \leq B$.

A path P is *feasible* if $L(P) \leq B$. Again we can solve these two problems in $O(n^2)$ time by using the same quantities introduced in the previous sections. In particular, for the discrete case, since our algorithm considers the paths between all pairs of vertices in T, the length constraint can be taken into account simply by checking the length of the path found at each step of the algorithm. In this way, we may avoid to visit a subtree T_u of the rooted tree, whenever extending a path P by adding a new edge (v,u) we have $L(P \cup (v,u)) > B$.

A similar stopping rule can be applied also in the continuous case. Indeed, recall that, at a given step of the algorithm, we have a path $P_{\bar{x}\bar{y}}$ with $\bar{x}$ belonging to a fixed edge (r,s) and $\bar{y}$ to an edge $(v,u) \in T_r$. Assume that $L(P_{\bar{x}\bar{y}}) \leq B$, we must find a new path P_{xy}, with $x \in (r,s)$ and y belonging to an edge $(u,m) \in T_u \subseteq T_r$ adjacent to (v,u), such that $L(P_{xy}) \leq B$, and which minimizes the variance. In order to do that, let us consider the path P_{ru}, if $L(P_{ru}) \geq B$ we may avoid visiting the subtree T_u. When $L(P_{ru}) < B$ we can find the new path P_{xy} by solving the following quadratic programming problem:

$$
\begin{array}{ll}
\text{Min} & \phi(x, y) \\
\text{s.t.} & \\
& x + y \leq B' \\
& 0 \leq x \leq \ell_{rs} \\
& 0 \leq y \leq \ell_{um}
\end{array}
\tag{14}
$$

where $B' = B - L(P_{ru})$ and $\phi(x, y)$ is defined by formula (7). In the following, we refer to the optimal solution of this problem by $(\tilde{x}, \tilde{y})$.

In order to solve the Continuous Path-Variance Problem with Bounded Length on a tree T, we compute the best feasible paths for all the possible pairs of edges of T. However, we must also consider the case in which the optimal path lies within a single edge. For this case we refer to the notation used in [5], and for each edge $(r, s) \in T$, we solve the following quadratic programming problem:

$$
\begin{array}{ll}
\text{Min} & VAR(P_{z_1 z_2}) \\
\text{s.t.} & z_1 + z_2 \leq \ell_{rs} \\
& z_1 + z_2 \geq \ell_{rs} - B \\
& z_1, z_2 \geq 0
\end{array}
\tag{15}
$$

where, according to [5], $VAR(P_{z_1 z_2})$ is given by

$$
\begin{aligned}
VAR(P_{z_1 z_2}) \quad = \quad & W(T_r)W(T_s)z_1^2 + W(T_r)W(T_s)z_2^2 \\
& -2W(T_r)W(T_s)z_1 z_2 + 2\{D(r;T_r) - W(T_r)D(P_{rs})\}z_1 \\
& +2\{D(s;T_s) - W(T_s)D(P_{rs})\}z_2 + VAR(P_{rs}) = W(T_r)W(T_s)[z_1 - z_2]^2 \\
& +2\{D(r;T_r)W(T_s) - D(s;T_s)W(T_s)\}[z_1 - z_2] + VAR(P_{rs})
\end{aligned}
$$

and z_1 and z_2 in (r, s) denote both the points along the edge and the distance from r and s, respectively.

In the following we sketch a pseudo-code of the algorithm for solving the Continuous Path-Variance problem with Bounded Length on a tree T.

BOUNDED CONTINUOUS PATH-VARIANCE

Input: A weighted tree T and a bound $B > 0$.
Output: A continuous path P^* of minimum variance $VAR(P^*)$ in T with $L(P^*) \leq B$.

1. Solve the path-variance problem for each edge e of T by (15)
   and let $P(e) = P_{z_1 z_2}$ be the best feasible path on e with $Var(P(e)) = Var(P_{z_1 z_2})$
2. **for each** edge $e = (a, b) \in T$
3.   Let $P_e^* = P_{\bar{x}\bar{y}}(a, b) = P(e)$, with $\bar{x}, \bar{y} \in e$
4.   **repeat** steps 5-20 for $r = a$ and $s = b$ and for $r = b$ and $s = a$
5.   root the tree T at vertex s and let $T(s)$ be the resulting rooted tree
6.   visit $T(s)$ bottom up and compute all the information at all the vertices $v \in V$
7.   visit top down level-by-level the subtree rooted at r and let T_r be the rooted tree
8.   **for each** edge $(u, m) \in T_r$
9.   **if** $u \equiv r$
10.   solve (14) and let $\bar{x} = \tilde{x}$, $\bar{y} = \tilde{y}$ and $VAR(P_{\bar{x}\bar{y}}(u, m)) = \phi(\tilde{x}, \tilde{y})$
11.   **if** $VAR(P_{\bar{x}\bar{y}}(u, m)) < VAR(P_e^*)$ **then**
12. $P_e^* = P_{\bar{x}\bar{y}}(u, m)$
13. $VAR(P_e^*) = VAR(P_{\bar{x}\bar{y}}(u, m))$
14.   **else**
15.   **if** $L(P_{ru}) \geq B$ STOP visiting T_u
16.   **else**
17.   solve (14) and let $\bar{x} = \tilde{x}$, $\bar{y} = \tilde{y}$ and $VAR(P_{\bar{x}\bar{y}}(u, m)) = \phi(\tilde{x}, \tilde{y})$
18.   **if** $VAR(P_{\bar{x}\bar{y}}(u, m)) < VAR(P_e^*)$ **then**
19. $P_e^* = P_{\bar{x}\bar{y}}(u, m)$
20. $VAR(P_e^*) = VAR(P_{\bar{x}\bar{y}}(u, m))$
21. Let $P^* = \text{argmin}_e\{VAR(P_e^*)\}$
22. **output** P^* and $VAR(P^*)$

The above algorithm finds a path P^* in T of minimum variance $VAR(P^*)$ with $L(P^*) \leq B$ in $O(n^2)$. It is easy to check that it has the same time complexity of the CONTINUOUS PATH-VARIANCE algorithm since solving the two quadratic programs (14) and (15) requires constant time.

5 Concluding Remarks

In this paper we considered the Path-Variance problem on trees both in the continuous and in the discrete case. For the two problems we provided two $O(n^2)$ time algorithms. In particular, the algorithm for the continuous Path-Variance problem improves upon a $\log n$ term the previous algorithm presented in [5]. We showed that the path that minimizes the variance function on a tree T does not necessarily contain the point of minimum variance in T. We also showed that this nestedness property does not hold even for the vertex that minimizes the variance in T in the discrete case. The common idea of the two algorithms is that, by visiting top down level-by-level a rooted tree, we can efficiently compute all the paths that minimize the variance among those having a starting point in a given edge in the continuous case, or at a given vertex in the discrete case. The computation of the variance for all these paths is done by introducing saving functions. Exploiting the decomposition of the variance formula, these functions allow to update the variance objective function when a new path P' is obtained from a given path P efficiently. We extended our algorithms to the case with an additional constraint on the length of the path, and we maintain the same overall time complexity of $O(n^2)$.

As discussed in the Introduction, the variance function is a common measure of equity in location theory which is often used when the dispersion of the clients' demand with respect to a facility must be considered. Another interesting dispersion measure is given by the *coefficient of variation* which has the nice property of being independent from scaling factors. We noticed that we can use our algorithms to solve the problem of finding a continuous or discrete path in a tree which minimizes the coefficient of variation, too. In particular, for the continuous case, in [4] is proved that, given two edges (r, s) and (v, u), the coefficient of variation, expressed as a function of the variables $x \in (0, \ell_{rs})$ and $y \in (0, \ell_{vu})$, is a pseudo-convex function with respect to the open set $(0, \ell_{rs}) \times (0, \ell_{vu})$. This suffices for applying our algorithms also for these problems.

References

[1] I. Averbakh, O. Berman (1999). Algorithms for path medi-centers of a tree, *Computer & Operations Research*, 26, 1395-1409.

[2] R. I. Becker, Y. Chiang, I. Lari, A. Scozzari, G. Storchi (2002). Finding the L-core of a Tree, *Discrete Applied Mathematics* 118, 25-42.

[3] R. I. Becker, I. Lari, A. Scozzari. Algorithms for Central-Median paths with Bounded Length on Trees, *European Journal of Operational Research*, forthcoming.

[4] T. Cáceres (2001). Localizacion con Criterios de Igualidad, *Ph.D. Thesis* Departamento de Matematica Aplicada I, Universidad de Sevilla.

[5] T. Cáceres, M.C. López-de-los-Mozos, J.A. Mesa (2004). The path-variance problem on tree networks, *Discrete Applied Mathematics*, 145, 72-79.

[6] S. L. Hakimi, E. F. Schmeichel, M. Labbé (1993). On Locating Path- or Tree- Shaped Facilities on Networks, *Networks*, 23, 543-555.

[7] O. Maimon (1986). The Variance Equity Measure in Locational Decision Theory, *Annals of Operations Research*, 6, 147-160.

[8] M. T. Marsh, D. A. Schilling (1994). Equity Measurement in Facility Location Analysis: a Review and Framework, *European Journal of Operational Research*, 74, 117.

[9] B. Boffey, J. A. Mesa (1996). A Review of Extensive Facility Location in Networks, *European Journal of Operational Research*, 95, 592-600.

[10] J. A. Mesa, J. Puerto, A. Tamir (2003). Improved Algorithms for Several Network Location Problems with Equality Measures, *Discrete Applied Mathematics*, 130, 437-448.

[11] C. A. Morgan, J. P. Slater (1980). A Linear Algorithm for a Core of a Tree, *Journal of Algorithms*, 1, 247-258.

[12] S. Peng, A. B. Stephens, Y. Yesha (1993). Algorithms for a Core and a k-tree Core of a Tree, *Journal of Algorithms*, 15, 143-159.

[13] J. Puerto, F. Ricca, A. Scozzari (2005). Extensive Facility Location Problems on Networks with Equity Measures, Technical Report del Dipartimanto di Matematica per le Decisioni Economiche, Finanziarie ed Assicurative, anno XII, n. 4.

[14] J. Puerto, A.M. Rodríguez-Chía, A. Tamir, D. Pérez. The Bi-Criteria Doubly Weighted Center-Median Path Problem On A Tree, *Networks*, to appear.

[15] J. Puerto, A. Tamir (2005). Locating Tree-shaped Facilities Using the Ordered Median Objective, *Mathematical Programming A*, 102, 313-338.

[16] A. Tamir, J. Puerto, D. Pérez-Brito (2002). The Centdian Subtree on Tree Networks, *Discrete Applied Mathematics*, 118, 263-278.

[17] A. Shioura, M. Shigeno (1997). The Tree Center Problems and the Relationship with the Bottleneck Knapsack Problems, *Networks*, 29, 107-110.

[18] A. Tamir, J. Puerto, J. A. Mesa, A. M. Rodríguez-Chía (2005). Conditional Location of Path and Tree Shaped Facilites on Trees, *Journal of Algorithms*, 56, 50-75.

[19] A. Tamir (1998). Fully Polynomial Approximation Schemes for Locating a Tree-shaped Facility: a Generalization of the Knapsack Problem, *Discrete Applied Mathematics*, 87, 229-243.