Moving Walkways, Escalators, and Elevators

Jean Cardinal *† Sébastien Collette *‡ Ferran Hurtado §
Stefan Langerman *¶ Belén Palop ||

Abstract

We study a simple geometric model of transportation facility that consists of two points between which the travel speed is high. This elementary definition can be used to model shuttle services, tunnels, bridges, teleportation devices, and moving walkways. Red-blue and unidirectional variants can also be used to model escalators and elevators. The travel time between a pair of points is defined as a time distance, in such a way that a customer uses the transportation facility only if it is helpful. We give algorithms for finding the optimal location of such a transportation facility, where optimality is defined with respect to the maximum travel time between two points in a given set.

1 Introduction

Facility location is an old, well-studied topic in computational geometry. One of the simplest example of facility location problem is finding the smallest disk covering a given set of points, which can be done in linear time, or its min-sum counterpart, the Fermat-Weber problem, which is difficult to solve exactly in usual models of computation. Surveys on both combinatorial and geometric facility location problems can be found in [16, 9, 10, 15].

The optimality of a location is often defined with respect to a distance measure. Recently, an interest has emerged in the definition and analysis of time distances, measuring the travel time between points in the plane given a geometric transportation facility, such as a highway or a geometric network made of line segments [1, 2, 5, 3, 13]. In general, the transportation facility is a subset of the plane where the travel speed is high, and the travel time is naturally defined as the minimum time needed to reach a point. Primary concerns in previous works on these notions include in particular efficient constructions of the Voronoi diagrams for these distances.

In this work, we consider optimal location of geometric transportation facilities on a line and in the plane. The transportation facilities we define are probably the simplest one can think about, and consist of a pair of points between which one can travel faster than anywhere else in the plane. This simple definition can be used to model for instance moving walkways in airports, shuttle connections or tunnels in a city. For simplicity, we refer to these as *moving walkways*. Given a set of points and a moving walkway, we define the travel time diameter as the maximum travel time between two points in the set. A moving walkway is said to be optimal if it minimizes the travel time diameter.

*Université Libre de Bruxelles (ULB). Supported by the Communauté française de Belgique - Actions de Recherche Concertées (ARC).
†jcardin@ulb.ac.be.
‡secollet@ulb.ac.be. Aspirant du FNRS.
§Ferran.Hurtado@upc.edu. Universitat Politècnica de Catalunya, Barcelona, Spain. Research of Ferran Hurtado partially supported by projects MEC MTM2006-01267 and Gen. Cat. 2005SGR00692.
¶slanger@ulb.ac.be. Chercheur Qualifié du FNRS.
||b.palop@infor.uva.es. Universidad de Valladolid, Spain. Research of Belén Palop partially supported by project VA031B06.

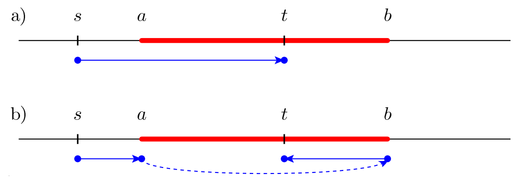

Figure 1: Moving walkways on a line. The travel time between two points is the minimum between a) the direct distance, b) the travel time using the walkway.

Moving walkways on a line are defined and analyzed in section 2. We first provide an $\Omega(n \log n)$ lower bound on the computation of the travel-time diameter for a given walkway in the algebraic computation tree model. Then we give a simple linear-time algorithm for finding the optimal moving walkway on a line.

In section 3, we define moving walkways in the plane, and provide an $O(n \log n)$ deterministic algorithm for the diameter. We show, using Chan's technique for implicit quasiconvex programs [8], that this implies an $O(n \log n)$ algorithm for finding an optimal horizontal walkway. We then derive an algorithm for computing a $(1+\varepsilon)$-approximation of an optimal moving walkway with arbitrary orientation and constant speed in time $O(\varepsilon^{-1} n \log n)$.

Related works. We are not aware of many previous contributions on the topic of optimal transportation facility location based on time distances.

In [7], we presented a family of geometric optimization problems in which the objective function was in a min-max-min form. As an example, we proposed the problem of finding the location of a moving walkway on a line such that the maximum travel time between n given pairs of source-destination points is minimized. The current paper presents extensions and generalizations of this result.

Recently, an effort has been made by several authors to solve the problem of locating a highway that minimizes the maximum travel time among a set of points [4]. In this work, a highway is modeled as a line in the plane on which the travel speed is higher, and customers can enter and exit the highway at any point on the line. In particular, they give an algorithm for the case where the travel speed on the highway is infinite, and another algorithm for the case where the highway is restricted to be vertical. The goal of this work is similar, but the transportation model that we study is different.

2 Moving walkways on a line

Suppose a new moving walkway is to be installed in a long corridor, for instance in the concourse of an airport. The moving walkway is used by customers to go from source to destination points in the corridor. We model the corridor as the real line, source and destination points as real numbers, and the moving walkway by an interval $[a, b]$ on the line. We denote by $v > 1$ the speed on the moving walkway, and assume that the speed outside the moving walkway is 1. Customers can only enter the walkway at point a, step down of it at point b, and always follow the shortest path. This is illustrated in figure 1.

The travel time between a point s of the line and any other point $t > s$, given a walkway $[a, b]$, is a time distance [2, 5], defined as follows:

$$d_{\text{time}}(s, t, a, b) = \min\{t - s, |s - a| + |t - b| + \frac{1}{v}(b - a)\}.$$

Note that by assuming $t > s$ we implicitly make the moving walkway bidirectional: for going from t to s, we enter the walkway at point b and exit at point a, which makes the travel time a symmetric distance function.

We consider the computation of the travel-time diameter and the optimal location of such a moving walkway.

2.1 Travel time diameter

A first problem is, given a set of points and a walkway $[a, b]$, to compute the travel time diameter, defined as the maximum travel time between two points of the set.

Problem 1. Given a set P of n real numbers, a real number $v > 1$, and a pair (a, b) of real numbers, find $\max_{s,t \in P : s \leq t} d_{\text{time}}(s, t, a, b)$.

Theorem 2.1. *The travel time diameter for a moving walkway on a line can be computed in $O(n \log n)$ time.*

The proof is omitted. The main point is to remark that computing the travel time diameter for points in the interval $[a, b]$ amounts to computing the angular diameter of points on a circle, since we can travel between any pair by following one of two paths, using the walkway or not.

Theorem 2.2. *There is an $\Omega(n \log n)$ lower bound on the computation of the travel time diameter for a moving walkway on a line in the algebraic computation tree model, even when $v = +\infty$.*

Proof. The proof of the lower bound is by reduction from the problem of set disjointness, consisting of checking whether $A \cap B = \emptyset$ for two sets of numbers A and B. It is similar to the proof of the same lower bound for the problem of computing the Euclidean diameter of a set of points.

We let $v = +\infty$, $a = 0$, $b = 2$ and scale the elements of A and B so that they are all strictly smaller than 1. We then let $P = A \cup \{x + 1 : x \in B\}$. Then the travel diameter of the points in P with respect to the moving walkway a, b is equal to 1 iff $A \cap B \neq \emptyset$. Note that since 1 is an upper bound on the travel time diameter, the lower bound also holds for the decision version of the problem. □

2.2 Minimum travel time diameter location

We now consider the following moving walkway location problem.

Problem 2. Given a set P of n real numbers, a real number $v > 1$, find a moving walkway $[a, b]$ of speed v that minimizes the travel time diameter of P: $\min_{a,b} \max_{s,t \in P : s \leq t} d_{\text{time}}(s, t, a, b)$.

There is a very simple deterministic algorithm solving this problem in $O(n)$ time. It is a surprising observation that we are able to find a moving walkway that minimizes the diameter faster than we are able to compute the diameter for a given walkway. This observation has already been made for the construction of optimal vertical highways [4]. We assume w.l.o.g. that $\min\{x \in P\} = 0$ and $\max\{x \in P\} = 1$, and denote by OPT the optimum value $\min_{a,b} \max_{s,t \in P : s \leq t} d_{\text{time}}(s, t, a, b)$.

Lemma 2.3. $OPT \leq \frac{v}{2v-1}$.

Proof. We define a moving walkway by $a = (1 - \frac{v}{2v-1})/2 = \frac{v-1}{4v-2}$ and $b = (\frac{v}{2v-1} + 1)/2 = \frac{3v-1}{4v-2}$. To go from a point p in the range $[0, 1 - \frac{v}{2v-1}]$ to a point q in the range $[\frac{v}{2v-1}, 1]$, the moving walkway is used and the diameter is at most $\frac{v}{2v-1}$. Otherwise we do not use the moving walkway; for every pair in $[0, \frac{v}{2v-1}]$, the time distance is bounded by $\frac{v}{2v-1}$. The same observation holds for every pair in $[1 - \frac{v}{2v-1}, 1]$. □

In what follows, we consider that we are given a set of points P and we try to find properties of its optimal moving walkway $[a, b]$. Let s be the smallest point in P such that, to go from 0 to s, it is better to use the moving walkway. We know that $s \leq b$, as to go from 0 to any point larger than or equal to b it is always better to use the moving walkway. Symmetrically, let r be the largest point in P such that to go from 1 to r, it is better to use the moving walkway; we know that $r \geq a$.

Lemma 2.4. *Let P be a set of points and $[a, b]$ an optimal moving walkway. The moving walkway is used only for pairs (p, q) where $p \in [0, r]$ and $q \in [s, 1]$. For every other pair, the facility is not needed.*

Proof. Suppose we have a pair (p, q) in $[0, s)$. As s is the smallest value such that using the moving walkway improves the distance, we know there is no need to use it to go from 0 to q, nor from 0 to p. So there is no need to use it for going from p to q. The same reasoning holds for pairs in $(r, 1]$. □

Lemma 2.5. *The optimal moving walkway is such that* $a = \frac{r}{2}$ *and* $b = \frac{s+1}{2}$.

Proof. We have not proved that *all* pairs (p, q) with $p \in [0, r]$ and $q \in [s, 1]$ require the use of the moving walkway. What we know however is that the moving walkway will be used for the pairs $(0, 1)$, $(0, s)$ and $(r, 1)$. Having $a = \frac{r}{2}$ and $b = \frac{s+1}{2}$ minimizes the maximum time distance for these pairs. Since the distance between these pairs are greater than all other distances using the walkway, this is optimal. □

This lemma showed that the optimal moving walkway is uniquely determined by the points r and s.

Lemma 2.6. *The point r is the largest point of P whose value is less than or equal to* $1 - \frac{v}{2v-1}$*; s is the smallest point of P whose value is more than or equal to* $\frac{v}{2v-1}$.

Proof. By Lemma 2.3 and 2.4, note that $r \leq 1 - \frac{v}{2v-1}$ and $s \geq \frac{v}{2v-1}$, otherwise we would need to use the moving walkway for the pairs $(0, r)$ or $(s, 1)$. By definition of s, to go from 0 to s it is better to use the moving walkway. This is true when

$$s \geq \frac{r}{2} + \frac{1}{v} \cdot \left(\frac{r}{2} + (s - r) + \frac{1-s}{2}\right) + \frac{1-s}{2} \geq \frac{v + vr - r + 1}{3v - 1}$$

By definition of s, it is the smallest value in P satisfying the constraint. Whatever the value of $r \leq 1 - \frac{v}{2v-1}$, the minimum value for s satisfying the constraint is $\frac{v}{2v-1}$. Symmetrically, for r we have

$$1 - r \geq \frac{r}{2} + \frac{1}{v} \cdot \left(\frac{r}{2} + (s - r) + \frac{1-s}{2}\right) + \frac{1-s}{2} \Leftrightarrow r \leq \frac{v + vs - s - 1}{3v - 1}$$

And by definition of r, it is the largest value in P satisfying the constraint, which for all value of $s \geq \frac{v}{2v-1}$ is $1 - \frac{v}{2v-1}$. □

Theorem 2.7. *Problem 2 can be solved in $O(n)$ deterministic time.*

Proof. By Lemma 2.6, we can find the points r and s of P in linear time. By Lemma 2.5, the optimal moving walkway is given by $a = \frac{r}{2}$ and $b = \frac{s+1}{2}$. □

3 Moving walkways in the plane

In this section, we generalize the previous moving walkway model to the plane. A moving walkway is defined by two points a and b, and customers can go from a point s to a point t either by going from s to t directly, or by entering the walkway at one of its endpoint and exiting at the other. The walkway is parameterized by a real number v so that the time to go from a to b is $\frac{1}{v}d(a, b)$. The distance measure $d(.,.)$ used in the definition of the travel time must be convex and satisfy the triangular inequality. For simplicity we assume it is the Euclidean distance, keeping in mind that all the following can be generalized to other distance measures. Note that the moving walkways are bidirectional, hence the travel time is a symmetric distance function:

$$d_{\text{time}}(s, t, a, b) = \min\{d(s, t), d(s, a) + \frac{1}{v}d(a, b) + d(b, t), d(s, b) + \frac{1}{v}d(a, b) + d(t, a)\}.$$

We first solve a location problem with n source-destination pairs, under the constraint that the moving walkway must be horizontal. Then we give an algorithm for computing the travel time diameter for a given moving walkway (not necessarily horizontal), and from this derive an algorithm for finding optimal horizontal walkways. We show that this algorithm can be used for obtaining an approximation algorithm for locating moving walkways with arbitrary orientation. We also outline other restricted location problems in the plane that we can solve in $O(n \log n)$ time.

3.1 Horizontal moving walkway location for n source-destination pairs

As a preliminary, we consider the problem of minimizing the maximum travel time between a number of designated source-destination pairs, with the constraint that the moving walkway must be horizontal. We let (p_x, p_y) denote the x and y-coordinates of a point p.

Problem 3. Given n pairs $\{(s_i, t_i)\}_{i=1}^n$ of points, and a real number $v > 1$, find a horizontal moving walkway $[a^*, b^*]$ of speed v minimizing the maximum travel time between a pair: $\min_{a,b:a_y=b_y} \max_i d_{\text{time}}(s_i, t_i, a, b)$,

The following lemma states that if one travels from left to right, then the moving walkway, if it is used, will be used from left to right.

Lemma 3.1. *Given a pair of points a, b such that $a_y = b_y$ and $a_x < b_x$, and a pair of points s, t such that $s_x < t_x$, we have: $d_{time}(s, t, a, b) = \min\{d(s, t), d(s, a) + \frac{1}{v}d(a, b) + d(b, t)\}$.*

Proof. As it is the case on the line, pairs of points s, t for which the walkway is useful are separated by the bisector of the segment $[a, b]$. Then we can see that if we use the walkway from b to a, using it from a to b can only reduce the travel time. □

Lemma 3.2. *Problem 3 is LP-type.*

Proof. We prove that the functions $f_i(a, b) = d_{\text{time}}(s_i, t_i, a, b)$ are quasiconvex: all the lower level sets $\{(a, b) : f_i(a, b) \leq y\}$ for any real number y are convex. From Amenta, Bern and Eppstein [6], this implies that the problem is LP-type. The domain of these functions is $\mathbb{R}^4$, however we only consider their restriction to the subspace of points a, b such that $a_x \leq b_x$ and $a_y = b_y$, for which quasiconvexity is preserved. We also assume, without loss of generality, that $s_{ix} \leq t_{ix}$. We define the following function g_i, defined on the same domain as f_i: $g_i(a, b) = d(s_i, a) + \frac{1}{v}d(a, b) + d(b, t_i)$. This function is convex, since it is a positively weighted sum of convex functions. Now we have $f_i(a, b) = \min\{d(s_i, t_i), g_i(a, b)\}$. The value $d(s_i, t_i)$ does not depend on a and b, and can therefore be considered as a constant. So f_i is defined as the minimum between a constant and a convex function. All the levels of such a function are necessarily convex. □

LP-type problems are known to be solvable in expected time linear in the number of constraints [17], which implies that Problem 3 is solvable in $O(n)$ expected time.

Remark 3.3. The restriction on the orientation of the moving walkway is useful in making the travel time functions f_i quasiconvex. If this assumption is lifted, then it is easy to show that the levels of the functions are not even always connected. There are examples such that when interpolating between a pair of moving walkways a, b and a', b', the travel time starts to increase, then decrease afterwards because the customer uses the walkway in the other direction.

Remark 3.4. In the above proof, we implicitly use the assumption on the computation model that constant-size subproblems can be solved in constant time. This is a standard assumption, but it can make the given algorithms difficult to implement in practice.

3.2 Travel time diameter

As a next step toward an algorithm for minimum travel time diameter location, we give an $O(n \log n)$ time decision procedure for the travel time diameter given a moving walkway (not necessarily horizontal). Note that the travel time diameter for a moving walkway in the plane generalizes the travel time diameter for a moving walkway on a line, so the $O(n \log n)$ lower bound given in theorem 2.2 still holds and our algorithm is optimal.

Problem 4. Given a set P of n points, a real number $v > 1$, and a pair of points a, b, find $\max_{s,t \in P} d_{\text{time}}(s, t, a, b)$.

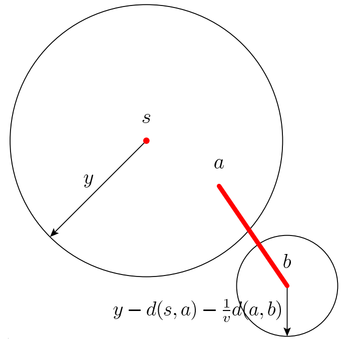

Figure 2: The travel time disk $\mathcal{B}(s, y)$.

Lemma 3.5. *The decision version of Problem 4 can be solved in optimal $O(n \log n)$ deterministic time.*

Proof. The decision problem is to check, for a given walkway a, b, whether the largest travel time $\max_{s,t \in P} d_{\text{time}}(s, t, a, b)$ between two points of P is not greater than a given value y.

We first observe that the pairs of points $s, t \in P$ such that $d_{\text{time}}(s, t, a, b) < d(s, t)$ are all separated by the bisector of the line segment $[ab]$. If both points are on the same side of the bisector, then it is always more advantageous to go directly from s to t than to use the walkway. So as a first step, we can partition the point set in two subsets $R = \{s \in P : d(s, a) \leq d(s, b)\}$ and $B = \{t \in P : d(t, a) > d(t, b)\}$, say of red and blue points, according to which side of the bisector the points lie. Then we have :

$$\max_{s,t \in P} d_{\text{time}}(s, t, a, b) = \max\{\max_{s,t \in R} d(s, t), \max_{s,t \in B} d(s, t), \max_{s \in R, t \in B} d_{\text{time}}(s, t, a, b)\}.$$

So by first computing, in time $O(n \log n)$, the Euclidean diameter of the red and blue sets, the travel time diameter problem boils down to computing the red-blue travel time diameter.

The red-blue travel time diameter is not greater than y if and only if each point $t \in B$ is contained in the intersection of the *travel time disks* of the points in R, where the travel time disk $\mathcal{B}(s, y)$ of $s \in R$ is the set of points reachable in time y from s. We observe that for a red-blue pair $s \in R, t \in B$, we have $d_{\text{time}}(s, t, a, b) = \min\{d(s, t), d(s, a) + \frac{1}{v} d(a, b) + d(b, t)\}$, which means that for going from a red to a blue point, the walkway is used from a to b. So the set $\mathcal{B}(s, y)$ for a point $s \in R$ is either (see figure 2) the disk of radius y centered at s if $y < d(s, a) + \frac{1}{v} d(a, b)$, or the union of a disk of radius y centered at s and a disk of radius $y - d(s, a) - \frac{1}{v} d(a, b)$ centered at b, otherwise.

To check the intersection condition, we proceed by first sorting the points in R in nondecreasing order of their distance to a. We denote by $s_1, s_2, \ldots, s_{|R|}$ the points in R, such that $d(s_1, a) \leq d(s_2, a) \leq \ldots \leq d(s_{|R|}, a)$.

For each point $t \in B$, we identify, in $O(\log |R|)$ time, the smallest index i such that $\forall j \geq i$: $d(s_j, a) > y - \frac{1}{v} d(a, b) - d(b, t)$. All the red points s_j for $j \geq i$ are such that we cannot go from s_j to t in time less than y by using the walkway. So for all these points, we must be able to reach t directly, which means that t must lie in the intersection of the Euclidean unit disks with centers s_j and radius y.

To check the latter condition, we can use a simple point location structure for convex circular polygons. The difficulty lies in the fact that we must keep a version of the point location structure for each intersection of the form $C_i = \bigcap_{j=i}^{|R|} \mathcal{B}(s_j, y)$ for $i = 1, 2, \ldots, |R|$. This can be achieved for free using persistent data structures techniques [11].

The point location structure that we use is made of two balanced search trees, one for the upper boundary of the intersection C_i, and another for the lower boundary. These trees allow to compute, in time logarithmic in the number of vertices of C_i, for any value of the x-coordinate, the highest (resp. lowest) point on the boundary of C_i having this x-coordinate. From this, we can check if a point belongs to C_i. We now need a technical lemma.

Lemma 3.6. *Consider the intersection I of i unit disks, and an additional, distinct, unit disk $\mathcal{B}$. The boundary of $\mathcal{B}$ intersects the boundary of I in at most two points.*

This shows that the number of vertices of C_i is $O(i)$ and that the point location data structure answers queries in time $O(\log i) = O(\log n)$. Using the partial-persistence transform [11], the additional space cost due to persistence is the total number of pointer changes on the structure over a sequence of operations. Since there are at most two new vertices on the intersection of the disks for each new value of i, the total number of changes is $O(|R|)$. So the persistent data structure uses linear space.

So for each point $t \in B$, we can check in $O(\log n)$ time whether $t \in C_i$ for the computed index i. Hence the whole algorithm takes $O(n \log n)$ deterministic time. □

3.3 Minimum travel time diameter location of horizontal moving walkways

Now given our algorithm for computing the travel time diameter and the observation that the location for n source-destination pairs is LP-type, we are able to solve the minimum diameter location problem, which is in fact an implicit quasiconvex program.

Problem 5. Given a set P of n points, and a real number $v > 1$, find a horizontal moving walkway $[a^*, b^*]$ of speed v minimizing the travel time diameter of P: $\min_{a,b:a_y=b_y} \max_{s,t \in P} d_{\text{time}}(s, t, a, b)$.

Theorem 3.7. *Problem 5 can be solved in $O(n \log n)$ randomized expected time.*

Proof. The result is by using a lemma from Chan [8] for solving some LP-type problems with a large number of constraints. This lemma states that if an LP-type optimization problem is decomposable (in some precise sense) and there exists an efficient algorithm for testing a given solution, then the whole problem can be solved within the same time bounds. This requires to be able to decompose a set of points P into r groups P_i, each of size at most $\alpha|P|$, such that the set of constraints encoded by P (the pairs $s, t \in P$) is the union of the corresponding sets for the P_i. Here, we can partition the set P in three equal-sized subsets, say Q, R and S. We form the following three groups of points : $P_1 = Q \cup R$, $P_2 = R \cup S$ and $P_3 = Q \cup S$, each group containing $2n/3$ points. We have $r = 3$ and $\alpha = 2/3$. This decomposition is satisfying, because each pair of points is included in at least one group. Another thing we need is an algorithm for the satisfaction-violation test for a group of constraints. This corresponds to a decision algorithm for the travel time diameter, which, from the previous lemma, can run in $O(n \log n)$ time. Hence from Chan's lemma, the optimization problem can be solved within the same time bounds, in the randomized expected sense. □

3.4 An approximation algorithm for moving walkway location in the plane

So far we only considered the constrained problem in which the moving walkway must be horizontal. The unconstrained problem is as follows.

Problem 6. Given a set P of n points and a real number $v > 1$, find a moving walkway $[a^*, b^*]$ of speed v minimizing the travel time diameter of P: $\min_{a,b} \max_{s,t \in P} d_{\text{time}}(s, t, a, b)$.

It is easy to verify that the constraints are not LP-type anymore, so we are not able to apply the previous randomized techniques. We can solve the problem approximately, however, using the following trick: rotate the set of points in $O(1/\varepsilon)$ directions, and for each direction solve the constrained version of the problem. The proof is omitted.

Theorem 3.8. *A $(1+\varepsilon)$-approximate solution for problem 6 can be found in $O(\frac{v}{\varepsilon} n \log n)$ randomized expected time.*

Remark 3.9. By combining the above trick with a grid technique (see for instance Har-Peled [14]), we can obtain a location algorithm whose complexity is of the form $O(n + \mathrm{p}oly(\varepsilon^{-1}, v))$. The main idea is to snap the points in P to a grid of size $\varepsilon^{-1} \times \varepsilon^{-1}$, and work on the point set formed by taking the centers of all nonempty cells. We can then run the above algorithm for $n = O(\varepsilon^{-2})$.

3.5 Escalators and Elevators

The constraint that the walkway must be horizontal is one way among others to make the problem more tractable by reducing it to an implicit quasiconvex program. We can also simply assume that the walkway is unidirectional, i.e. we can only travel from a to b, and not from b to a.

If we adopt this convention, however, the travel time diameter is always equal to the Euclidean diameter, since for a given source-destination pair s, t, the moving walkway cannot be useful in both directions. So we must further assume that for a given pair of points, the travel direction is determined. We can suppose, for instance, that we always travel from left to right. Note that this does not mean that we always use the walkway for going from left to right, so the problem is not equivalent to Problem 6. We can also assume that the point set is partitioned in red and blue subsets, and that we always travel from a red point to a blue point.

For these two restricted problems with unidirectional walkways (*escalators*), the quasiconvexity of the constraints is guaranteed, and the diameter algorithm can be adapted. So we can solve them in $O(n \log n)$ randomized expected time.

If we restrict the red-blue version further so that the escalator is a single point and all customers have to use it, then we have a suitable model for an *elevator* location problem, where the red and blue sets represent two different floors of a building. In this case, the problem is similar to a point location problem mentioned by Eppstein ([12], section 3.2), and solvable in linear expected time using Chan's trick again.

References

[1] M. Abellanas, F. Hurtado, C. Icking, R. Klein, E. Langetepe, L. Ma, B. Palop, and V. Sacristán. Voronoi diagram for services neighboring a highway. *Information Processing Letters*, 86:283–288, 2003.

[2] M. Abellanas, F. Hurtado, and B. Palop. Transportation networks and Voronoi diagrams. In *Proc. International Symposium on Voronoi Diagrams in Science and Engineering (VD)*, 2004.

[3] M. Abellanas, F. Hurtado, and B. Palop. The heavy-luggage metric. *International Journal of Computational Geometry and Applications*, to appear.

[4] H.-K. Ahn, H. Alt, T. Asano, S. W. Bae, P. Brass, O. Cheong, C. Knauer, H.-S. Na, , C.-S. Shin, and A. Wolff. Constructing optimal highways. In *Proc. 13th Conf. Computing: The Australasian Theory Symposium (CATS)*, volume 65. CRPIT, 2007. To appear.

[5] O. Aichholzer, F. Aurenhammer, and B. Palop. Quickest paths, straight skeletons, and the city Voronoi diagram. *Discrete & Computational Geometry*, 31(1):17–35, 2004.

[6] N. Amenta, M. Bern, and D. Eppstein. Optimal point placement for mesh smoothing. *Journal of Algorithms*, 30:302–322, 1999.

[7] J. Cardinal and S. Langerman. Min-max-min geometric facility location problems. In *Proc. European Workshop on Computational Geometry (EWCG)*, pages 149–152, 2006.

[8] T. M. Chan. An optimal randomized algorithm for maximum Tukey depth. In *Proc. ACM-SIAM Symposium on Discrete Algorithms (SODA)*, pages 430–436, 2004.

[9] Z. Drezner, editor. *Facility Location: A Survey of Applications and Methods.* Springer Series in Operations Research. Springer-Verlag, Jan. 1996.

[10] Z. Drezner and H. W. Hamacher, editors. *Facility Location: Applications and Theory.* Springer-Verlag, 2001.

[11] J. Driscoll, N. Sarnak, D. D. Sleator, and R. E. Tarjan. Making data structures persistent. *Journal of Computer and System Sciences*, 38(1):86–124, 1989.

[12] D. Eppstein. Quasiconvex programming. In *Combinatorial and Computational Geometry*, pages 287–331. MSRI Publications, 2005.

[13] R. Görke, C.-S. Shin, and A. Wolff. Constructing the city Voronoi diagram faster. *International Journal of Computational Geometry and Applications*, to appear.

[14] S. Har-Peled. A practical approach for computing the diameter of a point set. In *Proc. seventeenth annual symposium on Computational geometry (SoCG)*, pages 177–186, 2001.

[15] J. M. J.M. Díaz-Báñez and A. Schöbel. Continuous location of dimensional structures. *European Journal of Operational Research*, 152:22–44, 2004.

[16] J.-M. Robert and G. T. Toussaint. Computational geometry and facility location. In *Proc. International Conf. on Operations Research and Management Science*, volume B, pages 1–19, Dec. 1990.

[17] M. Sharir and E. Welzl. A combinatorial bound for linear programming and related problems. In *Proc. Annual Symposium on Theoretical Aspects of Computer Science (STACS)*, pages 569–579, 1992.