Shortest Tour of a Sequence of Disjoint Segments in L_1

Esther Arkin * Alon Efrat† Cesim Erten‡ Ferran Hurtado§ Joseph Mitchell*

Valentin Polishchuk* Carola Wenk¶

September 22, 2006

Abstract Given a sequence $s_1, \dots, s_K$ of K disjoint segments in the plane, a start point s and a target point t, we seek a path, that starts at s, visits in order each of the segments, and ends at t, such that the L_1 length of the path is minimized. We give an $O(K^2)$ algorithm that builds a data structure of size $O(K)$ such that the shortest path, visiting k first segments in the sequence, to any point in the plane can be output in $O(k)$ time.

Related work In [2] Dror et al. solved the problem in the Euclidean metric. The touring problem can be formulated as a convex optimization program — [3] uses conic programming to find optimal tours in $\mathbb{R}^d$.

Definitions and notation We say that a path π *visits* the sequence $s_1, \dots, s_K$ if it starts at s and there exist points $p_1 \in s_1, \dots, p_K \in s_K$ such that $p_1, \dots, p_K$ appear in order along π. Let p_k denote the first point of π (i.e., the point closest to s along the path) that lies in s_k and comes *after* p_{k-1} along π; we call the points $p_1, \dots, p_K$ the *first contact points*. For a point $z \in \mathbb{R}^2$ and $k \in \{1 \dots K\}$, a *k-path* to z is a path that visits $s_1, \dots, s_k$; let $\pi_k(z)$ be a shortest k-path to z; our problem is then to find $\pi_K(t)$. We let $p_k(z)$ denote the set of the possible first contact points of $\pi_k(z)$ with s_k, i.e., the points p in s_k such that $|zp| + |\pi_{k-1}(p)|$ is minimum over all $p \in s_k$. Let R_k be the smallest axis-aligned rectangle enclosing s_k. For a point $z \in R_k$ let $h_k(z)$, $v_k(z)$ denote the "horizontal" and "vertical" projection of z on s_k. We denote the endpoints of s_k by a_k, b_k.

As with the shortest Euclidean tours, we may consider, WLOG, only the tours that are polygonal chains that bend only at the segments. The first points of contact of the tour with the segments, $p_1, \dots, p_K$, may be classified into three types: a *bend* at a vertex, a *reflection* off a point interior to the segment, and a *pass-through*.

The algorithm of [2] builds a set of subdivisions $\mathcal{S}_k$, $k \in 1 \dots K$, such that for any point z in a cell of kth subdivision, $\pi_k(z)$ is in the "same-type" contact with s_k (see below for description of the contact types). The correctness of the algorithm is due to a uniqueness lemma: in L_2 metric, for any $x \in \mathbb{R}^2$, and any $k = 1 \dots K$, the path $\pi_k(x)$ is unique. Since in L_1 metric the shortest paths are, in general, not unique, the algorithm of [2] does not straightforwardly extend to L_1. Nevertheless, using the ideas from [2], we build similar subdivisions for the shortest L_1 tours. One important distinction is that the vertices of our subdivisions do not necessarily come from the endpoints of the segments, as is the case for the shortest Euclidean tours [2]. In particular, $\mathcal{S}_k$ may consist of just one vertex cell, say, $S(a_k) = \mathbb{R}^2$, in which case $\forall z \in \mathbb{R}^2$, $\pi_k(z)$ may WLOG go through a_k; we say then that $\mathbb{R}^2$ is a_k*-dominated*.

The structure of $\mathcal{S}_k$ We show that $\mathcal{S}_k$ has constant complexity and that for a point p in s_k the length $l_k(p) \equiv |\pi_k(p)| \equiv |\pi_{k-1}(p)|$ of the shortest k-path to p is a convex piecewise-linear function of p having at most $3k$ breakpoints. For each k we compute the subdivision $\mathcal{S}_k$ and the function $l_k(p)$.

Lemma 1. *The problem of finding the shortest k-path $\pi_k(z)$ to a point $z \in \mathbb{R}^2$ may be formulated as a linear program.*

Proof. Consider the following program for finding $\pi_k(z)$:

$$\begin{aligned} &\text{minimize } t_1 + \dots + t_{k+1} \\ &\text{subject to: } t_i = \|p_i - p_{i-1}\|_1 \qquad i = 1 \dots k+1 \qquad (1) \\ &p_i \in s_i \qquad i = 1 \dots k \qquad (2) \\ &p_0 = s, \quad p_{k+1} = z \end{aligned}$$

The decision variables of the program are the coordinates of the first contact points $p_1, \dots, p_k$. The constraints (2) are linear. Each of the constraints (1) may be written as a set of five linear constraints:

$$t_i = \|p_i - p_{i-1}\|_1 \quad \Leftrightarrow \quad \begin{aligned} t_i^x &\geq p_i^x - p_{i-1}^x \\ t_i^x &\geq p_{i-1}^x - p_i^x \\ t_i &= t_i^x + t_i^y \\ t_i^y &\geq p_i^y - p_{i-1}^y \\ t_i^y &\geq p_{i-1}^y - p_i^y \end{aligned}$$

where p_i^x, p_i^y are the coordinates of p_i. Thus, the program is an LP. □

Next we show that $l_k(p)$, $l_k : s_k \mapsto \mathbb{R}^+$ is a convex function.

Lemma 2. *For a point $p \in s_k$, $l_k(p)$ is a convex function of p.*

Proof. It is a well-known fact [1, Problem 6.70] that as the right-hand-side of a minimization LP changes linearly with rate λ, the objective function is a convex function of λ. □

Let z be a point in the plane.

Lemma 3. *The distance $|zp|$ from z to a point $p \in s_k$ is a convex function of p.*

Lemma 4. *$p_k(z)$ is a contiguous subset of s_k.*

Proof. Consider $|\pi_k(z)| = |zp| + |\pi_{k-1}(p)| = |zp| + l_k(p)$ as a function of $p \in s_k$. From Lemmas 2, 3 it is a convex function and thus, its minimizers form a contiguous subset of its domain. □

Remark. We will assume sometimes that for some $z \in \mathbb{R}^2$, $p_k(z)$ is a single point. This assumption can be made WLOG since we can always perturb the slope of s_k.

We now describe the structure of $\mathcal{S}_k$. Some points in the plane may be easily assigned to the vertex cells. Indeed, let $S(a_k)$ (resp., $S(b_k)$) be the points in the quadrant of the coordinate system, with the origin at a_k (resp., b_k), that is "opposite" s_k (Fig. 1, left). The tours to the points in the cell may WLOG go through the corresponding vertex:

Lemma 5. *For $z \in S(a_k)$, $a_k \in p_k(z)$; for $z \in S(b_k)$, $b_k \in p_k(z)$.*

*Applied Math and Statistics, Stony Brook University.
†Computer Science, the University of Arizona.
‡Computer Science and Engineering, Isik University.
§Matemática Aplicada II, Universitat Politécnica de Catalunya.
¶Computer Science, the University of Texas at San Antonio.

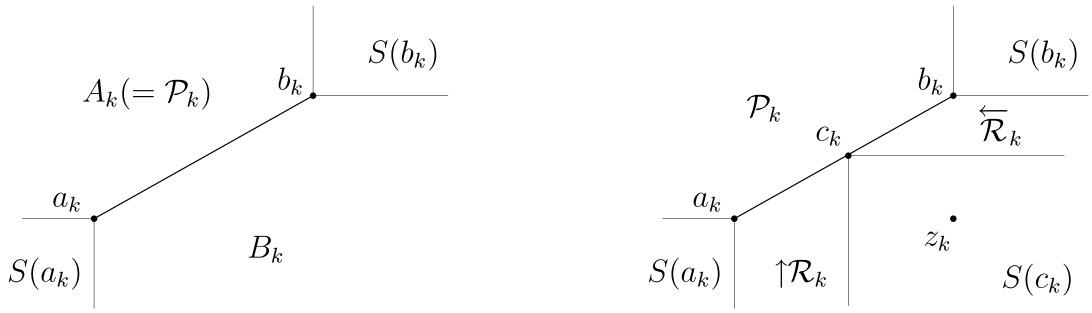

Figure 1: Left: Vertex cells $S(a_k)$ and $S(b_k)$ and the pass-through cell $\mathcal{P}_k$ are easily constructed. Right: $\mathcal{S}_k$.

The points in the plane that are not assigned yet to the vertex cells form an "hourglass" with s_k as the "neck"; call the two "bulbs" of the hourglass, A_k and B_k, the *sides* of s_k (so that $\forall x, y \in \mathbb{R}^2 \setminus (s_k \cup S(a_k) \cup S(b_k))$, $xy \cap s_k = \emptyset$ iff x and y are on the same side of s_k).

Lemma 6. *Either shortest $(k-1)$-paths arrive from the same side of s_k for any point in s_k, or $\mathbb{R}^2$ is a_k-dominated, or $\mathbb{R}^2$ is b_k-dominated: $\forall p \in s_k$ either $pp_{k-1}(p) \cap A_k = \emptyset$, or $pp_{k-1}(p) \cap B_k = \emptyset$, or $a_k \in p_{k-1}(p)$, or $b \in p_{k-1}(p)$.*

Proof. (Sketch.) By Lemma 4 and continuity. □

By Lemma 6 it is enough to check one point in each of A_k, B_k to understand which of them (if any) is the pass-through cell $\mathcal{P}_k$ of $\mathcal{S}_k$; say, $\mathcal{P}_k = A_k$ (Fig. 1, left). Now the only missing part in $\mathcal{S}_k$ is the assignment of the points in B_k.

Lemma 7. *$\mathcal{S}_k$ has at most three cells in B_k.*

Proof. Let $\mathcal{T}_k = R_k \cap B_k$ be the right triangle $z_k a_k b_k$ so that $a_k = v_k(z_k)$, $b_k = h(z_k)$, $z_k \in B_k$. Since the k-paths to any point in $B_k \setminus \mathcal{T}_k$ may WLOG go through z_k, it is enough to consider only the restriction of $\mathcal{S}_k$ to $\mathcal{T}_k$. Assume (WLOG) that $c_k = p_k(z_k)$ is a single point. Let R'_k be the axis-aligned rectangle having z_k and c_k as opposite corners. Let z be a point in R'_k. Since the path $z_k p_k(z_k) = z_k c_k$ may WLOG go through z, $p_k(z) = c_k \forall z \in R'_k$. It is easy to see that for a point $z \in \mathcal{T}_k$, $p_k(z) \subset v_k(z)h_k(z)$. By continuity, for $z \in \uparrow\mathcal{R}_k$, $zp_k(z)$ is vertical, $p_k(z) = v_k(z)$; for $z \in \overleftarrow{\mathcal{R}}_k$, $zp_k(z)$ is horizontal, $p_k(z) = h_k(z)$, refer to Fig. 1, right. □

Remark. It may happen that one (or both) of the cells $\uparrow\mathcal{R}_k$, $\overleftarrow{\mathcal{R}}_k$ is empty, if, say, $c_k = a_k$; in this case the vertex cell $S(a_k)$ is a halfplane through a_k.

Building $\mathcal{S}_k$ We build the subdivisions recursively. First we assign $S(a_k)$ and $S(b_k)$. Then we determine which cells of $\mathcal{S}_{k-1}$ are intersected by s_k. For p in the part of s_k, intersecting a vertex cell $S_{k-1}(v_{k-1})$ of $\mathcal{S}_{k-1}$, $l_k(p) = |pv_{k-1}| + l_{k-1}(v_{k-1})$. For p in a reflect cell, say, $\overleftarrow{\mathcal{R}}_{k-1}$, $l_k(p) = |ph_{k-1}(p)| + l_{k-1}(h_{k-1}(p))$. For $p \in s_k \cap \mathcal{P}_{k-1}$, we look at the subdivision $\mathcal{S}_{k-2}$ to deduce $l_k(p)$. We then find k-paths to the vertices of R_k to deduce $\mathcal{P}_k$. Finally, we find $p_k(z_k) = \arg\min_{p \in s_k \cap (\mathbb{R}^2 \setminus (S(a_k) \cup S(b_k) \cup \mathcal{P}_k))} [|z_k p| + l_{k-1}(p)]$ to complete the subdivision.

The complexity The time-dominating step of the above procedure is computing $l_k(p)$. Since, in principle, part of s_k may fall into $\mathcal{P}_{k-1}$, the complexity C_k of $l_k(p)$ may be at least $C_{k-1} + C_{k-2}$, which is at least exponential in k. We show (Lemma 9) that in fact, C_k is linear and thus, $\mathcal{S}_k$ and $l_k(p)$ can be built in $O(k)$ time.

Lemma 8. *Let p move uniformly from a_k to b_k. Then $p_{k-1}(p)$ moves (weakly) monotonically along a segment on s_{k-1}.*

Proof. Since the cells of $\mathcal{S}_{k-1}, \ldots, \mathcal{S}_1$ are convex, once p leaves a cell, it never enters it again. While p is within a (reflect or vertex) cell σ, $p_{k-1}(p)$ either moves together with p (horizontally or vertically, if σ is a reflect cell) or does not move at all (if σ is a vertex cell). □

Similarly, all first points of contact, $p_{k-2} = p_{k-2}(p_{k-1}(p))$, $p_{k-3} = p_{k-3}(p_{k-2}(p_{k-1}(p)))$,$\ldots$,$p_1$, move (weakly) monotonically along their segments. The breakpoints of $l_k(p)$ correspond to events when one of $p_1, \ldots, p_{k-1}$ hits a vertex of the corresponding subdivision. Thus,

Lemma 9. $C_k \leq 3(k-1)$.

Finding the optimal tour Given the subdivisions $\mathcal{S}_1, \ldots, \mathcal{S}_K$ we can readily compute, for any $k = 1 \ldots K$, the optimal tour $\pi_k(q)$ to any point $q \in \mathbb{R}^2$. We start by locating q in $\mathcal{S}_k$. If q is in the pass-through cell of $\mathcal{S}_k$, then $\pi_k(q) = \pi_{k-1}(q)$ and we continue by locating q in $\mathcal{S}_{k-1}$. If q is in a vertex cell $S_k(v)$ of $\mathcal{S}_k$, then $p_k = v$. If q is in a reflect cell, then we follow the "arrow" of the cell until hitting the boundary of P_k at p_k. We then recursively compute $\pi_{k-1}(p_k)$.

Theorem 10. *The subdivisions $\mathcal{S}_1, \ldots, \mathcal{S}_K$ of the plane, of total size $O(K)$, can be built time $O(K^2)$ that enable computing shortest tours $\pi_k(q)$ to a query point q in $O(k)$ time.*

Future work We believe that the approach of this paper can be generalized to find shortest tours through a sequence of arbitrary convex bodies (disjoint or not) and to higher dimensions.

References

[1] M. S. Bazaraa, J. J. Jarvis, and H. F. Sherali. *Linear Programming and Network Flows*. John Wiley & Sons, New York, NY, USA, second edition, 1990.

[2] M. Dror, A. Efrat, A. Lubiw, and J. S. B. Mitchell. Touring a sequence of polygons. In *STOC '03: Proceedings of the thirty-fifth annual ACM symposium on Theory of computing*, pages 473–482, New York, NY, USA, 2003. ACM Press.

[3] V. Polishchuk and J. S. B. Mitchell. Touring convex bodies – a conic programming solution. In *Proceedings of the 17th Canadian Conference on Computational Geometry (CCCG'05)*, pages 290–293, 2005.