Computing an area-optimal convex polygonal stabber of a set of parallel line segments

Asish Mukhopadhyay
Chanchal Kumar
Department of Computer Science
Indian Institute of Technology, Kanpur

Binay Bhattacharya
School of Computing Science
Simon Fraser University, Bunaby

Abstract

We describe an $O(n \log n)$ algorithm for computing an area-optimal convex polygon that intersects a set of parallel line segments. We also consider the problem of maintaining such a minimum-area polygon when both insertion and deletion of line segments are allowed.

1 Introduction

The problem of finding a set of necessary and sufficient "local conditions" for the existence of a transversal of a set of ovals (a closed, bounded and convex set) is a well-studied one in Combinatorial Geometry. [HDK64], [GPW91].

Interest in the algorithmic aspects of this problem is however quite recent. Depending on the type of the transversal and the set of ovals we get different variants of the basic problem.

O'Rourke gave an on-line algorithm for intersecting a set of parallel line segments by straight lines [O'R81]. Computing transversals for a set of arbitrarily oriented line segments was discussed by Edelsbrunner et al [EMP+82].

Determining line transversals of more complex objects than line segments was studied by Edelsbrunner [Ede85]. Atallah and Bajaj gave a general technique for computing line transversals of a set of objects requiring constant size storage description by tying up the problem with that of computing Davenport-Schnizel sequences [AB87]. This result was further generalised by Srinivasaraghavan et al [SM92]. Avis and Doskas [AD87], Avis and Wenger [AW89] studied the problems of computing line transversals for lines, line segments and polyhedra in three and higher dimensional spaces.

Goodrich and Snoeyink were (probably) the first to consider the a different kind of transversal than straight lines. They studied the problem of computing a convex polygon that stabs a set of parallel line segments [GS90].

By defining a measure m on the class of transversal in question we get an optimisation problem. An example is that of computing the shortest line segment that intersects a set of discs. Currently, there are very few optimisation results. Lyons et al studied the problem of computing a minimum perimeter convex polygon that intersects a set of isothetic line segments [LMR]. $O(n \log n)$ time algorithms for computing a shortest line segment that intersects a family of lines, a family of line segments and finally a family of convex polygons were given by Bhattacharya et al [BCE+91]. Jadhav et al gave a linear time algorithm for computing a smallest radius circle that intersects a set of convex polygons [JMB92].

In this paper we enlarge this set of results by solving the problem of computing a minimum area convex polygon that intersects a set of parallel line segments. A line segment intersects a convex polygon if it intersects the interior or the boundary of the polygon.

We also consider the dynamic version of this problem : maintain a minimum-area convex polygon that intersects a dynamically changing set of line segments

The paper is organised as follows. In the next section we introduce some notations and definitions. The static version of the algorithm is described in the third section. The following section

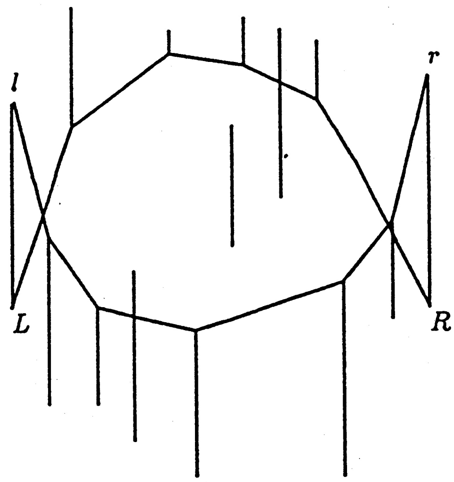

Figure 1: Convex chains $luc(S)$ and $hdc(S)$ for a set of line segments.

contains an analysis of this algorithm. In the next section we take up the dynamic version of this algorithm and analyse its complexity. In the fifth and last section we conclude the paper and also indicate directions for further research.

2 Definitions and Notations

We denote a line segment with end-points p and q by $\overline{pq}$. Let S denote a set of n parallel line segments. The functions top(.) and bot(.) return the upper and lower end-points of a line segment.

The upper chain of the convex hull of the lower end-points of the line-segments in S has the property that bot(s) of each line-segment s lies on or below it. If we partially order convex chains over a given range of x-values by defining a chain to be "less than or equal" to another if at every point of the range the corresponding y-value of the former is less than or equal to the corresponding y-value of the latter, then the upper hull of the lower end-points is the "smallest" one in the given partial order to have the above property . To reflect this we denote this *l*owest *u*pward-*c*onvex chain by $luc(S)$.

Similarly, the lower chain of the convex hull of the upper end-points is the "largest" among all convex chains which have top(s) for each line segment s lying on or above it. We denote this *h*ighest *d*ownward-*c*onvex chain by $hdc(S)$.

We assume, without loss of generality, that there is a unique leftmost line-segment $\overline{lL}$ and a unique rightmost line-segment $\overline{rR}$. Clearly, the end-points of $luc(S)$ are L and R and those of $hdc(S)$, l and r. Let $< u_1, u_2, ..., u_p >$ be the ordered set of vertices on $luc(S)$ from L to R and $< v_1, v_2, ..., v_q >$ those on $hdc(S)$ from l to r. Let P' be the convex region that consists of the points lying on or below $luc(S)$ and on or above $hdc(S)$ (Fig. 1).

It is easy to see that if P is a convex polygon that intersects all the line-segments in S then at every value of x between $\overline{lL}$ and $\overline{rR}$ the upper hull of P lies "on or above" $luc(S)$ and its lower hull lies "on or below" $hdc(S)$. It follows that P' is contained in P. Further we can assume that no part of P lies to the left of $\overline{lL}$ or to the right of $\overline{rR}$. In fact we can restrict our search to the class of polygons that has exactly one point in common with each of the extreme line segments.

Let u and v be the vertices of P that lie on the leftmost and rightmost line-segments respectively; $UC(P)$ and $LC(P)$ are its upper and lower chain respectively.

3 The Algorithm

We first need to characterise the area-optimal convex polygon P.

To eliminate a trivial case, we assume that the extreme line segments are of non-zero length.

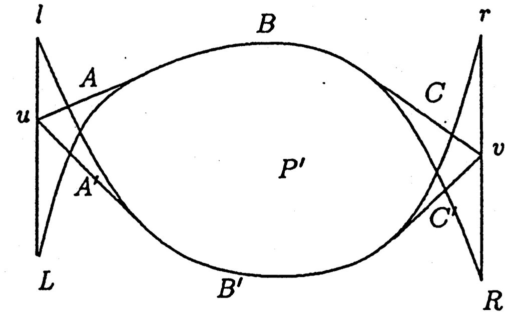

Figure 2: Parts $A(A')$, $B(B')$ and $C(C')$ of the upper(lower) chain of a convex polygon, enclosing P'.

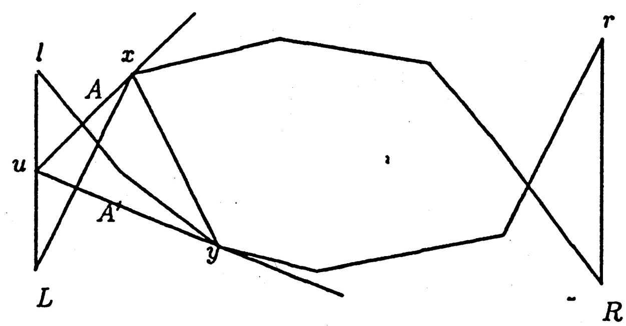

Figure 3: Edges A and A', touching P' at x and y

It is intuitively clear that $UC(P)$ ($LC(P)$) must hug $luc(S)$ ($hdc(S)$) as closely as possible. The problem is to express this in a quantitative way. We argue for $UC(P)$; the argument is identical for $LC(P)$.

The chain $UC(P)$ has three distinct parts - a left part A between u and the first point of contact of $UC(P)$ with $luc(S)$, a middle part B that is common with $luc(S)$ and a right part C between the last point of contact of $UC(P)$ with $luc(S)$ and v (Fig. 2).

It is obvious that $A(B)$ cannot have any vertex of $UC(P)$ lying strictly between $u(v)$ and the first(last) point of contact between $UC(P)$ and $luc(S)$. Otherwise P would not be of minimum area. Therefore the parts A and B are straight-line segments. If we extend these, each is tangent to $luc(S)$ along an entire edge or at a vertex only.

Let A', B' and C' be the corresponding parts of LC(P). The following lemmas provide a necessary characterisation of P.

Lemma 1 *If P is a convex polygon of minimum area that intersects all the line segments in S, then either A (respectively B), when extended, touches $luc(S)$ along an edge or A'(respectively B'), when extended, touches $hdc(S)$ along an edge.*

If one of A or A', when extended, touches along an edge and the other at a vertex, the edge and the vertex are related as in the lemma below.

Lemma 2 *If A when extended touches the edge $\overline{xy}$(vertex z) of $luc(S)$ and A' when extended touches the vertex z (edge $\overline{xy}$) of $hdc(S)$ then the vertical line through z intersects $\overline{xy}$.*

Lemma 3 *If A when extended touches the edge $\overline{uv}$ of $luc(S)$ and A' when extended touches the edge $\overline{u'v'}$ of $hdc(S)$ then either the vertical line through u intersects $\overline{u'v'}$ or the vertical line through u' intersects $\overline{uv}$.*

Lemmas 2 and 3 above hold when we replace A and A' by B and B' respectively.

The above characterisation is also sufficient because if P' is any other polygon which satisfies the conditions of the above lemmas, then any small perturbation of P' only increases its area.

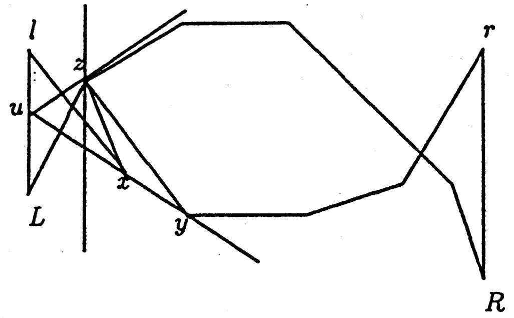

Figure 4: Edge of contact $\overline{xy}$, lying to the right of contact point z.

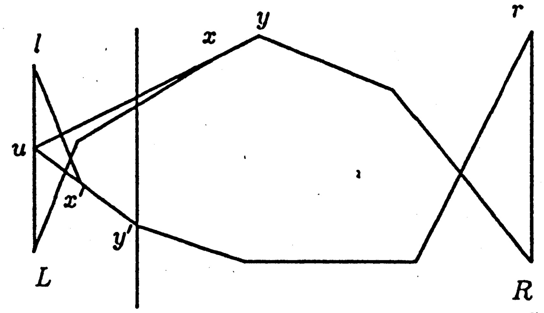

Figure 5: Edge of contact $\overline{xy}$, lying to the right of edge of contact $\overline{x'y'}$.

To compute A and A' we consider an edge of $luc(S)$ or $hdc(S)$ and extend it. If it intersects the leftmost segment, we draw a tangent from the point of intersection to the other chain, and check if any of Lemmas 2 or 3 applies. If not, we repeat until such an edge is found. Parts B and B' are computed similarly.

We can now formally state the algorithm to compute the parts A and A'.

```
Algorithm MinPolyStabber(S, A, A')
Step 1. Compute the upper hull luc(S) of the points bot(s)
        and the lower hull hdc(S) of the points top(s).
Step 2. If all edges on luc(S) are marked then
           go to Step 3
        else
           choose an unmarked edge e on luc(S);
        if the supporting line of e does not intersect lL then
           mark e and go to Step 2
        else
           draw a tangent to hdc(S) from the point of intersection;
        if Lemma 2 or Lemma 3 applies to the configuration
        of tangents so obtained then
           return A, A' and quit
        else
           mark e and go to Step 2.

Step 3. Repeat Step 2 with hdc(S) replaced by luc(S) and
        vice versa, returning B, B' instead of A, A'.
```

4 Analysis of the Algorithm

The complexity of Step 1 is bounded above by $O(n \log n)$. Since it requires time logarithmic in the size of a chain to compute a tangent to it from a given point, the complexity of Step 2 is bounded above by $O(n \log n)$ also. Same goes for Step 3. Therefore the time complexity of the algorithm to compute parts A and A' is $O(n \log n)$. Since B and B' can be computed in the same time complexity, the time complexity of the algorithm is $O(n \log n)$.

5 Dynamic algorithm

In this section we consider a dynamic version of the above algorithm in which both insertion and deletion of line segments are allowed. Since insertion and deletion of line segments require us to update the chains *luc* and *hdc*, we can use the dyanmic convex hull maintenance algorithm of Overmars and van Leeuwen [OvL81] for this purpose. It is well-known that dynamic convex-hull maintenance requires $O(log^2 n)$ worst-case time for each insertion/deletion operation. Thus we can maintain a minimum area convex polygon which intersects a dynamic set of line segments provided we can find a more efficient way of determining the points on the leftmost and rightmost line segments from which to draw tangents to the chains *luc* and *hdc* than given in the algorithm of the last section.

It is enough to examine this problem for the rightmost segment. From r we draw a tangent to the chain *luc*; let u be the leftmost point of contact between this tangent and *luc*. We will do a binary search of the part of *luc* from u to R. The part of *hdc* over which we do binary search will be automatically defined as follows. We extend an edge in the middle of the sub-chain we have found to intersect $\overline{rR}$ and then draw a tangent to *hdc* from this intersection point. The leftmost point of tangency and r define the binary search range for *hdc*.

How is the search carried out ? Well, if the traingle defined by the tangents lies to the left of the vertical through its apex, which, say, belongs to *hdc*, then we move to the middle of the remaining range of *luc* else to the middle of the remaining range of *hdc*. The search terminates when the vertical through the apex of the triangle intersects its base.

Since we can draw a tangent to a convex chain from a point in time logarithmic in the size of the chain, and binary search is logarithmic in the size of the range we can do all of the above in $O(log^2 n)$ time.

Hence dynamic maintenance of a minimum area convex polygon that intersects a set of vertical line segments can be done in worst-case time of $O(log^2 n)$.

6 Conclusions

In this paper we have described an $O(n \log n)$ algorithm for computing a minimum area polygonal disk that stabs a set of parallel line segments. The natural problem to tackle next is to compute a minimum area convex polygon that stabs a set of isothetic line segments, with the ultimate goal of solving this problem for an arbitrary set of line segments. Another interesting problem to look at is that of characterising a set of parallel line segments that can be stabbed by the boundary of a convex polygon.

Acknowledgements

The authors thank Mr. Shreesh 'Genius' Jadhav for writing all the Latex routines for drawing the figures and for a careful reading of the manuscript.

References

[AB87] M. Atallah and C. Bajaj. Efficient algorithms for common transversals. *Information Processing Letters*, 25:87–91, 1987.

[AD87] D. Avis and M. Doskas. Algorithms for high dimensional stabbing problems. *Discrete Applied mathemtics*, 27:39–48, 1987.

[AW89] D. Avis and R. Wenger. Polyhedral line traversals in space. *Discrete and Computational Geometry*, 4, 1989.

[BCE+91] B. K. Bhattacharya, J. Czyzowicz, P. Egyed, G. Toussaint, I. Stojmenovic, and J. Urrutia. Computing shortest transversals of set. In *Proc. of the Seventh Annual ACM Symp. on Computational Geometry*, pages 71–80, 1991.

[Ede85] H. Edelsbrunner. Finding transversals for sets of simple geometric figures. *Theoretical Computer Science*, 35:31–45, 1985.

[EMP+82] H. Edelsbrunner, H. A. Maurer, F. P. Preparata, A. L. Rosenberg, E. Welzl, and D. Wood. Stabbing line segments. *BIT*, 22:274–281, 1982.

[GPW91] Jacob E. Goodman, Richard Pollack, and Rephael Wenger. Geometric transversal theory. Technical Report 91-76, DIMACS, 1991.

[GS90] M. Goodrich and J. Snoeyink. Stabbing parallel segments with a convex polygon. *Computer vision, Graphics and Image Processing*, 49:152–170, 1990.

[HDK64] H. Hadwiger, H. Debrunner, and V. Klee. *Combinatorial Geometry in the plane.* Holt, Rinehart and Winston, 1964.

[JMB92] S. Jadhav, A. Mukhopadhyay, and B. Bhattacharya. An optimal algorithm for the intersection radius of a set of convex polygons. In *Proc. of the Twelvth Annual FSTTCS Conference*, 1992.

[LMR] Kelly A. Lyons, Henk Meijer, and David Rappaport. Minimum polygon stabbers of isothetic line segments. Department of Computing and Information Science, Queen's University, Ontario, Canada.

[O'R81] J. O'Rourke. An on-line algorithm for fitting straight lines between data ranges. *CACM*, 24:574–578, 1981.

[OvL81] Mark Overmars and J. van Leeuwen. Maintenance of configurations in the plane. *Journal of Computer and System Sciences*, 23:166–204, 1981.

[SM92] G. Srinivasaraghavan and Asish Mukhopadhyay. Stabbing simple planar sets. Technical Report TRCS-92-152, IIT Kanpur, Computer Science and Engg. Dept., 1992.