Information Processing Letters 38 (1991) 15–21
North-Holland

12 April 1991

Bisections and ham-sandwich cuts of convex polygons and polyhedra

Ivan Stojmenović *

Computer Science Department, University of Ottawa, Ottawa, Ont., Canada K1N 9B4

Communicated by M.J. Atallah
Received 22 July 1990
Revised 31 August 1990 and 5 December 1990

Keywords: Computational geometry, design of algorithms, bisection of polytopes

Introduction

We present a linear time sequential algorithm for finding a straight line that *bisects* two given disjoint convex polygons (i.e. cuts both of them into parts of equal area). The solution can be generalized to other measures (for instance, perimeter) and other proportions of cutting. Also, the problem is studied in three-dimensional space: find a plane that cuts each of three given disjoint convex polyhedra into two parts of equal volume. If no straight line exists that intersects all polyhedra, then the problem can be solved in

$$O(n \log n + nK^3)$$

time, where n is the total number of vertices of the polytopes, and K is the number of output bits of accuracy in the solution.

To the best of our knowledge, only discrete versions of a related problem, cutting two given regions by a straight line into two subregions containing the same number of points, have been studied in [4,5] (called also the ham-sandwich problem).

Bisections and ham-sandwich cuts of polygons

In this section we will study the two-dimensional ham-sandwich cut problem:

Problem 1. Given two disjoint convex polygons in the plane, find a straight line that bisects both polygons.

Problem 1 has at most one solution, as an immediate consequence of the following property which can be easily shown:

Property 1. *The intersection of any two straight lines that bisect a convex polygon P lies inside P.*

The straight line bisection of P passing through a vertex V of P is called the *V-vertex bisector of P.*

---

* This research is supported by the National Science and Engineering Research Council of Canada.

In order to solve Problem 1 we first consider the following one:

Problem 2. Given a vertex V of a convex polygon P, construct the V-vertex bisector of P.

Let $P_1, P_2, \dots, P_n$ be vertices of P listed in counterclockwise order, and let $m(P)$ be the area of P. In linear time one can determine the areas of all the triangles, $P_1P_{i-1}P_i$, i.e.

$$m(P_1P_{i-1}P_i) \quad (3 \leqslant i \leqslant n).$$

Applying a prefix sum technique, one can find

$$m(P_1P_2 \dots P_{i-1}P_i) \quad (3 \leqslant i \leqslant n),$$

and detect a vertex P_i of P, with the smallest possible index i, for which

$$m(P_1P_2 \dots P_{i-1}P_i) \geqslant m(P)/2,$$

again in linear time. We refer to edge $P_{i-1}P_i$ and vertices P_{i-1} and P_i as the *opposite* edge and vertices for vertex P_1. Then the P_1-vertex bisection is found by "interpolation" within the opposite edge $P_{i-1}P_i$ of P_1 (the interpolation, clearly, takes constant time). It is easy to show that the above scan procedure takes linear time.

Problem 3. Construct the V-vertex bisector for each vertex V of a given convex polygon P.

If the orientations of all vertex bisectors are chosen from vertices toward the interior of P, then, on the basis of Property 1, one can show the following:

Property 2. *The angles of vertex bisectors of* P *(with respect to a fixed direction) are ordered by the same circular order as the vertices of* P.

Using Property 2 one can solve Problem 3 by a counterclockwise scan around the vertices of P, starting by vertex P_1 (i.e. starting by solving Problem 2). Suppose the last vertex bisector passed through vertex P_l and intersected the opposite edge $P_{i-1}P_i$. Find

$$m(P_{l+1}P_{l+2} \dots P_{i-1}P_i)$$

from

$$m(P_lP_{l+1} \dots P_{i-1}P_i)$$

and $m(P_lP_{l+1}P_i)$. Starting from P_i, vertices of P are checked (in counterclockwise order) one by one until one is found which adds to more than half of the area. Then the P_{l+1}-vertex bisector is found by interpolation (in constant time) within the corresponding opposite edge. Thus Problem 3 can be solved in linear time, which is optimal. We now turn our attention to Problem 1.

Suppose we are given two disjoint convex polygons P and Q, with n' and n'' vertices, respectively. Without loss of generality, we assume that P and Q are separable by a vertical line v. We divide all vertex bisectors of P and Q into two groups: inner and outer, according to whether or

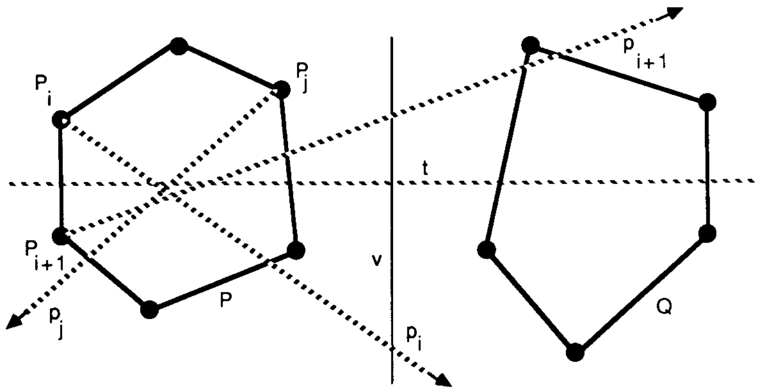

Fig. 1.

not (respectively) the corresponding oriented half-lines intersect v. Let p_i denote the P_i-vertex bisector of P ($1 \leqslant i \leqslant n'$). Then, for instance, in Fig. 1, p_i and p_{i+1} are the inner, while p_j is the outer bisector of P. Given two convex polygons P and Q and a bisecting line e of P, let $F(e, Q)$ be the area of the fraction of Q that is cut by e and lies below e. Then the following property is valid:

Property 3. *For each of the four sets of inner or outer vertex bisectors of P or Q, the function F increases (on the interval of bisectors of P that intersect Q) as the angle of the bisecting line increases, and the intersections of the vertex bisectors with the other polygon are ordered on the boundary of that polygon.*

The proof is based on Property 1 and is illustrated in Fig. 1. According to Property 1, two vertex bisectors p_i and p_{i+1} of P must intersect inside P. Therefore, if p_{i+1} intersects Q, then $F(p_{i+1}, Q) \geqslant F(p_i, Q)$.

Now Problem 1 can be solved in the following way:

Step 1. Compute areas $m(P_1P_2 \dots P_i)$ for each vertex P_i ($3 \leqslant i \leqslant n'$) of P and $m(Q_1Q_2 \dots Q_i)$ for each vertex Q_i ($3 \leqslant i \leqslant n''$) of Q (by a prefix sum, adding the area of a new triangle at each step).

Step 2. Find vertex bisectors for both polygons P and Q (Problem 3).

Step 3. Find intersections of vertex bisectors of P with polygon Q (this can be done by a linear scan on the boundary of Q, due to Property 3) and vice versa.

Step 4. Find $F(e, Q)$ for each vertex bisector e of P and $F(e, P)$ for each vertex bisector e of Q. For instance, let P_i and P_j be the two vertices of P that lie immediately below a given vertex bisector e of Q (these two points are found in the previous step). Then

$$m(P_jP_{j+1} \dots P_{i-1}P_i) = m(P_1P_2 \dots P_i) - m(P_1P_2 \dots P_j) - m(P_1P_jP_i)$$

while

$$F(e, P) = m(P_jP_{j+1} \dots P_{i-1}P_i) + m(P'P''P_iP_j),$$

where P' and P'' are the two intersection points of e with P (see Fig. 2).

Step 5. For each of the four sets of inner and outer bisectors of $Q(P)$ determine vertex bisector e for which $F(e, P)$ ($F(e, Q)$) has a minimal value but no less than $m(P)/2$ ($m(Q)/2$, respectively). For instance, in Fig. 1,

$$F(p_i, Q) < m(Q)/2 = F(t, Q) \leqslant F(P_{i+1}, Q)$$

and $e = p_{i+1}$ for the set of inner bisectors of P (t is the solution line, i.e. bisecting line for both P and Q). Since t and p_{i+1} intersect inside P (Property 1) and $F(p_i, Q) < F(t, Q)$, the portion of p_i that lies in the halfplane bounded by v and containing Q is below t and thus P_i is above t. Similarly P_{i+1} is below t, hence t intersects edge

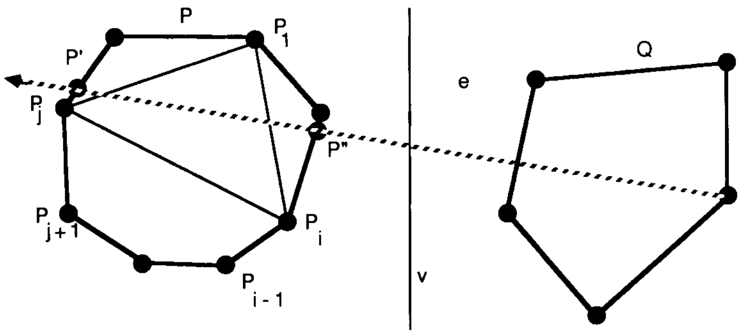

Fig. 2.

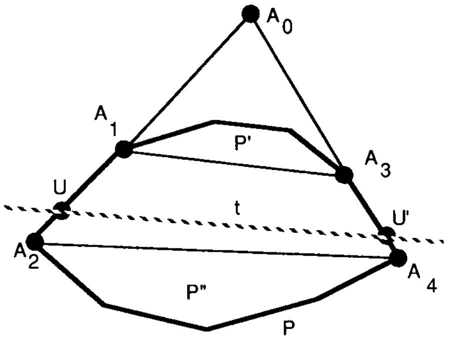

Fig. 3.

P_iP_{i+1}. Thus one can find two edges of P and two edges of Q that the solution line t must intersect.

Step 6. The exact position of t is then found by interpolation. The interpolation step can be solved by numerical or nonnumerical means.

A numerical solution is given in [7]. The nonnumerical solution leads to solving a polynomial of degree 4. Let A_1A_2 and A_3A_4 (B_1B_2 and B_3B_4) intersect at point A_0 (B_0, respectively) (see Fig. 3 where only P is given). Referring to Fig. 3, let

$$u = |A_0U| / |A_0A_2| \quad \text{and} \quad u' = |A_0U'| / |A_0A_4|.$$

Segments A_1A_3 and A_2A_4 divide P into three polygons P', P'', and quadrangle $A_1A_3A_4A_2$. Since t bisects P,

$$m(P') + m(UA_1A_3U') = m(P)/2$$

and thus $m(UA_1A_3U')$ is constant; hence the area $m(A_0UU')$ of triangle A_0UU' is a constant. However,

$$m(A_0UU') = |A_0U|\,|A_0U'| \sin\alpha = uu'\,|A_0A_2|\,|A_0A_4| \sin\alpha,$$

where α is the angle $A_1A_0A_3$. Therefore, the product uu' is constant, and $u' = c'/u$ for some c'. Analogously one can define v and v' of Q and conclude that $v' = c''/v$ for some constant c''. In the special case of parallel edges A_1A_2 and A_3, A_4,

$$u = |A_1U| / |A_1A_2| \quad \text{and} \quad u' = |A_3U'| / |A_3A_4|,$$
$$m(UA_1A_3U') = m(A_1UU') + m(A_1A_3U')$$

which gives a linear relationship between u and u': $c_1u + c_2u' = c_3$ for some constants c_1, c_2, and c_3. An analogous relation holds in the case of parallel edges B_1B_2 and B_3B_4. The points U, U', V, and V' are collinear, which allows v to be expressed as a function of u. A careful but tedious calculation (which we omit; see [7] for details) leads to a polynomial of degree 4 (with real coefficients) in u, which can be solved in constant time. This completes the solutions of Problem 1.

It is clear that all steps of the above algorithm can be implemented in linear sequential time. The correctness of the algorithm follows easily from the obvious properties that are listed.

The case of convex polyhedra

We now consider the three-dimensional version of our main problem.

Problem 4. Given three convex disjoint polyhedra T_1, T_2, and T_3, find a plane that splits each of them into two subpolyhedra of equal volume (i.e. find a plane that simultaneously bisects T_1, T_2, and T_3).

It can be shown (by using the continuity of functions separating volumes) that there exists exactly one such plane if there is no straight line that intersects all three polyhedra. Otherwise there may be more than one solution. An $O(n^4 \log n)$ algorithm for determining the existence of such a line for a set of polyhedra, called line transversal, is given in [1], where n is the total number of vertices of all polyhedra. The existence of a line transversal for three convex polyhedra T_1, T_2, and T_3 can be detected faster, in $O(n \log n)$ time, by the following procedure. There is no line transversal for T_1, T_2, and T_3 if and only if each of the three polyhedra can be separated (by a plane) from the other two (a proof is given in [7]). A procedure to check the condition may find the convex hulls of the union of any two polyhedra (in linear time [6]) and detect separability with the third one (in $O(n \log n)$ time [2]). If a line transversal does not exist, our algorithm below finds the ham-sandwich cut in

$$O(n \log n + nK^3)$$

time, where K is the number of output bits of accuracy in the solution; otherwise the algorithm does not guarantee to find any solution. The algorithm uses a numerical search to obtain the desired accuracy. The solution will be presented by considering several related problems.

Problem 5. Given a convex polyhedron T and a plane h, find the volumes of the two parts obtained by cutting T with h.

These volumes can easily be found in linear time (a procedure is given in [7]).

The following property will be used in the sequel, and can be proved easily (similar to Property 1):

Property 4. *The intersection of any two planes that bisect a convex polyhedron T is a line that intersects T.*

Problem 6. Given a straight line e which does not intersect a convex polyhedron T, find a plane h containing e that bisects T.

Problem 6 has a unique solution (due to Property 4) which can be found in the following way:

Step 1. Find two planes containing e that are tangent to T. This can be done in linear time [4,7].

Step 2. Apply numerical binary search. At each step test a plane containing e, and find the volumes of the two parts obtained by cutting T with the plane (Problem 5). Depending on the result of the test choose the next test plane following the goal of making volumes equal. This interpolation requires $O(K)$ steps. Since each step may be done in $O(n)$ time, the interpolation requires $O(nK)$ time. Therefore, Problem 6 can be solved in $O(nK)$ time.

Problem 7. Given a point S in space such that there is no straight line that contains S and intersects both polyhedra T_1 and T_2, find a plane h that passes through S and simultaneously bisects both polyhedra T_1 and T_2.

Its solution goes as follows:

Step 1. Find a plane τ that contains S and separates T_1 from T_2. One such plane can be found in $O(n \log n)$ time as follows:

- merge T_1 and T_2 to obtain the convex hull $CH(T_1 \cup T_2)$ of the union of T_1 and T_2 (this can be done in linear time [6]),
- find any plane τ' that separates S from $CH(T_1 \cup T_2)$ (in $O(n \log n)$ time [3]),
- project T_1 and T_2 onto τ' using point (central) projection from S; let polygons T_1' and T_2' be the projections of T_1 and T_2, respectively,
- due to the condition of Problem 7, T_1' and T_2' do not intersect each other; determine in linear time (see [4] and [7]) a line l that separates them,
- plane τ determined by S and l separates T_1 from T_2 as required.

Step 2. Find the intersection Q of τ with $CH(T_1 \cup T_2)$. Q is a convex polygon which does not contain S. Find tangents from S to Q. These tangents determine the interval of angles for numerical binary search. To show this, note that if a bisecting plane h for T_1 intersects τ at a line which is outside the interval, then h does not even intersect T_2 (not to mention the need to bisect it) because, in the opposite case, h must contain a point from both T_1 and T_2 and therefore the line passing through these points must intersect Q. This step requires $O(n)$ time.

Step 3. Apply a numerical binary search. At each step test a line e from the above interval (e belongs to τ and contains S), having chosen an angle, and find a plane h containing e that bisects T_1 (Problem 6). Find the volumes of the two parts obtained by cutting T_2 with h (Problem 5). Depending on the result of the test choose the next angle e following the goal of making the volumes of the two parts of T_2 equal. The interpolation takes $O(K)$ choices of e to achieve K bits of accuracy. Since Problems 6 and 5 require $O(nK)$

and $O(n)$ time (respectively), Step 3 requires $O(nK^2)$ time.

Therefore, Problem 7 can be solved in

$$O(n \log n + nK^2)$$

time. The correctness of the numerical binary search follows from the following observation. Due to Property 4, the intersection of any two planes (containing S) that bisect T_1 is a line that passes through S and intersects T_1. This line does not intersect T_2 (the condition of Problem 7). Thus, the portions of the volume of T_2 that are cut by bisecting planes for T_1 are ordered in the same fashion as the intersections of bisecting planes with τ.

Now the solution of the main problem (Problem 4) can be presented as follows:

Step 1. For a given vertex S from T_3, do Step 1 of Problem 7. Let l be the intersection line of τ and τ' (see Fig. 4 where all objects are orthogonally projected onto a plane perpendicular to both τ' and τ). For simplicity, we choose l to be a vertical line. The orthogonal line l' from S to l is then a horizontal line. If L is a point from l, the angle of the segment SL is defined as the angle between SL and l'.

The numerical search in the next step is based on the following property (valid under the condition of Problem 4):

Property 5. *The volumes of the portions of T_3 that are cut by bisecting planes for both T_1 and T_2 are ordered in the same manner as the intersections of these bisecting planes with the line l.*

This property follows from the observation that any two bisecting planes for T_1 and T_2 intersect at a line passing through T_1 and T_2 (Property 4). This line cannot intersect T_3 (condition of Problem 4) nor the halfplane of plane τ bounded by l and containing a point from T_3 (since τ' separates T_3 from T_1 and T_2; see Fig. 4). Therefore, bisecting planes for T_1 and T_2 are ordered on T_3 in the same manner as their intersections with l. This allows a numerical search for a ham-sandwich cut in the following step.

Step 2. Apply a numerical binary search (starting with zero angle; the interval for search is $(-\pi, \pi)$). At each step test a point L from l, having chosen angle (angle SL is defined above), and find a plane h containing L that bisects T_1 and T_2 (Problem 7; Step 1 may be omitted since the same plane τ will always do). Find the volumes of the two parts obtained by cutting T_3 with h (Problem 5). Depending on the result of the test choose the next middle angle SL following the goal of making the volumes of the two parts of T_3 equal. The interpolation takes $O(K)$ choices of L, and for each such choice, solving Problem 7 (without Step 1) takes $O(nK^2)$ time. Therefore Step 2 requires $O(nK^3)$ time.

The time complexity of the described algorithm for Problem 4 is

$$O(n \log n + nK^3).$$

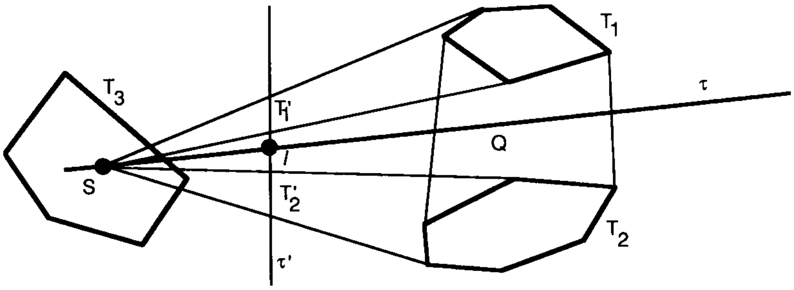

Fig. 4.

The correctness of the algorithm follows easily from Property 5.

Other measures and cutting proportions

A monotone measure for convex polygons is any measure m such that $m(Q \cup R) \leqslant m(Q) + m(R)$. Examples of such measures are area, perimeter and the number of vertices. The generalization of Problem 1, using measure m and any proportion of splitting (instead of even splitting), can be solved in linear time, along the same lines as Problem 1. Analogous generalization (of the volume and even splitting) may be made for the case of polyhedra.

Conclusion

A preliminary version of this paper has recently been presented at the 2nd Canadian Conference in Computational Geometry [8]. At the same conference, Diaz and O'Rourke [2] presented an $O(n \log n)$ algorithm for bisecting a simple polygon and ham-sandwich cuts of two simple polygons. Their ham-sandwich cut algorithm is based on a numerical search while our algorithm for convex disjoint polygons is nonnumerical. Thus finding a nonnumerical ham-sandwich cut algorithm for simple polygons remains for further investigation.

There are a number of other open problems which arise from the presented material. We mention some of them here. It may be of interest to solve Problem 1 if the convex polygons may intersect. What is the set of intersection points of two or more bisecting lines of a convex polygon (area kernel)? In what cases does the area kernel of a convex polygon consist of one point only, i.e. under what condition do all bisecting lines intersect at the same point? These problems may be generalized to higher dimensions.

Acknowledgment

The author appreciates comments given by Joseph O'Rourke and Jorge Urrutia on an early version of the paper, and suggestions given by both referees and Mikhail J. Atallah that have significantly improved the clarity and presentation of the paper.

References

[1] D. Avis and R. Wenger, Polyhedral line transversals in space, *Discrete Comput. Geom.* **3** (1988) 257–265.

[2] M. Diaz and J. O'Rourke, Ham-sandwich sectioning of polygons, in: *Proc. 2nd Canadian Conf. in Computational Geometry*, Ottawa (1990) 282–286.

[3] D.P. Dobkin and J.I. Munro, Efficient uses of the past, in: *Proc. 21st Symp. on Foundation of Computer Science* (1980) 200–206.

[4] H. Edelsbrunner, *Algorithms in Combinatorial Geometry* (Springer, Berlin, 1987).

[5] N. Megiddo, Partitioning with two lines in the plane, *J. Algebra* **6** (1985) 430–433.

[6] F.P. Preparata and S.J. Hong, Convex hulls of finite sets of points in two and three dimensions, *Comm. ACM* **20** (1977) 87–93.

[7] I. Stojmenović, Bisections and ham-sandwich cuts of convex polygons and polyhedra, Tech. Rept. TR-90-42, Computer Science Dept., University of Ottawa, 1990.

[8] I. Stojmenović, Cutting the volumes of convex polyhedra by a plane, in: *Proc. 2nd Canadian Conf. in Computational Geometry*, Ottawa (1990) 291–295.