CUTTING THE VOLUMES OF CONVEX POLYHEDRA BY A PLANE

Ivan Stojmenović

Computer Science Department, University of Ottawa,
Ottawa, Ontario, Canada K1N 9B4
(This research is supported by the NSERC)

Abstract

We present a linear time sequential algorithm for finding a straight line that cuts each of two disjoint convex polygons into two parts of equal area. The solution can be generalized to other measures (for instance, perimeter) and other proportions of cutting. The problem is generalized in k dimensions, to find a hyperplane that cuts each of k disjoint convex polytopes into two parts of equal volume. If no $(k-2)$-dimensional space exist that intersects all polytopes then the problem can be solved in $O(n^k \log n)$ time.

Cutting convex polygons by a straight line

To the best of our knowledge, the problem was not previously studied in the literature. A related problem, cutting each of two regions by a straight line into two subregions containing the same number of points has been studied in [E,GM,M] (called also ham-sandwich problem). Instead of counting the number of points in each region, we here consider continuous measures like area and perimeter.

We will begin by studying the two-dimensional version of the following problem:

Problem 1. Given two disjoint convex polygons in the plane, find a straight line which cuts both polygons into two pieces of equal areas.

The problem is illustrated on Fig. 1 (where $m(P)$ refers to the area of P). It can be easily shown (using continuity arguments) that given any two disjoint convex polygons, there exist exactly one straight line that splits both of them evenly into two subpolygons of equal area.

We say that a given straight line q *partitions* a convex polygon P if q cuts P into two parts of equal areas. The straight line partition of P passing by a vertex V of P is called the *V-vertex partition* of P.

A natural first attempt to solve Problem 1 does not work: find the "centroid" of each polygon and draw a line through the centroids. Fig. 2 shows that no such centroid exist, since straight lines that partition given polygon do not necessarily share a common point. However, one can easily show the following property.

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

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 partition of P.

Let $P_1, P_2, \ldots, P_n$ be vertices of P listed in counterclockwise order. In linear time one can determine the areas of all triangles $P_1P_{i-1}P_i$, i.e. $m(P_1P_{i-1}P_i)$ $(3 \leq i \leq n)$. Applying a prefix sum technique, one can find $m(P_1P_2 \ldots P_{i-1}P_i)$, and detect a vertex P_i of P, with smallest possible index i, for which $m(P_1P_2 \ldots P_{i-1}P_i) \geq m(P)/2$, again in linear time (see Fig. 3). We refer to edge $P_{i-1}P_i$ and vertices P_{i-1} and P_i as being *opposite* edge and vertices for vertex P_1. Then the vertex partition for P_1 is found by "interpolation" within the opposite edge $P_{i-1}P_i$ of P_1, to get two pieces of equal areas. Here by interpolation we ment determining the point of intersection of the vertex partition of P_1 with its opposite edge (the interpolation takes constant time). It is easy to show that the scan procedure takes $O(n)$ time, where n is the number of vertices of P.

Problem 3. Construct V-vertex partitions for each vertex V of given convex polygon P.

If we choose the orientation of all vertex partitions from vertices toward interior of P then, on the basis of Property 1, one can show the following (see Fig. 4).

Property 2. Slopes of vertex partitions of P are ordered by the same circular order as vertices of P.

Using Property 2 one can solve Problem 3 by a linear scan around vertices of P, starting by a vertex P_1 (i.e. starting by solving Problem 2). In the scan vertex partitions are found in, say, counterclockwise order of vertices. Suppose the last vertex partition passed by vertex P_l and intersected the opposite edge $P_{i-1}P_i$. Find $m(P_{l+1}P_{l+2}...P_{i-1}P_i)$ from $m(P_lP_{l+1}...P_{i-1}P_i)$ 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 area. Then the partitioning straight line is found by interpolation (in constant time) within the corresponding opposite edge.

Thus Problem 3 can be solved in O(n) sequential time, which is optimal. This solution can be used to answer also the following query, given as Problem 4 (note that we do not need to solve Problem 4 to answer to Problem 1; it is given here since it may be of indenpendent interest).

Problem 4. Given a straight line s and a convex polygon P, find a straight line t parallel to s such that t partitions P.

Problem 4 can be solved in linear time by constructing all vertex partitions of P (Problem 3) and finding (by two binary searches) two edges of P such that the slope of s is between the slopes of corresponding vertex partitions of endpoints of the edges. Then t intersect these two obtained edges and its exact position can be found by a numeric interpolation (similar interpolation will be described below for solving Problem 1).

Suppose we are given two disjoint convex polygons P and Q (without loss of generality, we assume that P and Q are separable by a vertical line; in the sequel the relations "above" and "below" are well defined with respect to two orientations on the vertical line). Our goal is to find a straight line t that partitions both of them (Problem 1). We divide all vertex partitions into two groups: inner and outer, according to whether or not corresponding oriented half-lines intersect the vertical separating line (each half-line starts at corresponding vertex and penetrates the interior of corresponding polygon; see Fig. 4 and Fig. 6). Given a line t and convex polygon Q let $F(t,Q)$ be the area of the fraction of Q that is cut by t and lies below t. Then the following property is valid.

Property 3. For each of four sets of inner and outer vertex partitions of P and Q the function F increases as the slope of partitioning line increases.

The proof is based on Property 1 and is illustrated on Fig. 6.

Now we turn our attention to solving Problem 1. First we find vertex partitions for both polygons (Problem 3). Then we use Property 3 to locate desired straight line t between two neighboring inner and two neighboring outer vertex partitions of P. Due to this property, a vertex partition of P will be below desired partition line t (the relation "below" is well defined inside polygon Q) if it cuts less than half area out of Q Applying Property 1, intersections of vertex partitions of P with polygon Q are ordered on the perimeter of Q which allows linear search for them. In addition, the area $m(P_jP_{j+1}...P_{i-1}P_i)$ can be determined in constant time from the areas $m(P_1P_2...P_i)$, $m(P_1P_2...P_j)$ and $m(P_1P_jP_i)$ ($m(P_jP_{j+1}...P_{i-1}P_i)= m(P_1P_2...P_i) - m(P_1P_2...P_j) - m(P_1P_jP_i)$). On the other hand, if m is the perimeter then $m(P_jP_{j+1}...P_{i-1}P_i)=m(P_1P_2...P_i) - m(P_1P_2...P_j) - l(P_1P_i) + l(P_1P_k) + l(P_iP_j)$, where $l(AB)$ denotes the lenght of the straight line segment AB (see Fig. 5). This enables determining the area of both pieces obtained by cutting Q with vertex partitions of P in constant time, ones the intersection points with Q are found. Thus there is unique pair of neighboring vertex partitions such that the slope of t is between their slopes. Moreover, t will intersect the edge incident to these vertices. In fact, there are two such pairs of neighboring vertices, corresponding to two edges of P that t intersect. Analogously one can determine two edges of Q intersected by t. The exact position of t is then found by interpolation, which in this case is equivalent to solving a polynom of degree 4 (with real coefficient) and therefore takes constant time to solve it (by a formula or by numeric means).

It is clear that all steps of the former algorithm can be implemented in linear sequential time. Therefore two disjoint convex polygons can be cut by a straight line each into two pieces of equal area in optimal $O(n)$ time, where n is total number of vertices of P and Q

For readers that are familiar with parallel models of computation we note that all mentioned problems can be solved in O(log n) time on a CREW PRAM with n processors (cf. [A] for details on CREW PRAM).

Plane partitions of convex polyhedra

We now consider the three and higher dimensional version of our main problem. The obtained solutions are not generalizations of the solution in 2-D case since the properties that enabled optimal solutions in two dimensional space do not hold in higher dimensions. Thus we describe new solution for the three dimensional case; the solution can be generalized to higher dimensions in a rather straightforward way. However, the algorithms are not optimal. In three dimensions the problem can be stated as follows.

Problem 5. Given convex disjoint polyhedra P_1, P_2, and P_3, find a plane that splits each of them evenly into two subpolyhedra of equal volume (i.e. find a plane that simultaneously partitions P_1, P_2, and P_3).

It can be shown (by using the continuity of functions separating volumes) that there exist exactly one such plane if the following condition is satisfied: there is no straight line that intersect 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 (called also line transversal) is given in [AW] (n is total number of vertices of three polyhedra). If a line transversal does not exist, our algorithm below finds the solution in $O(n^2 \log n)$ time, where n is total number of vertices on all polyhedra; otherwise the algorithm does not garranty to find any solution. The solution will also be presented by solving several related problems.

Problem 6. Given a straight line p which does not intersect a convex polyhedron P, find a plane c passing by p that partitions P.

Problem 6 can be solved in $O(n \log n)$ time (n being the number of vertices of P) in the following way:

- Sort vertices of P around p in (say) clockwise order.
- Apply binary search (in $O(\log n)$ steps). At each step test a plane passing by p and a vertex of P, and find the volumes of two parts obtained by cuting P with the plane. The test takes $O(n)$ time since triangulation of a convex polyhedron can be done in linear time. Depending on the result of the test chose the next middle point following the goal of making volumes equal.
- After the binary search all vertices of P are divided into two sets such that the partitioning plane of P separates them. To find exact position of the partitioning plane h, an interpolation step should be done. Plane h has one degree of freedom, since it passes through p (say, the slope according to a chosen direction normal to p). The edges of P connecting two vertices of P that are separated by h are intersected by h; the intersection is expressed in the same variable. The two volumes obtained by cutting with h are expressed as qubic polynomial in the same variable, which can be solved in constant time. This completes solution of Problem 6.

Problem 7. Given a point S in space (such that there is no straight line that passes through S and intersects both polyhedra P_1 and P_2), find a plane h that passes through S and simultaneously partitions both polyhedra P_1 and P_2.

Solution of the Problem 7 goes as follows:

-For each vertex A_i of P_2 find the partition plane h_i for P_1 that passes through S and A_i.

-Find the volumes of two pieces of P_2 obtained by cutting P_2 with plane h_i. On the basis of the obtained relative sizes of volumes one can decide on which side of desired partitioning plane h the point A_i lies.

-The tests done in former step separate points from P_2 into two sets such that the separation is the same as done by desired partitioning plane h.

- Repeate above steps with the role of P_1 and P_2 interchanged.

- The exact position of h is obtained by interpolating between points from P_1 and P_2. The interpolation now involves the plane with two unknown variables leading to a polynom of degree greater than four in the final equation. Thus, the solution of the polynom is done by numeric means (Newton method, for example). We assume constant time for the solution (otherwise the final time complexity should be multiplied by the time to find the roots of a constant degree polynom).

The time needed to solve the Problem 7 is $O(n^2 \log n)$.

Now the solution of the main problem (Problem 5) can be presented as follows.

- For each point B_i from P_3 find the partitioning plane h_i for both P_1 and P_2 passing by B_i (Problem 7).

- Find the volumes of two pieces of P_3 obtained by cutting P_3 with h_i, and decide on the basis of the test results about the position of B_i with respect to the desired partition plane for P_1, P_2, and P_3.
- Perform the same procedure for vertices of two other polyhedra.
-Interpolate the partition plane for P_1, P_2, and P_3 by solving a system of equations that is equivalent to finding the root of a constant degree polynom.

The time complexity of the described algorithm to solve Problem 5 is $O(n^3 \log n)$ (multiplied by the time needed to find a root of a constant degree polynom by numeric methods). It can be implemented on a CREW PRAM with $O(n^3)$ processors to run in $O(\log n)$ time.

Extension to higher dimensions

Problem 8. Given k convex polytopes in k-dimensional space such that no $(k-2)$-dimensional space exist that intersects all polytopes, find a hyperplane that partitions all polytopes.

It can be shown that such a partition is unique, and can be obtained by generalizing the process described for 3-dimensional case. The time complexity will be $O(n^k \log n)$.

Other measures and cutting proportions

Our results can be easily generalized for other measures (not only area) and other proportions of splitting (not only in two equal halves).

The monotone measure of convex polygons is any measure m such that if a convex polygon P is equal to the union of two convex polygons Q and R then $m(P) \leq m(Q) + m(R)$. Examples of such measures are area, perimeter and the number of vertices. Using the same approach one can solve the following problem: given two disjoint convex polygons P and Q draw a straight line which partitions P (Q) into two parts P_1 and P_2 (Q_1 and Q_2, respectively) such that $m(P_1)=\alpha\ m(P_2)$ and $m(Q_1)=\beta\ m(Q_2)$, where α and β are two given positive reals. We presented a linear time sequential algorithm to solve the problem (assuming the measure of any triangle can be determined in constant time). When implemented in parallel, this leads to an $O(\log n)$ time algorithm on a CREW PRAM with n processors, where n is the number of vertices of P and Q

To be more precise, the time complexities (for area as measure and even splitting) should be multiplied by $T(n)$, where $T(n)$ is the time needed to solve the following problems for a given measure m: the interpolation problem in a convex 4-gon (quadrangle); finding the measure of union of two convex polygons that share an edge; finding the measure of a triangle etc. This time is constant for the area and the perimeter as measures.

Open problems

There a number of open problems which arise from the presented material. We mention some of them. It may be of interest to solve Problem 1 if the convex polygons may intersect and Problems 1-4 for simple polygons. What is the set of intersection points of two or more partitioning lines of a convex polygon (area kernel)? In which case the area kernel consists of one point only, i.e. under what condition all partition lines intersect at the same point? These problems may be generalized to higher dimensions. Finally, find a more efficient algorithm to solve three and higher dimensional version of Problems 5-8.

Acknowledgements

We appreciate comments given by Joseph O'Rourke and Jorge Urrutia on an early version of the paper.

References

[A] S.G. Akl, The Design and Analysis of Parallel Algorithms, Prentice Hall, 1989.
[AW] D. Avis, R. Wenger, Polyhedral line transversals in space, Discr. and Comp. Geom., 3, 257-265, 1988.
[E] H. Edelsbrunner, Algorithms in Combinatorial Geometry, Springer-verlag, 1987.
[GM] P. Gburzynski, M. McLeish, An efficient heuristic algorithm for a 3-space n-partitioning problem used in polygon retrieval and non-parametric statistics, Proc. Allerton Conf., 1987, 908-916.
[M] N. Megiddo, Partitioning with two lines in the plane, J. Alg., 6, 430-433, 1985.

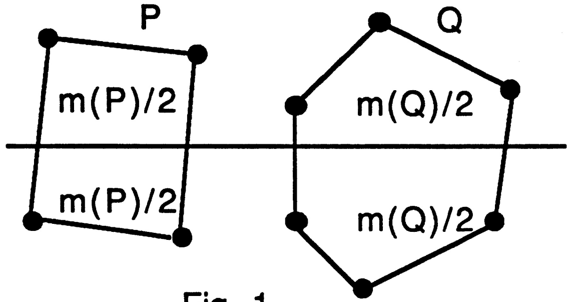

Fig. 1.

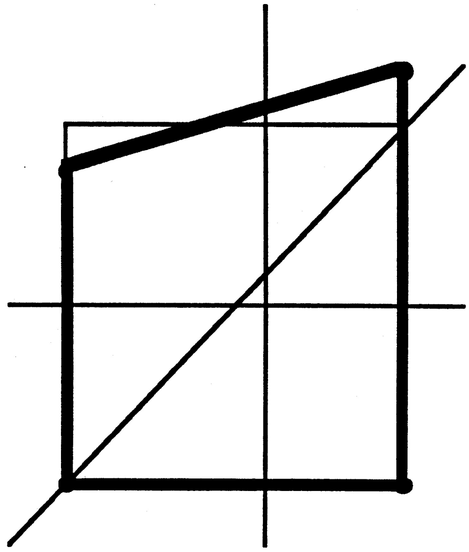

Fig. 2

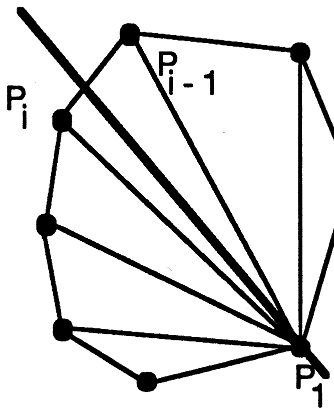

Fig. 3

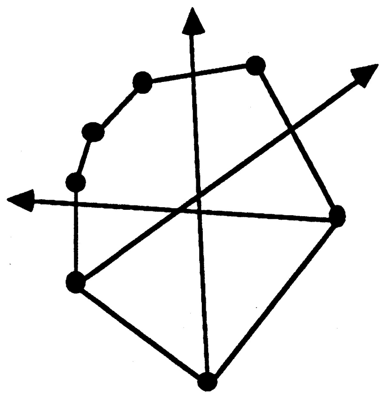

Fig. 4

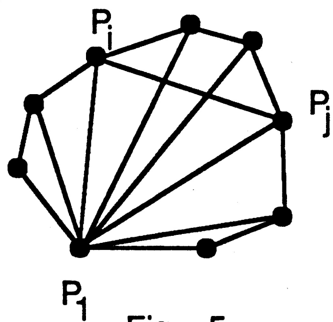

Fig. 5

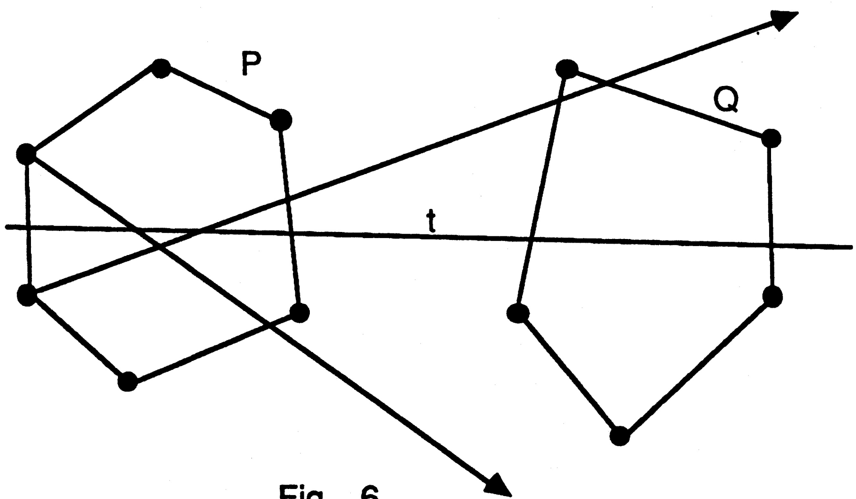

Fig. 6