Largest and Smallest Area Triangles on a Given Set of Imprecise Points

Vahideh Keikha* Maarten Löffler† Ali Mohades *

Abstract

In this paper we study the following problem: we are given a set of imprecise points modeled as parallel line segments, and we wish to place three points in different regions such that the resulting triangle has the largest or smallest possible area. We first present some facts about this problem, then we show that for a given set of line segments of equal length the largest possible area triangle can be found in $O(n \log n)$ time, and for line segments of arbitrary length the problem can be solved in $O(n^2)$ time. We also show that the smallest possible area triangle for a set of arbitrary length line segments can be found in $O(n^2)$ time.

1 Introduction

In this paper we study a traditional problem in computational geometry in an imprecise context. Let P be a set of points. We wish to find a sequence of k points in P such that if we connect them into a polygon Q, Q has specific attributes, e.g., Q has the largest or smallest possible area or perimeter, or Q is an empty k-gon, etc. Of course for a given set P such a k-gon does not necessarily exist, for $k > 3$.

1.1 Related Work

Numerous papers studied such problems previously. Dobkin and Snyder [5] presented a linear time algorithm for finding the largest area triangle inscribed in a convex polygon.

Boyce *et al.* [3] presented a dynamic programming algorithm for finding the largest possible area and perimeter convex k-gon on a given set of n points in $O(kn \log n + n \log^2 n)$ time. Aggarwal *et al.* [2] improved their result to $O(kn + n \log n)$.

Due to more applicability, there are more studies concerned with the problem of finding the minimum possible area and perimeter k-gon. Dobkin *et al.* [4] presented an $O(k^2 n \log n + k^5 n)$ time algorithm for finding minimum perimeter k-gons. Their algorithm was improved upon by Aggarwal *et al.* to $O(n \log n + k^4 n)$ time [1]. Eppstein *et al.* [7] studied three problems: finding the smallest possible k-gon, finding the smallest empty k-gon, and finding the smallest possible convex polygon on exactly k points, where the smallest means the smallest possible area or perimeter. They presented a dynamic programming approach with $O(kn^3)$ time and $O(kn^2)$ space, that can also solve the maximization version of the problem as well as some other related problems. Afterwards, Eppstein [8] presented an algorithm for minimum area k-gon problem that runs in $O(n^2 \log n)$ time and $O(n \log n)$ space for constant values of k.

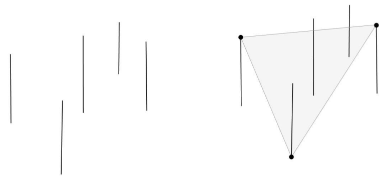

Figure 1: The largest possible area triangle for a set of line segments.

1.2 Problem Definition

We are given a set $L = \{L_1, L_2, \ldots, L_n\}$ of imprecise points modeled as parallel line segments, that is, every segment L_i contains exactly one point $P_i \in L_i$. This gives a point set $P = \{P_1, P_2, \ldots, P_n\}$, and we want to find the largest area or smallest area triangle in P, T_{max} and T_{min}. But because L is a set of imprecise points, we do not know where P is and what could be the possible values of the area. But there should be a lower bound and an upper bound and we are interested in computing these values. So, in this paper we compute the largest possible area of T_{max} and the smallest possible area of T_{min}. We named these problems *MaxMaxArea* and *MinMinArea*, respectively. An illustration of these problems can be seen in Figure 1. In this example, the solution of MinMinArea is zero, as we can find three collinear points.

1.3 Results

We show that

- MaxMaxArea can be solved in $O(n \log n)$ time and $O(n^2)$ time, respectively, for a given set of equal length and arbitrary length parallel line segments, and

*Laboratory of Algorithms and Computational Geometry, Department of Mathematics and Computer Science, Amirkabir University of Technology, Tehran, Iran, va.keikha@aut.ac.ir, mohades@aut.ac.ir

†Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands, m.loffler@uu.nl

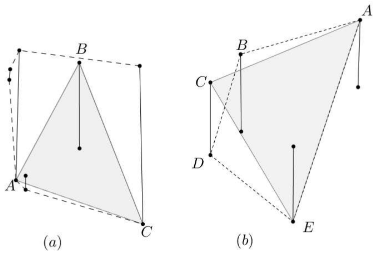

Figure 2: (a) The maximum area true triangle selects its vertices from the endpoints of regions, but not necessarily on the convex hull. (b) Maximum area convex hull does not contain maximum area triangle.

- MinMinArea can be solved in $O(n^2)$ time.

2 Preliminaries

In this section we first present related previous results that may be applicable to our problem, and then discuss some difficulties that occur when dealing with imprecise points.

Boyce *et al.* [3] defined a *rooted* polygon as a polygon with one of its vertices fixed at a given point. In the context of imprecise points, we define the *root* as a given point in a specific region. In this case we throw out the remainder of the root's region and try to find the other vertices of the k-gon in the other $n-1$ remaining regions, and a *rooted* polygon will be a polygon with one of its vertices fixed at a given point in a specific region. Boyce *et al.* [5] showed that the rooted largest area triangle can be found in linear time. They showed that the largest area k-gon only uses points on the convex hull (if there exist at least k points on the convex hull); also Löffler and van Kreveld [9] proved that the maximum area convex polygon always selects its vertices from the endpoints of the line segments, so one may think that the maximum area triangle selects its vertices from the endpoints of regions on the convex hull. But that it is not the case, as can be seen in Figure 2(a).

Also unlike in the precise context, the largest area triangle is not inscribed in the largest possible convex hull of the given set of imprecise points, as illustrated in Figure 2(b).

This problem is more complicated for larger values of k, as illustrated in Figure 3(a); even for k=4, we cannot find the area of the largest strictly convex k-gon, as the angle at a approaches π and we can enlarge the area of the convex 4-gon arbitrarily.

3 Maximum Area Triangle

In the following we first define some notation that we will use in subsequent sections. Let Z be the set of all endpoints of L, $Z = \{L_1^-, L_1^+, L_2^-, L_2^+, \ldots, L_n^-, L_n^+\}$, where L_i^+ denotes the upper endpoint of L_i, and L_i^- denotes the lower endpoint of L_i. We define $C_0 = CH(Z)$ as the convex hull of Z, and $C_1 = CH(Z \setminus C_0)$. By *true triangle* we mean a triangle constructed on three different regions. We assume the segments to be oriented vertically.

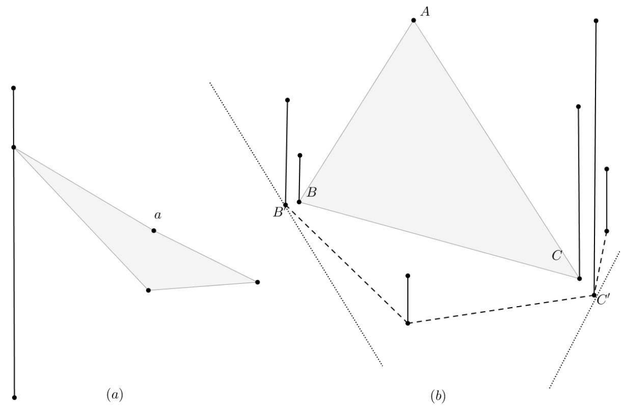

Figure 3: (a)The maximum area convex 4-gon constructed on a set consists of one imprecise and three fixed points, where the inner angle at a approaches π. (b)The largest area true triangle selects at least two vertices from the vertices of C_0.

Observation 1 *If at most two separate regions appear on C_0, there is an optimal solution to the MaxMaxArea problem, such that all the vertices are chosen at endpoints of the line segments, and the two regions which appear on C_0, always appear on the largest possible area true triangle.*

In this case, the largest possible area true triangle can be found in $O(n)$ time. From now on assume more than two different regions appear on C_0.

3.1 Stable Triangle

Dobkin and Snyder [5] defined a *stable* triangle ABC as a triangle with the root A fixed, such that forward advancement of either B or C along the convex hull results in a smaller area, and proved the *Pentagon* lemma (see full version), that is needed for the correctness proof of their algorithm. The area maximizing triangle will be chosen from one of these stable triangles. The idea of their algorithm is to start searching from three consecutive vertices A, B and C on the convex hull. They move C forward until the movement reduces the area, and then move B forward, and again move C, etc. If moving either of them would reduce the area, the triangle is stable and, they move A forward. When A returns to its starting position, they stop the algorithm and report the largest area stable triangle they found. This algorithm has linear running time (see full version for details).

In our imprecise context we define a *stable* triangle in the same way, as a triangle such that forward advancement of either B or C results in a smaller area triangle, but it can be a non-true triangle. So we should be careful that not all the stable triangles are non-true triangles, because then we do not find the largest true triangle among them.

In the following we first show that if we have a convex polygon with vertices from repeated regions (each region appears on the convex hull at most two times), we still can find the solution of MaxMaxArea problem in linear time, then we use this result for designing our algorithms.

4 Largest Area True Triangle

Let ABC be the initial triangle during the execution of the algorithm. Without loss of generality we may assume ABC is a true triangle (as we assume more than two different regions appear on C_0). So, the area of ABC is the initial value of T_{max}. We continue the algorithm naturally, but our movements may result in repeated regions. In full version we show what we do when we encounter repeated regions, and analyse all possible cases that cause a stable non-true triangle. Also, we disscuss the stable true triangle that we accept in each case.

Assume we are given a convex polygon $S = \{s_1, s_2, \ldots, s_n\}$. Similar to the notation of [5], we denote by $\alpha(s_a s_b s_c)$ the stable true triangle which we found during a step of the algorithm, where we started searching from $s_a s_b s_c$. For an arbitrary point $A = s_a$, define the A-rooted maximum true triangle to be $\alpha(s_a s_{a+1} s_{a+2})$. The following lemma states that for the $A+1$-rooted maximum true triangle, it is unnecessary to begin with the collapsed triangle $s_{a+1} s_{a+2} s_{a+3}$. of the A-rooted maximum.

Lemma 1 *If* $\alpha(s_a s_{a+1} s_{a+2}) = (s_a s_b s_c)$, *then* $\alpha(s_{a+1} s_{a+2} s_{a+3}) = \alpha(s_{a+1} s_b s_c)$.

Lemma 2 *Let* $S = \{s_1, s_2, \ldots, s_n\}$ *be a convex polygon, with vertices from repeated regions. There exist an* i $(1 \le i \le n)$ *such that an area maximizing true triangle on* s_i *is the area maximizing true triangle inscribed in* S.

Corollary 3 *Let* L *be a set of imprecise points modeled as a set of parallel line segments with arbitrary length. The largest possible area true triangle which selects its vertices from the vertices of* C_0 *can be found in* $O(n \log n)$ *time.*

4.1 Equal Length Parallel Line Segments

From the previous section we understand that if we prove that all the candidates points of the vertices of the largest possible area true triangle appear on C_0, and we know that all the possible stable triangles are true triangles, we can directly apply the existing algorithm [5].

Lemma 4 *Let* L *be a set of equal length parallel line segments. The largest possible area true triangle selects its vertices from the vertices on* C_0.

In case of equal length parallel line segments, when all the upper (and lower) endpoints of the line segments are collinear together, the maximum possible area triangle can be a non-true triangle. In this situation the largest possible area triangle would be constructed on the leftmost and rightmost line segments, and the largest possible area true triangle can be found in linear time. We can determine this situation in $O(n)$ time.

Lemma 5 *Let* L *be a set of equal length parallel line segments, that is, all the lower (or upper) endpoints are not collinear. The largest pssible area triangle is always a true triangle.*

Theorem 6 *Let* L *be a set of imprecise points modeled as a set of parallel line segments with equal length. The solution of the problem MaxMaxArea can be found in* $O(n \log n)$ *time.*

4.2 Arbitrary Length Parallel Line Segments

For simplicity we assume general position, that is, no two vertical line segments have the same x-coordinates. As we saw above, the largest possible area true triangle computed on a set of imprecise points modeled as arbitrary length parallel line segments does not necessarily select its vertices on the convex hull of the regions.

Lemma 7 *Let* L *be a set of imprecise points modeled as a set of parallel line segments with arbitrary length. At least two vertices of the largest area true triangle are located on* C_0 *and at most one of its vertices is located on* C_1.

4.2.1 Algorithm

Now we know the combinatorial structure of the largest possible area true triangle: it can select all of its vertices on C_0, or it selects two neighbor vertices on C_0 and one vertex on C_1, or it selects two non-neighbor vertices on C_0 and one vertex on C_1. The largest possible area true triangle is the largest area true triangle among them. In the first case, the largest area true triangle can be found in $O(n \log n)$ time using Corollary 3. In the second case, we try all the edges of C_0 as the base of the triangle. The third vertex can be found by doing a binary search on the

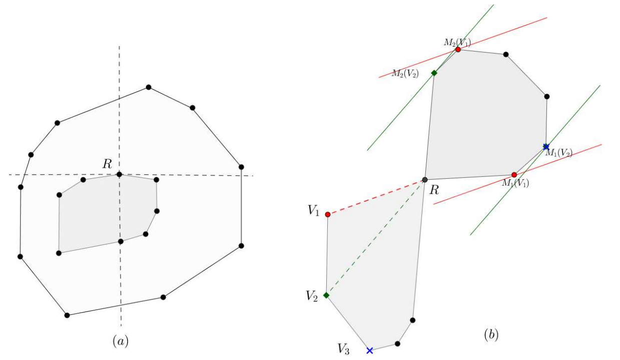

Figure 4: (a) Selection of a point on C_1 as the root R. (b) For the case of diagonal quadrants, for a given $R \in C_1$ and a fixed point V_i in quadrant three, we only need to look for the candidates of the largest area true triangle in one direction, and from $M_1(V_{i-1})$ and $M_2(V_{i-2})$ on C_0.

boundary of C_1. So in this case again we can find the maximum area true triangle in $O(n \log n)$ time. In the third case, assume each of the points of C_1 to be the origin point, R. For every point $R \in C_1$ as the origin, we partition C_0 into four quadrant convex chains, so that the largest area true triangle should be constructed on R and two points on the other quadrants (or only one quadrant), as illustrated in Figure 4(a). If one or two consecutive quadrants include the other vertices of the largest area true triangle, we can find the largest area true triangle in $O(n \log n)$ time by using Corollary 3. Suppose two other vertices are located on diagonal quadrants. Let the cyclic ordering of C_0 be counterclockwise, and let V_1 be the first vertex of quadrant three in the cyclic ordering of C_0. We first find the two candidates points in quadrant one for constructing the largest possible area true triangle on R and V_1, $M_1(V_1)$ and $M_2(V_1)$. For finding $M_1(V_2)$ and $M_2(V_2)$, we just need to start looking from $M_1(V_1)$ and $M_2(V_1)$, etc (see Figure 4(b)). In this case, we can find the largest possible area true triangle in $O(n^2)$ time (see full version for more details).

Theorem 8 *Let L be a set of imprecise points modeled as a set of parallel line segments with arbitrary length. The solution of the problem MaxMaxArea can be found in $O(n^2)$ time.*

5 Smallest Area True Triangle

In this problem, if we find three collinear points on three different input regions, the smallest area triangle would have zero area. We can understand this situation in $O(n^2)$ time. And we cannot hope to do it faster as the problem is 3SUM-hard. In the following we assume that MinMinArea has a non-zero solution.

Lemma 9 *Let L be a set of imprecise points modeled as a set of parallel line segments. Suppose there is no zero-area triangle in L. The smallest area true triangle selects its vertices on the endpoints of the line segments.*

We use the idea of the algorithm presented in [6] that solves the problem in dual space and on an arrangement of lines. As their duality preserves the vertical distances and is order preserving, the minimum area triangle on each vertex in dual space, can be constructed on a line that is located exactly above or below the vertex. In our problem, we need to continue looking in at most two neighbouring faces, when we encounter to repeated regions (see full version for details).

Theorem 10 *Let L be a set of imprecise points modeled as parallel line segments. The solution of the problem MinMinArea can be found in $O(n^2)$ time.*

Acknowledgements

This work was partially supported by the Netherlands Organisation for Scientific Research (NWO) under project no. 614.001.504.

References

[1] A. Aggarwal, H. Imai, N. Katoh and S. Suri. Finding k points with minimum diameter and related problems. *Journal of Algorithms*, 12: 38–56, 1991.

[2] A. Aggarwal, M. M. Klawe, S. Moran, P. Shor and R. Wilber. Geometric applications of a matrix-searching algorithm. *Algorithmica*, 2: 195–208, 1987.

[3] J. E. Boyce, D. P. Dobkin, R. L. Drysdale and L J. Guibas. Finding extremal polygons. *SIAM Journal on Computing*,14 (1), 134–147, 1985.

[4] D.P. Dobkin, R.L. Drysdale, and L.J. Guibas. Finding smallest polygons. *Computational Geometry, Advances in Computing Research*, 1: 181–214, 1983.

[5] D.P. Dobkin and L. Snyder. On a general method for maximizing and minimizing among certain geometric problems. *20th IEEE Symposium on Foundations of Computer Science (FOCS 1979)*, 9–17, 1979.

[6] H. Edelsbrunner, J. O'Rourke and R. Seidel. Constructing Arrangements of Lines and Hyperplanes with Applications. *SIAM Journal on Computing*, 15: 341–363, 1986.

[7] D. Eppstein, M. Overmars, G. Rote and G.J. Woeginger. Finding minimum area k-gons. *Discrete & Computational Geometry*, 7: 45–58, 1992.

[8] D. Eppstein. New algorithms for minimum area k-gons. *3th ACM-SIAM Symposium on Discrete Algorithms (SODA 1992)*, 83–88, 1992.

[9] M. Löffler and M. van Kreveld. Largest and Smallest Convex Hulls for Imprecise Points. *Algorithmica*, 56: 235–269, 2010.