SIAM J. COMPUT.
Vol. 28, No. 1, pp. 278–296

UNORIENTED Θ-MAXIMA IN THE PLANE: COMPLEXITY AND ALGORITHMS*

DAVID AVIS†, BRYAN BERESFORD-SMITH‡, LUC DEVROYE†, HOSSAM ELGINDY‡, ERIC GUÉVREMONT§, FERRAN HURTADO¶, AND BINHAI ZHU‖

Abstract. We introduce the unoriented Θ-maximum as a new criterion for describing the shape of a set of planar points. We present efficient algorithms for computing the unoriented Θ-maximum of a set of planar points. We also propose a simple linear expected time algorithm for computing the unoriented Θ-maximum of a set of planar points when $\Theta = \pi/2$.

Key words. maxima, plane sweep, lower bound, probabilistic analysis, expected complexity

AMS subject classifications. 68Q25, 60D05, 60C05

PII. S0097539794277871

1. Introduction.

The development of image processing has motivated the investigation of properties of point sets for the purpose of image classification and/or understanding. Connectivity graphs and various enclosing boundary sets have been used to characterize the shape of point sets. The minimum spanning tree, the Gabriel graph, and the Delaunay triangulation are important connectivity graphs. Convex, maximal, and α-hulls [KLP75, EKS83] are instances of boundary sets. The kth iterated hull [Ch85] and the related concept of k-hull [CSY87] have also been proposed.

In this paper we introduce unoriented Θ-maxima as a generalization of extreme and maximal vectors. These are useful as boundary descriptors, and remain invariant under rotation.

Let S be a set of n planar points. A ray from a point $p \in S$ is the collection of all points $\{p + \lambda(v - p) : \lambda > 0\}$, where v is a fixed point in the plane not equal to p. A ray from a point $p \in S$ is called a *maximal ray* if it passes through another point $q \in S$. A cone is defined by a point p and two rays A and B emanating from it: it is the convex set $\{\lambda u + (1 - \lambda)v : u \in A, v \in B, \lambda \in [0, 1]\}$. A point $p \in S$ is said to be a *maximum* (or *maximal*) with respect to S if there exist two rays, A and B, emanating from p such that A and B are parallel to the $+x$- and $+y$-axes, respectively (thus, $v = p + (1, 0)$ and $v = p + (0, 1)$ in the definition of A and B), and the points of S lie outside the ($\pi/2$-angle) cone defined by p, A, and B. A point $p \in S$ is an unoriented Θ-maximum with respect to S if and only if there exist two *maximal* rays, A and B, emanating from p with an angle at least Θ between them so that the points of S lie outside the (Θ-angle) cone defined by p, A, and B (see Figure 1). We let S_Θ denote the subset of S whose elements are unoriented Θ-maxima. For the remainder of this

*Received by the editors December 1, 1994; accepted for publication (in revised form) December 30, 1996; published electronically June 15, 1998.
http://www.siam.org/journals/sicomp/28-1/27787.html

†School of Computer Science, McGill University, Montreal, QC H3A 2A7, Canada (avis@cs.mcgill.ca, luc@cs.mcgill.ca).

‡Dept. of Computer Science, The University of Newcastle, NSW, Australia (bbs@cs.newcastle.edu.au, hossam@cs.newcastle.edu.au).

§School of Computing Science, Simon Fraser University, Burnaby, BC V5A 1S6, Canada.

¶Dept. de Mathemàtica Aplicada II, Universitat Politècnica de Catalunya, Barcelona, Spain (hurtado@ma2.upc.es).

‖Los Alamos National Laboratory, Group C-3, Los Alamos, NM 87545 and Dept. of Computer Science, City University of Hong Kong, Kowloon, Hong Kong (bhz@cs.cityu.edu.hk).

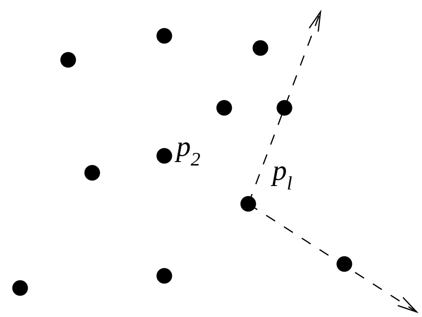

FIG. 1. *Point p_1 is an unoriented $\pi/2$-maximum whereas p_2 is not.*

paper, we only consider the problem of computing $S_{\pi/2}$. The algorithms apply to other values of $\Theta > \pi/2$. The additional technique for handling values of $\Theta < \pi/2$ is discussed in the appendix.

For each $p \in S_{\pi/2}$, our algorithms report two witnesses in the form of two maximal rays with an included angle, denoted by $\alpha_p \geq \pi/2$. Each of these maximal rays intersects the same edge of the convex hull of S and contains a ray parallel to either the x- or the y-axis in the cone between the two maximal rays. The above properties lead to two different approaches for computing $S_{\pi/2}$, which we outline in the following paragraphs.

1.1. Convex hull approach. The following geometric properties of $S_{\pi/2}$ (to be proven later) form the pillars of this approach and allow for a reduction of the problem into simpler tasks equal in number to the convex hull of S.

1. For each point $p \in S_{\pi/2}$, there exist two maximal rays emanating from p which intersect the same edge of the convex hull of S and such that the points of S lie outside the $\pi/2$-cone between the two rays.
2. For each point $p \in S_{\pi/2}$ there exist no more than *three* pairs of maximal rays which satisfy the previous property.
3. A pair of maximal rays which satisfies the first property includes the perpendicular from p to the corresponding convex hull edge in the $\pi/2$-cone between them.

The first task involves reporting unoriented maxima whose corresponding maximal rays intersect the same convex hull edge of S, and the other two properties facilitate the use of efficient computational geometry tools to develop an optimal running time algorithm. A detailed description of this approach is given in section 2.

1.2. Restricted unoriented maximum approach. This approach is based on the following simple property: for each point $p \in S_{\pi/2}$ there exist two maximal rays emanating from p which contain the $+x$-, the $-x$-, the $+y$-, or the $-y$-axis in the $\pi/2$-angle cone between them.

The problem is thus reduced to reporting for each of the four (directed) axes the unoriented maxima whose corresponding maximal rays contain it. For each axis, e.g., the $+y$-axis, we first sort points of the set in the direction perpendicular to the selected axis. We then perform two more linear passes. In the first pass, we scan the points of S from left to right constructing the convex hull of the visited points. Before $p \in S$ is processed, we compute the empty angle between the tangent from p to the convex hull and the selected axis, and call it θ. Perform a similar pass from right to left, storing the angle at p in ξ. A simple geometric argument shows that with respect to the selected axis a point $p \in S_{\pi/2}$ if and only if $\theta + \xi \geq \pi/2$.

It is natural to observe the similarity between the two approaches. However, the restricted unoriented maxima (RUM) approach is more suitable for handling the discrete versions of the problem, namely, answering unoriented maximum queries, and identifying unoriented maxima of a set in parallel models of computation. This follows from the fact that focusing on a particular direction allows for the use of the divide-and-conquer technique with an efficient merging process. Moreover, the RUM approach is more suitable for probabilistic analysis. A detailed description and the probabilistic analysis of this approach is given in section 4.

The rest of the paper is organized as follows. Section 2 is dedicated to the details of computing unoriented $\pi/2$-maxima for a given set of planar points. In section 3 a lower bound for the algebraic computation tree model is developed, which implies that our algorithm is optimal. Finally, in section 4 the expected number of unoriented $\pi/2$-maxima is analyzed (and used) to obtain a linear expected running time algorithm. In conclusion, we discuss an approach for handling arbitrary values of Θ and some related results and unsolved problems.

2. Computing unoriented $\pi/2$-maxima. Let $S = \{X_1, X_2, \ldots, X_n\}$ denote a set of n planar points in general position (no three points are collinear). Its convex hull $CH(S)$ is the pair $(V(S), E(S))$, where $V(S)$ is the set of vertices and $E(S)$ is the set of edges. We denote the size of the convex hull by h ($h = |V(S)| = |E(S)|$). A point $p \in S - V(S)$ is called a *candidate* for an edge $e \in E(S)$ if there exist two rays emanating from p with a $\pi/2$-angle cone between them which intersect the edge e. Clearly, a point which is an unoriented maximum must be a candidate for some edge of the convex hull, and all convex hull points are candidates. From now on, we pay attention to candidate points that are not on the convex hull. To report the elements of the set $S_{\pi/2}$, based on the convex hull approach, we first identify the *candidates* for each edge of $E(S)$; then we consider each subset separately and check whether a candidate is a true unoriented $\pi/2$-maximum (i.e., whether the $\pi/2$-cone defined by the candidate is empty or not). The following geometric properties of candidates are critical to the efficiency of our algorithm.

LEMMA 1. *Each point $p \in S - V(S)$ may be a candidate for at most three edges of $E(S)$.*

Proof. The circular angle around p is 2π and, moreover, the points are in general position. Therefore, if p is the candidate for more than three edges, then one of the cones must have angle less than $\pi/2$. ∎

LEMMA 2. *If point p is a candidate for the edge $e \in E(S)$, then p lies in the semicircle of diameter e which has a nonempty intersection with the interior of the polygon defined by $E(S)$.*

Proof. The proof is elementary and omitted. ∎

Therefore, we have h semicircles with the constraint that no point in $S - V(S)$ belongs to more than three semicircles. In the following subsection, we establish a linear bound on the number of intersections of such curves. Algorithms for identifying candidates for each edge and for reporting unoriented maxima are then presented in sections 2.2 and 2.3, respectively.

2.1. A combinatorial property of constrained circles. Let $C^{(h)} = \{C_1, \ldots, C_h\}$ be a set of h planar circles with the constraint that no point in the plane belongs to more than k circles ($k \leq h$). Let x_i and r_i denote, respectively, the center and the radius of the ith circle C_i.

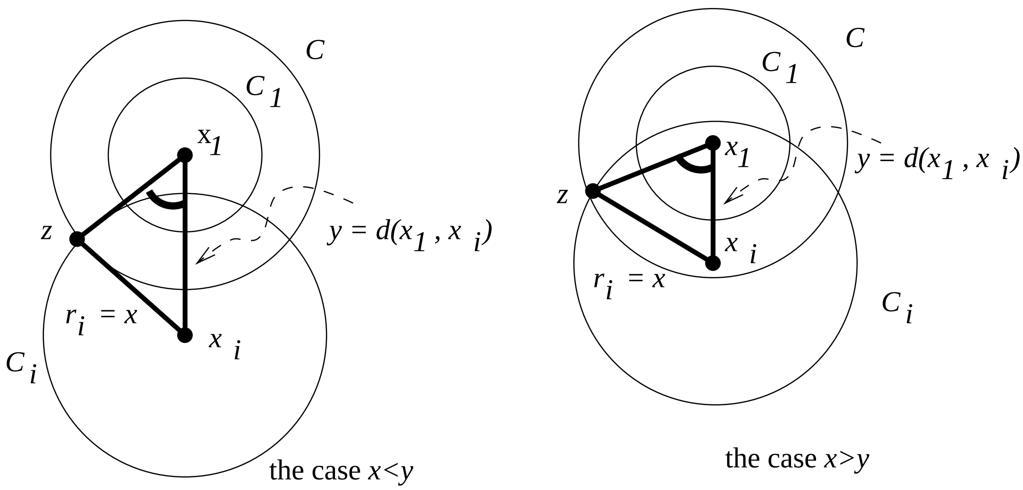

FIG. 2. *Finding a bound on θ.*

Without loss of generality, we assume that C_1 is the circle of $C^{(h)}$ with the smallest radius and that $r_1 = \min_{1 \le i \le h}\{r_i\} = 1$.

LEMMA 3. *At most $k-1$ circles can have their centers inside C_1.*

Proof. C_1 is the circle with smallest radius. Therefore, any circle C_i having x_i inside C_1 must contain x_1. Since x_1 cannot belong to more than k circles, the lemma follows. ∎

Let C be the circle concentric to C_1 with radius $\sqrt{3}$, let D_i be the disc consisting of circle C_i with its interior, and let the arc A_i be the intersection of D_i and the boundary of C if it exists. It is easy to see that any circle in $C^{(h)} - C_1$ that intersects C_1 and has a center outside C_1 must intersect C. The following lemma is based on Avis and Horton [AH81].

LEMMA 4. *If C_i intersects C_1 and x_i lies outside of C_1, then A_i subtends an angle of at least $\pi/3$ radians.*

Proof. Refer to Figure 2 for illustration. Let x, y and $\theta = \angle x_i x_1 z$ be as shown in Figure 2. Since C_i intersects C_1 and x_i lies outside of C_1, $y = d(x_1, x_i) \le r_1 + r_i = 1+x$. Therefore, we have $1 \le x, y \le 1+x$ and $\cos\theta = (3+y^2-x^2)/(2\sqrt{3}y)$; elementary geometry shows that $\cos\theta \le \sqrt{3}/2$. Therefore, $\theta \ge \pi/6$ radians, and thus the lemma follows. ∎

THEOREM 1. *At most $7k$ circles can intersect C_1.*

Proof. No point of the plane can belong to more than k circles. Therefore, Lemma 4 implies that no more than $6k$ circles can intersect C_1 and have their center outside C_1. Also, Lemma 3 implies that no more than k circles can intersect C_1 and have their center inside C_1. ∎

COROLLARY 1. *$C^{(h)}$ induces at most $14kh$ intersection points.*

Proof. Theorem 1 implies that C_1 can have at most $14k$ intersection points. By an inductive argument (removing C_1 from $C^{(h)}$ to obtain $C^{(h-1)}$), we can conclude that $C^{(h)}$ induces at most $14kh$ intersection points. ∎

It is clear that Corollary 1 holds for semicircles too.

COROLLARY 2. *In the arrangement of semicircles that was introduced in Lemma 2, there are at most $42h$ intersection points.*

Proof. The proof follows from Corollary 1 and Lemma 1. ∎

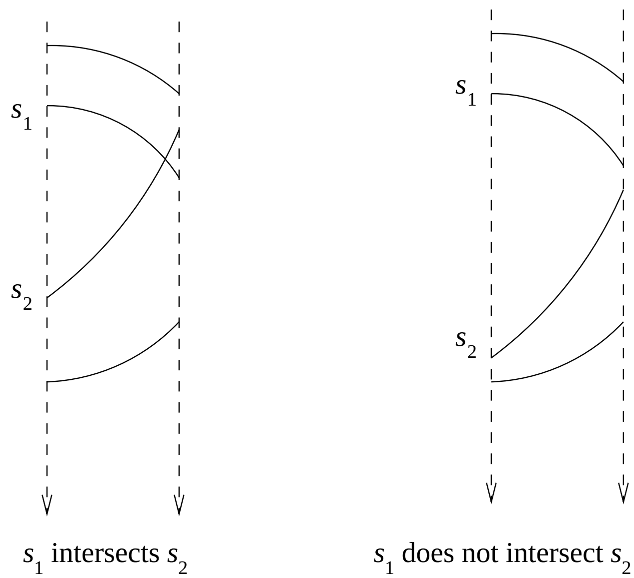

FIG. 3. *An example illustrating the idea of plane sweep.*

In this subsection, we showed that there are only a linear number of intersections among all the semicircles. We show in the next subsection how to apply this result to report candidates with respect to each hull edge.

2.2. Reporting candidates for each hull edge. In this section we will describe a procedure for reporting the candidates for $E(S)$. The procedure is based on the plane sweep technique of Bentley and Ottmann [BO79]. The idea of plane sweep can be described with the following simple example. Assume that we have two segments s_1, s_2, and without loss of generality, assume that the x-coordinates of the left (right) endpoints of s_1, s_2 are the same. The problem is to decide whether s_1 intersects s_2. We can see that s_1 intersects s_2 if and only if the order from top to bottom of the y-coordinates of the right endpoints of s_1, s_2 differs from the top-to-bottom order of the y-coordinates of the left endpoints of s_1, s_2. In general, the plane sweep method maintains a total order of some geometric objects (e.g., $O(n)$ segments) at a given stage. To check certain properties of two valid objects (e.g., whether s_1 intersects s_2), it simply checks whether the top-to-bottom order of these two objects switches at a later stage. Usually a dynamic balanced binary search tree is sufficient for the plane sweep method (to maintain the total order) [BO79]. In Figure 3 we illustrate an example for plane sweep for some xy-monotone (i.e., monotone in both the x- and y-directions) circular segments.

First we give a description of the procedure and then explain the essential details and analyze its correctness and performance.

PROCEDURE CANDIDATES

INPUT: A set S of n planar points.

OUTPUT: The list of edges of $E(S)$ together with a list of candidate points for each edge.

METHOD:

1. Compute the convex hull of S and store the edges of $CH(S)$, $E(S)$ in a doubly linked list.
2. Compute the semicircles having as diameters the edges of $E(S)$.
3. Partition each semicircle into at most three parts such that every (circular) segment produced is xy-monotone. Let H be the set of segments obtained (note that $|H| \in O(n)$).
4. Apply the Bentley and Ottmann [BO79] plane sweep algorithm on $H \cup (S - V(S))$ to report the intersection points of the monotone segments in H. When a point $p \in S - V(S)$ is met by the sweep line, an $O(\log n)$ search in a balanced search data structure T may be used to identify those edges of $CH(S)$ for which p is a candidate. At the end of this step, all candidates of $S - V(S)$ are known.
5. Produce the list of candidates for each edge of $E(S)$ using the output of step 4.

End of Procedure

Correctness of Procedure Candidates in computing the intersection points of the elements in $H \cup (S - V(S))$ follows directly from correctness of the sweep line algorithm in [BO79]. Computing such intersections is essential to maintaining a total vertical ordering of the segments in a search structure T where the following four operations can be implemented in $O(\log n)$ time.

1. INSERT(s, T) inserts the segment s into the total order maintained by T.
2. DELETE(s, T) deletes segment s from T.
3. ABOVE(s, T) returns the name of the segment immediately above s in T.
4. BELOW(s, T) returns the name of the segment immediately below s in T.

These operations are listed in [SH76] and referred to by [BO79]. They can be implemented using a balanced binary search tree.

For a given vertical sweep line L, T contains the total ordering of the monotone segments (of semicircles) intersecting L. They define vertical intervals on L, each of which corresponds to a unique intersection region. We modify the balanced search tree by keeping for each vertical interval (uniquely determined by two adjacent elements of H) the list of semicircles containing that segment. By Lemma 1, at most three such semicircles may exist. Therefore, the space complexity of the data structure is still linear. When a new semicircle is encountered (and two monotone segments are to be inserted), we use the information in its neighbor vertical intervals to establish its linked list. A deletion of a semicircle can be handled similarly. Finally, when an intersection point of two segments of semicircles is encountered, the appropriate linked list can be updated in constant time. Handling point $p \in S - V(S)$ requires performing a search of the structure T which returns the vertical interval that contains p. We can then determine the semicircles that contain p in constant time, and update the list of candidates for each of the corresponding convex hull edges in $E(S)$.

Step 1 can be done in $O(n \log n)$ time, and steps 2, 3, and 5 can be accomplished in $O(n)$ time. The Bentley and Ottmann [BO79] algorithm has an $O(n \log n + k \log n)$ running time, where k is the number of intersection points to be reported. Since we have $O(n)$ intersection points by Corollary 2, the execution time of step 4 is $O(n \log n)$. Therefore, Procedure Candidates reports the set of candidates for the convex hull edges in $O(n \log n)$ time and $O(n)$ space.

2.3. Computing unoriented maxima among candidates. Given the output of Procedure Candidates (i.e., a set of candidates for each edge of $E(S)$), we now

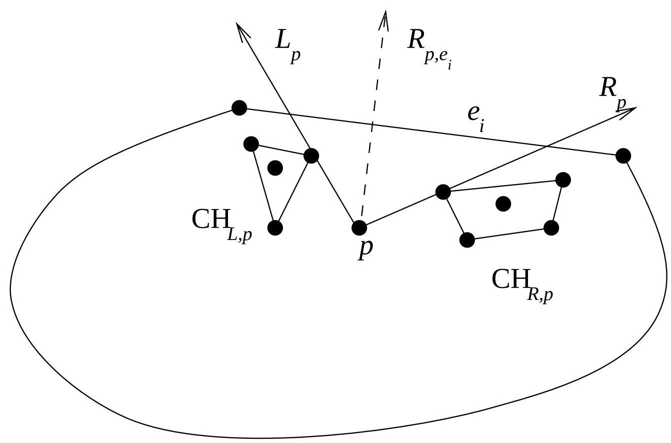

FIG. 4. *Computing unoriented maxima from candidates.*

develop a procedure to identify for each convex hull edge the unoriented maxima among its list of candidates.

Let $\mathcal{C}_i$ denote the set of candidates for the ith edge e_i of $E(S)$, and let k_i denote its size. If $p \in \mathcal{C}_i$, then the ray emanating from p and perpendicular to the edge e_i, denoted by R_{p,e_i}, properly intersects e_i since p lies inside the semicircle of diameter e_i. Let $wedge(p, e_i)$ be the largest angle at p which does not contain points of S and is bounded by two maximal rays, L_p and R_p, that emanate from p and intersect e_i (Figure 4).

LEMMA 5. *If point p is an unoriented maximum with respect to edge e_i, then R_{p,e_i} must belong to the cone defined by p and the maximal rays L_p and R_p.*

Proof. If R_{p,e_i} does not lie between the maximal rays L_p and R_p, then $wedge(p, e_i) < \pi/2$, a contradiction. □

LEMMA 6. *If the convex hulls of points in $\mathcal{C}_i - \{p\}$ to the left and to the right of R_{p,e_i}, denoted by $CH_{L,p}$ and $CH_{R,p}$ respectively, are known, then we can compute $wedge(p, e_i)$ in $O(\log n)$ time.*

Proof. Refer to Figure 4. Our problem is to compute the rightmost and leftmost (maximal) rays, R_p and L_p, emanating from p, intersecting e_i, and containing R_{p,e_i} in the cone (p, L_p, R_p). R_p (L_p) can be computed by finding the ray from p tangent to the convex hull to the right (left) of R_{p,e_i}, which can be done in $O(\log n)$ time [PS85]. If the angle between R_p and L_p (defined by the cone containing R_{p,e_i}) is $\geq \pi/2$, then p is an unoriented maximum. □

PROCEDURE UNORIENTED MAXIMA

INPUT: A list of candidates for the ith edge of *E(S)*.

OUTPUT: The unoriented maximal points and the rays defining their widest angles.

METHOD:

1. Sort the k_i points of $\mathcal{C}_i$ along e_i. Note that the sorted points define a simple polygonal chain.
2. Compute L_p for all points $p \in \mathcal{C}_i$ as follows:
   - $CH_L \leftarrow$ endpoint of e_i
   - Going from left to right using the order of the points of step 1:
     - Compute L_p using CH_L (as explained in Lemma 6).

– Insert p in CH_L using the rules of the convex hull algorithm of Avis, ElGindy, and Seidel [AES85].

3. Compute R_p for all points $p \in \mathcal{C}_i$ in a similar fashion to step 2, by scanning them from right to left.
4. For each $p \in \mathcal{C}_i$, compute angle $wedge(p, e_i)$ between L_p and R_p, and if $\alpha_p \geq \pi/2$, output (p, L_p, R_p).
5. Output $V(S)$.

End of Procedure

Correctness of the above procedure follows from the correctness of the on-line convex hull algorithm in [AES85] and from Lemma 6.

Step 1 is performed in $O(k_i \log k_i)$ time. Since the algorithm in [AES85] updates the convex hull of k_i points by insertion in $O(\log k_i)$ time, and since searching for L_p and R_p requires $O(\log k_i)$ time at most as explained in Lemma 6, then steps 2 and 3 require $O(k_i \log k_i)$ time. Step 4 is clearly performed in $O(k_i)$ time, hence $O(k_i \log k_i)$ total time is spent for edge e_i. Lemma 1 implies that $\sum_{i=1}^{h} k_i \log k_i \leq \log n \sum_{i=1}^{h} k_i \leq 3n \log n \in O(n \log n)$. Therefore, we can state the final result of this section as follows.

THEOREM 2. *All unoriented maximal points of S can be computed in $O(n \log n)$ time and $O(n)$ space.*

In the next section, we establish an $\Omega(n \log n)$ lower bound for computing unoriented maxima in the plane, thus proving that our algorithm is optimal.

3. Lower bound for the algebraic computation tree model. In this section we establish an $\Omega(n \log n)$ lower bound for computing unoriented Θ-maxima in the plane. This $\Omega(n \log n)$ lower bound for computing the unoriented maxima $S_\Theta \subseteq S$ in the plane, for $\pi/2 \leq \Theta \leq \pi$, is achieved by a reduction from the integer element uniqueness problem. Note that when $\Theta \geq \pi$, the unoriented maxima $S_\Theta \subseteq S$ are exactly the convex hull (extreme) points, and it is well known that computing the extreme points of a set of n points has a lower bound of $\Omega(n \log n)$ under the algebraic computation tree model [PS85]. Our result is as follows.

THEOREM 3. *The problem of computing $S_\Theta \subseteq S$ for $\pi/2 \leq \Theta \leq \pi$ is $\Omega(n \log n)$ under the algebraic computation tree model, where $|S| = n$.*

Proof. We use a reduction from integer element uniqueness. In Yao [Ya89] this problem is shown to have a lower bound of $\Omega(n \log n)$ under the algebraic computation tree model.

We are given a set of integers $M = \{x_1, \ldots, x_n\}$, input to the integer element uniqueness problem. For each x_i, produce the following six points: $(i + \epsilon, (nx_i)^2)$, $(i+\epsilon, (nx_i)^2+\epsilon)$, $(i+\epsilon, (nx_i)^2-\epsilon)$, $(i-\epsilon, (nx_i)^2)$, $(i-\epsilon, (nx_i)^2+\epsilon)$, and $(i-\epsilon, (nx_i)^2-\epsilon)$. The value of $\epsilon = 1/4$ is used for our proof. Let S be the set containing all of these points.

If $x_i = x_j$ then at least two out of the twelve induced points cannot be unoriented maxima (Figure 5). On the other hand, if x_i is unique in M, then the six points created for x_i are all unoriented maxima. Hence all x_i's in M are distinct if and only if there are exactly $6n$ unoriented maxima in S. We have thus reduced integer element uniqueness to computing the unoriented maxima in linear time. Since the integer element uniqueness problem has a lower bound of $\Omega(n \log n)$ under the algebraic computation tree model, the theorem follows. □

We have thus obtained an optimal algorithm for computing unoriented Θ-maxima in the plane. In the next section we present the RUM algorithm which will beat the $\Omega(n \log n)$ lower bound when the points are drawn from a common distribution. This is obtained via a careful probabilistic analysis of the expected number of unoriented

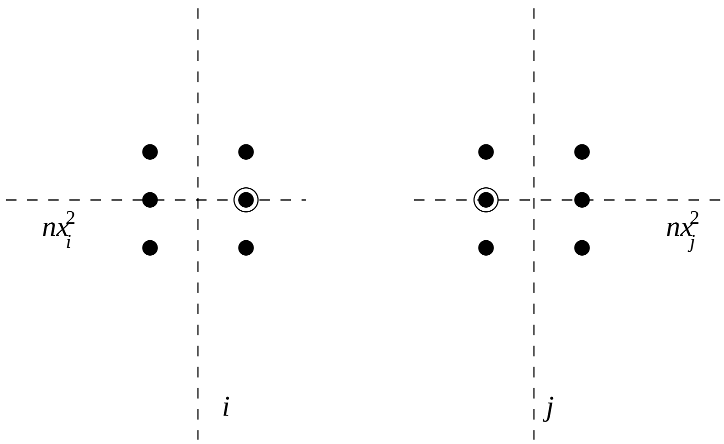

FIG. 5. *Lower bound proof: the case when $x_i = x_j$. The marked points cannot be unoriented maxima.*

maxima, together with a simple divide-and-conquer algorithm.

4. Expected number of unoriented maxima. In this section, we analyze the expected number of unoriented maxima when elements of the set S are independently drawn from a common distribution. Since n points on the perimeter of a convex set are all unoriented maxima, it is only natural to exclude such pathological cases. This is done by assuming that the distribution of the prototype data point is absolutely continuous; i.e., it has a density f. This has the added benefit that with probability one, no two points have the same coordinates. We also assume that f has compact support. Without loss of generality, we can then assume that f vanishes off $[0,1]^2$. We will show that under a mild condition on f, which is satisfied for most distributions that appear in probabilistic models, the expected number of unoriented maxima is $O(\sqrt{n})$. In section 4.4, we describe a divide-and-conquer algorithm that runs in linear expected time for this class of distributions.

The notion of unoriented maximum generalizes that of a maximal vector, for which algorithms can be found in [BS78, BKST78, De80, De85, GBT84, BCL90, Go94, KS85, KLP75]. The expected time was considered in all but the last two of these papers. For additional analysis, see [Dw90, Bu89]. All linear expected time algorithms described in these papers have conditions on the distribution that are more restrictive than the ones used in this paper.

4.1. Preliminaries. We define a cone $C_\theta(x, \eta)$ for $x \in \mathbb{R}^2$, $\theta \in [0, 2\pi)$, and $\eta \in [0, 2\pi)$ as the collection of all points $y \in \mathbb{R}^2$ with polar coordinate representation $y = x + re^{i\phi}$ for some $r > 0$ and $\phi \in (\theta - \eta/2, \theta + \eta/2)$. Thus, x is the top of the cone, and θ is the direction of the bisector, while η is the opening angle. Given a set of vectors $\mathcal{X}_n = \{X_1, \ldots, X_n\}$ in $\mathbb{R}^2$, we say that X_j is an UNORIENTED MAXIMUM if there exists a θ such that $C_\theta(X_j, \pi/2) \cap \mathcal{X}_n = \emptyset$. Thus, every maximal vector and every point on the convex hull of $\mathcal{X}_n$ is an unoriented maximum of $\mathcal{X}_n$.

It is helpful to cut the problem into manageable subproblems. To do so, we introduce the notion of a restricted unoriented maximum or RUM. Fix a direction $\zeta \in [0, 2\pi)$. Call X_j a RUM of $\mathcal{X}_n$ if there exists a direction θ such that

$$C_\theta(X_j, \pi/2) \cap \mathcal{X}_n = \emptyset$$

and

$$C_\theta(X_j, \pi/2) \supseteq C_\zeta(X_j, \pi/3) \ .$$

Call this collection of directional unoriented maxima S_ζ. Obviously, if S is the collection of all unoriented maxima, we have

$$S = \cup_{\zeta\in[0,2\pi)} S_\zeta = \cup_{j=0}^{11} S_{j\pi/6} \ .$$

This property allows us to focus on RUMs. In what follows, we fix $\zeta = \pi/2$ and abbreviate the restricted unoriented maxima with respect to this ζ to RUMs. The set of all RUMs among $X_1, \ldots, X_n$ is denoted by $\mathcal{S}_n$. We list three structural properties of $\mathcal{S}_n$.

1. The Lipschitz property. If $X_i \in \mathcal{S}_n$, $X_j \in \mathcal{S}_n$, then the line segment joining X_i and X_j has an angle with the x-axis within $\pi/3$ of 0 or π. Suppose that the segment forms an angle of ξ degrees, with $\pi/2 \geq \xi > \pi/3$. Then either

   $$X_j \in C_\theta(X_i, \pi/2) \supseteq C_\zeta(X_i, \pi/3)$$

   for some θ, or vice versa,

   $$X_i \in C_\theta(X_j, \pi/2) \supseteq C_\zeta(X_j, \pi/3) \ .$$

   In the former case, X_i is not a RUM, and in the latter case, X_j is not a RUM. If we sort all the RUMs from left to right and join them by straight line segments, we obtain a piecewise linear curve that is Lipschitz of constant not more than $\pi/3$. (A function f is Lipschitz of constant C if $|f(x) - f(y)| \leq C|x - y|$.)
2. The monotonicity property.

   $$\text{RUM}(X_1, \ldots, X_{n+1}) \subseteq \text{RUM}(X_1, \ldots, X_n) \cup \{X_{n+1}\} \ .$$

3. The transitive property.

   $$\text{RUM}(X_1, \ldots, X_{n+m}) = \text{RUM}(\text{RUM}(X_1, \ldots, X_n), \text{RUM}(X_{n+1}, \ldots, X_{n+m})) \ .$$

We will need the following elementary lemma.

LEMMA 7. *If N is a binomial (n,p) random variable, then* $\mathbf{P}\{N > enp\} \leq e^{-np}$.

Proof. By Chernoff's bounding method [Ch52], for $t > 0$ and $\lambda > 0$,

$$\begin{aligned}
\mathbf{P}\{N > t\} &\leq \mathbf{E}\left\{e^{\lambda N - t}\right\} \\
&\leq \left(e^\lambda p + 1 - p\right)^n e^{-\lambda t} \\
&\leq \exp\left((e^\lambda - 1)np - \lambda t\right) \\
&= \exp\left(t - np - t\log\left(\frac{t}{np}\right)\right) \quad (\text{take} \ \ \lambda = \log(t/(np)) \ )
\end{aligned}$$

so that

$$\mathbf{P}\left\{N > enp\right\} \leq e^{-np} \ . \qquad \square$$

Theorem 4 deals with distributions having a bounded density: for such distributions, there is limited dependence between the components of the random vector X. In a later section, we will obtain analogous results for unbounded densities. In the bounds presented in this paper, the dependence upon f is measured through $\|f\|_\infty$ or $\int f^\alpha$.

THEOREM 4. *Let X be a random vector on $[0,1]^2$ whose density is bounded by $\|f\|_\infty$. For an i.i.d. sample $X_1, \ldots, X_n$ drawn from X, let $\mathcal{S}_n$ be the collection of* RUM*s. Then*

$$\lim_{n\to\infty} \mathbf{P}\{|\mathcal{S}_n| > C\sqrt{n}\} = 0,$$

where $C = e\sqrt{2(1+2/\sqrt{3})\|f\|_\infty \log 4}$. Also,

$$\limsup_{n\to\infty} \frac{\mathbf{E}\{|\mathcal{S}_n|\}}{C\sqrt{n}} \le 1 \ .$$

Proof. As described in the caption of Figure 6, the unit square is covered by a circumscribed rhombus with angles 120, 60, 120, and 60 degrees. From top to bottom, it measures $2a = 1+\sqrt{3}$, and from left to right $2b = 1 + 1/\sqrt{3}$. The area of the rhombus is $1+2\sqrt{3}$. Partition the rhombus into $m \times m$ equal rhombi as shown in the figure. This is achieved by taking m slabs A_i and m slabs B_j, and defining rhombi by the intersections $A_i \cap B_j$. There are m^2 small rhombi that can be addressed by index pairs (i,j), $1 \le i,j \le m$. A chain of cells is an ordered collection of such pairs, beginning with $(1,1)$ and ending with (m,m), satisfying the successor rule: (i,j) must be followed by either $(i,j+1)$ or $(i+1,j)$. See the lightly shaded collection in Figure 6. Thus, the chain contains precisely $2m-1$ cells, and by a simple counting argument, it is easy to see that there are exactly

$$\binom{2m-2}{m-1}$$

possible chains. Let us mark each cell that contains at least one RUM (dark in Figure 6). We claim that the marked cells are contained in a chain. This, of course, follows from the Lipschitz curve property we established above and our choice of angles when defining the partition. We let $N(\mathcal{C})$ denote the number of data points in the chain $\mathcal{C}$. Thus,

$$|\mathcal{S}_n| \le \max_{\text{all chains } \mathcal{C}} N(\mathcal{C}).$$

By the inclusion–exclusion inequality, we have

$$\begin{aligned}
\mathbf{P}\left\{|\mathcal{S}_n| > t\right\} &\le \mathbf{P}\left\{\max_{\text{all chains } \mathcal{C}} N(\mathcal{C}) > t\right\} \\
&\le \sum_{\text{all chains } \mathcal{C}} \mathbf{P}\left\{N(\mathcal{C}) > t\right\} \\
&\le \binom{2m-2}{m-1} \sup_{\text{all chains } \mathcal{C}} \mathbf{P}\left\{N(\mathcal{C}) > t\right\} \ .
\end{aligned}$$

Next, observe that the probability of a cell is given by

$$\int_{A_i \cap B_j} f(x,y)\,dx\,dy \le \|f\|_\infty \int_{A_i \cap B_j \cap [0,1]^2} dx\,dy \le \frac{\|f\|_\infty (1+2/\sqrt{3})}{m^2} \ .$$

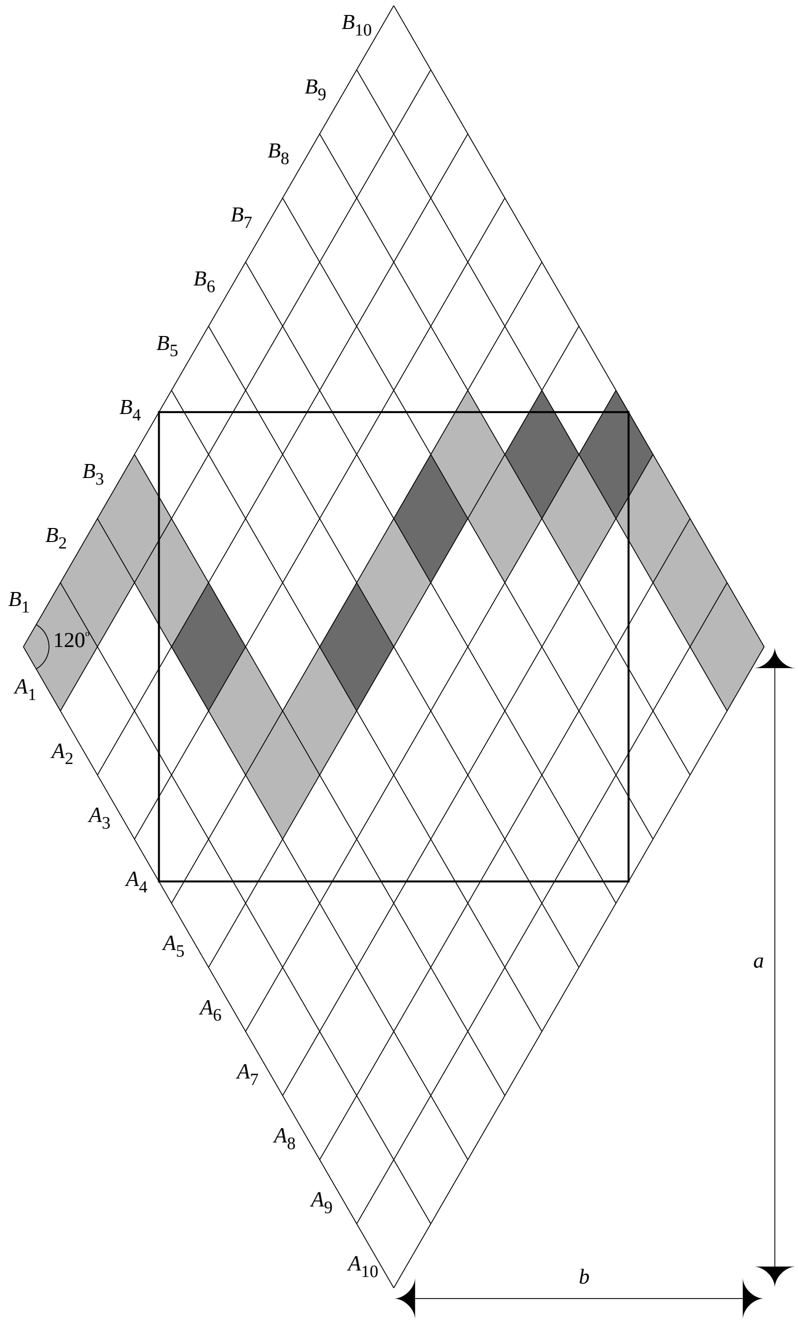

FIG. 6. *The unit square* $[0,1]^2$ *is shown in dark lines. Consider the rhombus of angle* 120 *degrees that circumscribes the square. Partition the rhombus into a grid of* $m \times m$ *similar small rhombi. A chain (in lightest shading) is any collection of small rhombi where the first and last rhombus are the leftmost and rightmost rhombi, respectively, and intermediate rhombi share one side. The rhombi in a chain must have increasing* x*-coordinate values of their centers. The dark shaded rhombi are those that contain at least one restricted unoriented maximum. Observe that these cells always belong to a chain.*

Thus, for a chain of $2m-1$ cells $\mathcal{C}$, $N(\mathcal{C})$ is binomial with parameters n and

$$\int_{\mathcal{C}} f \le \frac{\|f\|_\infty (1+2/\sqrt{3})(2m-1)}{m^2} \le \frac{2\|f\|_\infty(1+2/\sqrt{3})}{m} \stackrel{\text{def}}{=} q\ .$$

Hence, $\mathbf{P}\{N(\mathcal{C}) > t\} \le \mathbf{P}\{\text{Binomial}(n,q) > t\}$. Therefore, by Lemma 7, if $m \to \infty$ as $n \to \infty$,

$$\begin{aligned}
\mathbf{P}\left\{|\mathcal{S}_n| > enq\right\} &\le \binom{2m-2}{m-1} \sup_{\text{all chains } \mathcal{C}} \mathbf{P}\left\{N(\mathcal{C}) > enq\right\} \\
&\sim \frac{2^{2m-2}}{\sqrt{\pi m}} \sup_{\text{all chains } \mathcal{C}} \mathbf{P}\left\{N(\mathcal{C}) > t\right\} \\
&\le \frac{2^{2m-2}}{\sqrt{\pi m}} \mathbf{P}\left\{\text{Binomial}(n,q) > enq\right\} \\
&\le \frac{2^{2m-2} e^{-nq}}{\sqrt{\pi m}}\ .
\end{aligned}$$

Define $q' = mq$. First, take

$$m = \left\lceil \sqrt{\frac{nq'}{\log 4}} \right\rceil\ .$$

Then

$$\mathbf{P}\left\{|\mathcal{S}_n| > enq\right\} \le (1+o(1)) \frac{\exp(m\log 4 - nq'/m)}{4\sqrt{\pi m}} \le \frac{4+o(1)}{\sqrt{\pi m}} \to 0\ .$$

This proves the first part of Theorem 4. For the second part, choose $\epsilon > 0$ very small and set

$$m = \left\lfloor (1-\epsilon)\sqrt{\frac{nq'}{\log 4}} \right\rfloor\ .$$

We verify quickly that

$$\mathbf{P}\left\{|\mathcal{S}_n| > enq\right\} \le e^{-c\sqrt{n}}$$

for some constant $c > 0$ depending upon ϵ. Then,

$$\mathbf{E}|\mathcal{S}_n| \le enq + n\mathbf{P}\{|\mathcal{S}_n| > enq\} = enq + o(1) \sim e\sqrt{nq'\log 4}/(1-\epsilon)\ .$$

Theorem 4 now follows since ϵ was arbitrary. □

4.2. Lower bounds. The number of unoriented maxima is larger than the number of maximal vectors, i.e., the number of data points X_i for which one of $C_{\pi/4}(X_i, \pi/2)$, $C_{3\pi/4}(X_i, \pi/2)$, $C_{5\pi/4}(X_i, \pi/2)$, and $C_{7\pi/4}(X_i, \pi/2)$ has a nonempty intersection with $X_1, \ldots, X_n$. We denote the set of maximal vectors for $X_1, \ldots, X_n$ by $\mathcal{M}_n$. Thus, $|\mathcal{S}_n| \ge |\mathcal{M}_n|$. This can be used to show that the bound of Theorem 4 cannot be improved for many simple distributions. To clarify this, just take the uniform distribution on the trapezoid T formed by intersecting $[0,1]^2$ with $\{(x,y) : y <$

$x < y + c\}$, with $0 < c \le 1$. The area of the trapezoid is $1/2(1-(1-c)^2) = c - c^2/2$. Hence,

$$f(x,y) = \frac{1}{c - c^2/2} I_T(x,y) \ .$$

Thus, $\|f\|_\infty = 1/(c - c^2/2)$. We take an integer m large enough such that $1/m < c$. Then partition the unit square into a rectangular grid of m by m with sides equal to $1/m$. Mark the m grid cells that straddle the diagonal of the square. Let $E_1, \ldots, E_m$ be the indicators of the events that the marked grid cells contain at least one data point, with E_1 referring to the cell with the largest y-values, and so on down. A simple geometric argument shows that

$$|\mathcal{M}_n| \ge \sum_{i=1}^{m} E_i \ .$$

Hence, if the marked grid cells intersected with our trapezoid T yield the triangles $S_1, \ldots, S_m$,

$$\begin{aligned}
\mathbf{E}|\mathcal{M}_n| &\ge m\mathbf{E}E_1 \\
&= m\,(1 - (1 - \|f\|_\infty \text{ area}(S_1))^n) \\
&= m\left(1 - (1 - \|f\|_\infty/(2m^2))^n\right) \\
&\ge m\left(1 - \exp(-\|f\|_\infty n/(2m^2))\right) \\
&\ge m/2 \\
&\ge \sqrt{\|f\|_\infty n/4\log 4} - 1
\end{aligned}$$

if we choose $m = \lfloor \sqrt{\|f\|_\infty n/\log 4} \rfloor$. Recall that n has to be large enough to insure that $1/m < c$. Thus, we have

$$\mathbf{E}|\mathcal{M}_n| \ge \sqrt{\|f\|_\infty n/4\log 4} - 1 \ .$$

The upper bound in Theorem 4 cannot be improved upon in terms of $\|f\|_\infty$ and n unless the class of distributions is further restricted.

4.3. Random vectors with very dependent coordinates. If f is unbounded, Theorem 4 becomes useless. It is possible, however, that $\int f^\alpha < \infty$ for some $\alpha > 1$. This fact can be used to obtain a different collection of upper bounds.

THEOREM 5. *Let X be a random vector on $[0,1]^2$ whose density satisfies $\int f^\alpha < \infty$ for some $\alpha > 1$. For an i.i.d. sample $X_1, \ldots, X_n$ drawn from X, let $\mathcal{S}_n$ be the collection of* RUMs. *Then*

$$\lim_{n\to\infty} \mathbf{P}\{|\mathcal{S}_n| > Cn^{\alpha/(2\alpha-1)}\} = 0,$$

where

$$C = e\left(\left(\int f^\alpha\right)^{1/\alpha} \left(2(1+2/\sqrt{3})\right)^{1-1/\alpha}\right)^{\alpha/(2\alpha-1)} (\log 4)^{(2\alpha-1)/(\alpha-1)} \ .$$

Also,

$$\limsup_{n\to\infty} \frac{\mathbf{E}\{|\mathcal{S}_n|\}}{Cn^{\alpha/(2\alpha-1)}} \le 1 \ .$$

Proof. We follow the proof of Theorem 4. Note that $N(\mathcal{C})$ is binomial with parameters n and p, with p given by

$$\begin{aligned}\int_{\mathcal{C}} f(x,y)\,dx\,dy &\le \left(\int f^\alpha\right)^{1/\alpha}\left(\int_{\mathcal{C}\cap[0,1]^2} dx\,dy\right)^{1-1/\alpha}\\ &\le \left(\int f^\alpha\right)^{1/\alpha}\left(\frac{2(1+2/\sqrt{3})}{m}\right)^{1-1/\alpha}\\ &\stackrel{\text{def}}{=} q \stackrel{\text{def}}{=} q'/m^{1-1/\alpha}\ .\end{aligned}$$

Here we used Hölder's inequality and an inequality from the proof of Theorem 4. Therefore, $N(\mathcal{C})$ is binomial with parameters n and p where $p \le q$, and $\mathbf{P}\{N(\mathcal{C}) > t\} \le \mathbf{P}\{\text{ Binomial}(n,q) > t\}$. As in the proof of Theorem 4, when $m \to \infty$ as $n \to \infty$,

$$\mathbf{P}\left\{|\mathcal{S}_n| > enq\right\} \le \frac{(1+o(1))2^{2m-2}e^{-nq}}{\sqrt{\pi m}}\ .$$

With

$$m = \left\lceil \left(\frac{nq'}{\log 4}\right)^{\alpha/(2\alpha-1)}\right\rceil\ ,$$

we obtain

$$\begin{aligned}\mathbf{P}\left\{|\mathcal{S}_n| > enq\right\} &\le (1+o(1))\frac{\exp(m\log 4 - nq'/m^{1-1/\alpha})}{4\sqrt{\pi m}}\\ &\le \frac{4+o(1)}{\sqrt{\pi m}}\\ &\to 0\ .\end{aligned}$$

This proves the first part of Theorem 5. The second part follows from the first part by using arguments analogous to those of Theorem 4. □

Remark 1. We note that the condition $\int f^\alpha < \infty$ imposes a condition on the peakedness of the density f. For bounded densities, we clearly have $\int f^\alpha < \infty$. Theorem 4 is obtained as a limit of Theorem 5 when we let $\alpha \to \infty$. □

Remark 2. If ψ is a positive convex strictly increasing function, then for the chain $\mathcal{C}$ in the proof, we have by Jensen's inequality,

$$\int_{\mathcal{C}} f \le \int_{\mathcal{C}\cap[0,1]^2} dx\,dy \times \psi^{\text{inv}}\left(A/\int_{\mathcal{C}\cap[0,1]^2} dx\,dy\right)\ ,$$

where $A = \int \psi(f)$. Using this instead of Hölder's inequality, with $\psi(u) = u\log^a(1+u)$ for $a > 0$, we see that

$$\mathbf{E}|\mathcal{S}_n| = O\left(\frac{n}{\log^a n}\right)$$

whenever $\int f\log^a(1+f) < \infty$. Observe also that this condition is satisfied whenever $\int f^b < \infty$ for some $b > 1$. □

Remark 3. Theorems 4 and 5 remain valid with different constants for cones $C_\theta(x,\eta)$, $\eta \in (0,\pi]$. □

4.4. Divide-and-conquer algorithms for unoriented maxima. At least five strategically different algorithms can be used for finding the outer layer $\mathcal{M}_n$ of $X_1, \ldots, X_n$ in the plane. Let $L_n = |\mathcal{M}_n|$ denote the number of points on the outer layer.

1. **The naive algorithm.** For each X_i, determine in linear time whether a point is a maximal vector. The time taken by this algorithm is $\Theta(n^2)$, while the space is $\Theta(n)$.
2. **One sort and one elimination pass.** Sort the data points according to their y-coordinates, and eliminate unwanted points in a second stage by passing through the sorted array and keeping partial extrema in the x-direction. This may be implemented in $O(n \log n)$ worst-case time.
3. **Divide-and-conquer** [BS78]. Start with n singleton outer layers, marry (merge) all outer layers pairwise, and repeat these pairwise marriages until one outer layer is left. Noting that outer layers of sizes k and m can be married in $O(k+m)$ time, and that about $\log_2 n$ rounds of merging are needed, it is easy to see that the time taken by this algorithm is $O(n \log n)$. However, since many points are thrown away at early stages, there is reasonable hope of obtaining linear expected time $\mathbf{E}T$. The following is known: $\mathbf{E}T = O(n)$ when the components of X_1 are independent [BS78, De83]. In the general case, $\mathbf{E}T = O(n)$ if and only if $\sum_n \mathbf{E}L_n/n^2 < \infty$ by a general theorem on the expected time analysis of divide-and-conquer algorithms [De83]. An important class of problems is that in which f is bounded, in which case we see that $\mathbf{E}L_n = O(\sqrt{n})$ and thus $\mathbf{E}T = O(n)$ [De85].
4. **Bucketing methods.** Partition $[0,1]$ into a grid of size about $\sqrt{n} \times \sqrt{n}$, assign all points to grid locations, and mark in all columns (rows) the topmost (leftmost) and bottommost (rightmost) occupied grid cells, together with their inner neighbors. Finally, use the naive algorithm (1) to obtain the outer layer among the points in all the marked cells [De86]. This too yields linear expected time for bounded densities, but it uses a different computational model because truncation is assumed to be available at unit time cost. [Ma84] use another grid in which in each cell, the outer layer is found, and the overall outer layer is found in a second step.
5. **Output-sensitive algorithms based on lazy sorting.** In [KS85], algorithms are presented that take worst-case time bounded by $O(n \log L_n)$. The expected time therefore is bounded by a constant times $\mathbf{E} n \log L_n \leq n \log \mathbf{E} L_n$.

In this section, using the results of the previous sections, we offer a linear expected time divide-and-conquer algorithm for finding the set $\mathcal{S}_n$ of all RUMs that runs under conditions weaker than any condition mentioned above for linear expected time for outer layers. A similarly adapted divide-and-conquer algorithm for outer layers would yield linear expected time under the same general conditions.

PROCEDURE RESTRICTED UNORIENTED MAXIMA

INPUT: A set of n planar points $X_1, \ldots, X_n$.

OUTPUT: The set $\mathcal{S}_n$ of all RUMs of $X_1, \ldots, X_n$.

METHOD:

1. Put all data points X_i in singleton sets S_i.
2. Put all sets S_i in a queue Q.

3. WHILE $|Q| > 1$ DO
   - Dequeue sets S and T from Q.
   - Compute $V = \text{RUM}(S \cup T)$.
   - Enqueue Q with V.

End of Procedure

THEOREM 6. *If the divide-and-conquer algorithm is used on data that are i.i.d. and have a density of compact support such that* $\int f \log^a(1+f) < \infty$ *for some* $a > 1$, *and if the merging of two sets of* RUMs *is supported in linear time, then the overall expected time is* $O(n)$. *The running time is* $O(n \log n)$ *in the worst case.*

Proof. The expected time analysis of general divide-and-conquer algorithms given in [De83] shows that linear expected time is obtained if the data constitute an i.i.d. sequence, the MERGE step takes linear time in the size $|S| + |T|$, and

$$\sum_{n=1}^{\infty} \frac{\mathbf{E}|\mathcal{S}_n|}{n^2} < \infty .$$

By Remark 2,

$$\mathbf{E}|\mathcal{S}_n| = O\left(\frac{n}{\log^a n}\right)$$

when f has compact support and $\int f \log^a(1+f) < \infty$. Theorem 6 follows when $a > 1$. ∎

Remark 4. The condition mentioned in the proof above was rediscovered later in the context of randomized incremental algorithms by Clarkson and Shor [CS88, CS89]. For a slightly different approach with conditions deduced from recursions, see [BS78].

Remark 5. One can push things further and get linear expected time if f has compact support and if for some $a > 1$, $\int f \log(1+f) \log^a \log(1+f) < \infty$. ∎

Remark 6*: On merging sets of* RUMS. Performing the MERGE step in linear time requires keeping track of the sets of RUMs according to increasing x-coordinates. First, we merge the sorted sets S and T into a set W, sorted by x-coordinate. We then perform two more linear passes. In the first pass, we construct the convex hull in clockwise fashion from left to right as we visit points of W (in fact, this only gives the upper part of the convex hull; the lower part is not needed). This is done by Graham's incremental algorithm [Gr72]. As X_i is processed, we note the angle between the convex hull edge leading to X_i, and the y-axis, and call it θ_i. Repeat a similar pass in counterclockwise manner from right to left, storing the angles in ξ_i. A simple geometric argument shows that $X_i \in \text{RUM}(W)$ if and only if $\theta_i + \xi_i \le \pi/2$. The entire procedure takes linear time. ∎

Remark 7*: Lazy merging of rums.* If we find $\text{RUM}(W)$ in time $O(|W| \log |W|)$, results from [De83] guarantee overall linear expected time if

$$\sum_{n=1}^{\infty} \frac{\mathbf{E}|\mathcal{S}_n| \log |\mathcal{S}_n|}{n^2} < \infty .$$

Since $\log |\mathcal{S}_n| \le \log n$, it suffices to verify that

$$\sum_{n=1}^{\infty} \frac{\mathbf{E}|\mathcal{S}_n| \log n}{n^2} < \infty .$$

By Theorem 5, this is satisfied if for some $\alpha > 1$, $\int f^\alpha < \infty$. By Remark 2, it also suffices that $\int f \log^a(1+f) < \infty$ for some $a > 2$. ∎

5. Concluding remarks. We introduced unoriented Θ-maximal points and described an optimal $O(n \log n)$ algorithm for identifying them when $\Theta \geq \pi/2$. The case $\Theta < \pi/2$ is handled in the Appendix. We also showed that if the points are random and have a common density (satisfying mild regularity conditions), then we can compute the unoriented $\pi/2$-maxima in $O(n)$ expected time.

6. Appendix. For values of $\Theta < \pi/2$, the geometric properties of Lemmas 2 and 5 become useless. However, we are able to modify them slightly as shown below to obtain efficient algorithms for this case.

LEMMA 8. *If point p is a candidate for the edge $e \in E(S)$, then p lies in the part of the circle which has e as a chord, and p makes an angle Θ with e which has a nonempty intersection with the interior of $CH(S)$.*

LEMMA 9. *If point p is an unoriented Θ-maximum with respect to edge e_i, then the angle between the rays L_p and R_p must contain either R_{p,e_i} or one of $\pi/\Theta - 2$ directions which are separated from R_{p,e_i} by integer multiples of Θ.*

A point $p \in S - V(S)$ may be a candidate for at most $2\pi/\Theta$ edges of $CH(S)$. Therefore, Corollary 1 implies that the circles defined in Lemma 8 cannot have more than $14(2\pi/\Theta)h$ ($\in O(n/\Theta)$) intersections, which changes the running time of Procedure Candidates to $O((n/\Theta) \log n)$. In addition, the procedure of section 2.3 for computing unoriented Θ-maxima among candidates needs to be executed $(\pi/\Theta) - 1$ times for each convex hull edge. As a result, we can compute the set S_Θ, for $\Theta < \pi/2$, in $O((n/\Theta) \log n)$ running time. The algorithm is clearly optimal for fixed values of Θ. However, no matching lower bound is known when Θ is part of the input.

REFERENCES

[AES85] D. AVIS, H. ELGINDY, AND R. SEIDEL, *Simple on-line algorithms for convex polygons*, in Computational Geometry, G.T. Toussaint, ed., North-Holland, Amsterdam, 1985, pp. 23–42.

[AH81] D. AVIS AND J. HORTON, *Remarks on the sphere of influence graph*, Ann. New York Acad. Sci., 1981, pp. 323–327.

[BCL90] J. L. BENTLEY, K. L. CLARKSON, AND D. B. LEVINE, *Fast linear expected-time algorithms for computing maxima and convex hulls*, in Proc. 1st Annual ACM-SIAM Symposium on Discrete Algorithms, SIAM, Philadelphia, 1990, pp. 179–187.

[BKST78] J. L. BENTLEY, H. T. KUNG, M. SCHKOLNICK, AND C. D. THOMPSON, *On the average number of maxima in a set of vectors*, J. Assoc. Comput. Mach., 25 (1978), pp. 536–543.

[BO79] J. L. BENTLEY AND T. A. OTTMANN, *Algorithms for reporting and counting geometric intersections*, IEEE Trans. Comput., C-28 (1979), pp. 643–647.

[BS78] J. L. BENTLEY AND M. I. SHAMOS, *Divide and conquer for linear expected time*, Inform. Process. Lett., 7 (1978), pp. 87–91.

[Bu89] C. BUCHTA, *On the average number of maxima in a set of vectors*, Inform. Process. Lett., 33 (1989), pp. 63–65.

[Ch52] H. CHERNOFF, *A measure of asymptotic efficiency of tests of a hypothesis based on the sum of observations*, Ann. Math. Stat., 23 (1952), pp. 493–507.

[Ch85] B. CHAZELLE, *On the convex layers of a convex set*, IEEE Trans. Inform. Theory, IT-31 (1985), pp. 509–517.

[CS88] K. L. CLARKSON AND P. W. SHOR, *Algorithms for diametrical pairs and convex hulls that are optimal, randomized, and incremental*, in Proc. 4th Symposium on Computational Geometry, ACM, New York, 1988, pp. 12–17.

[CS89] K. L. CLARKSON AND P. W. SHOR, *Applications of random sampling in computational geometry* II, Disc. Comput. Geom., 4 (1989), pp. 387–422.

[CSY87] R. COLE, M. SHARIR, AND C. K. YAP, *On K-hulls and related problems*, SIAM J. Comput., 16 (1987), pp. 61–77.

[De80] L. DEVROYE, *A note on finding convex hulls via maximal vectors*, Inform. Process. Lett., 11 (1980), pp. 53–56.

[De83] L. Devroye, *Moment inequalities for random variables in computational geometry*, Computing, 30 (1983), pp. 111–119.

[De85] L. Devroye, *On the expected time required to construct the outer layer*, Inform. Process. Lett., 20 (1985), pp. 255–257.

[De86] L. Devroye, *Lecture Notes on Bucket Algorithms*, Birkhäuser-Verlag, Boston, 1986.

[Dw90] R. A. Dwyer, *Kinder, gentler average-case analysis for convex hulls and maximal vectors*, SIGACT News, 21 (1990), pp. 64–71.

[EKS83] H. Edelsbrunner, D. G. Kirkpatrick, and R. Seidel, *On the shape of a set of points in the plane*, IEEE Trans. Inform. Theory, IT-29 (1983), pp. 551–559.

[GBT84] H. N. Gabow, J. L. Bentley, and R. E. Tarjan, *Scaling and related techniques for geometry problems*, in Proc. 16th Annual ACM Symposium on the Theory of Computing, 1984, pp. 135–143.

[Go94] M. J. Golin, *A provably fast linear-expected-time maxima-finding algorithm finite planar set*, Algorithmica, 11 (1994), pp. 501–524.

[Gr72] R. Graham, *An efficient algorithm for determining the convex hull of a finite planar set*, Inform. Process. Lett., 1 (1972), pp. 132–133.

[KLP75] H. T. Kung, F. Luccio, and F. P. Preparata, *On finding the maxima of a set of vectors*, J. Assoc. Comput. Mach., 22 (1975), pp. 469–476.

[KS85] D. G. Kirkpatrick and R. Seidel, *Output-sensitive algorithms for finding maximal vectors*, in Proc. 2nd Annual Symposium on Computational Geometry, ACM, New York, 1985, pp. 89–96.

[Ma84] M. Machii and Y. Igarashi, *A Hashing Method of Finding the Maxima of a Set of Vectors*, Technical Report CS-84-2, Department of Computer Science, Gunma University, Gunma, Japan, 1984.

[PS85] F. P. Preparata and M. I. Shamos, *Computational Geometry: An Introduction*, Springer-Verlag, New York, 1985.

[SH76] M. I. Shamos and D. Hoey, *Geometric intersection problems*, in Proc. 17th Annual IEEE Symposium on Foundations of Computer Science, 1976, pp. 208–215.

[Ya89] A. C. C. Yao, *Lower bounds for algebraic computation trees with integer inputs*, in Proc. 30th Annual IEEE Symposium on Foundations of Computer Science, 1989, pp. 308–313.