

# REPORTING BICHROMATIC SEGMENT INTERSECTIONS FROM POINT SETS\*

CARMEN CORTÉS<sup>†</sup>, DELIA GARIJO<sup>‡</sup>, MARÍA ÁNGELES GARRIDO<sup>§</sup>,  
CLARA I. GRIMA<sup>¶</sup>, ALBERTO MÁRQUEZ<sup>||</sup>,  
AUXILIADORA MORENO-GONZÁLEZ<sup>\*\*</sup> and JESÚS VALENZUELA<sup>††</sup>

*Departamento de Matemática Aplicada I, Universidad de Sevilla  
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain*

<sup>†</sup>*ccortes@us.es*

<sup>‡</sup>*dgarijo@us.es*

<sup>§</sup>*vizquete@us.es*

<sup>¶</sup>*grima@us.es*

<sup>||</sup>*almar@us.es*

<sup>\*\*</sup>*auxiliadora@us.es*

<sup>††</sup>*jesusv@us.es*

MARÍA TRINIDAD VILLAR

*Departamento de Geometría y Topología, Universidad de Sevilla  
Avda. Reina Mercedes s/n, 41012 Sevilla, Spain  
villar@us.es*

Received 15 April 2011

Revised 19 June 2012

Communicated by David Mount

## ABSTRACT

In this paper, we introduce a natural variation of the problem of computing all bichromatic intersections between two sets of segments. Given two sets  $R$  and  $B$  of  $n$  points in the plane defining two sets of segments, say red and blue, we present an  $O(n^2)$  time and space algorithm for solving the problem of reporting the set of segments of each color intersected by segments of the other color. We also prove that this problem is 3-Sum hard and provide some illustrative examples of several point configurations.

*Keywords:* Red-blue points; bichromatic segment intersection; discrimination and separation.

## 1. Introduction

Geometric intersection problems have been widely studied motivated by a number of applications, such as geometric packing and covering, solid modeling and collision detection. Mount<sup>15</sup> lists several of these problems and also different techniques

\*Partially supported by project MTM2008-05866-C03-01.

used to design intersection algorithms. Among them, highlights the *segment intersection problem*: report the intersections of  $n$  line segments in the plane. Bentley and Ottmann<sup>5</sup> provided an algorithm for reporting all  $k$  intersecting pairs of  $n$  line segments in  $O((k+n)\log n)$  time and  $O(n)$  space. Chazelle and Edelsbrunner<sup>7</sup> presented an  $O(k+n\log n)$  time algorithm which is optimal with respect to the running time but uses  $O(k+n)$  space. This was improved by Balaban<sup>3</sup> achieving the same running time but using only  $O(n)$  space.

Many variations of the segment intersection problem have been investigated but the most widely studied has been the *bichromatic segment intersection problem*: given two sets of segments, say red and blue, whose total size is  $n$ , report all the intersections between red segments and blue segments. These red-blue intersections are called *bichromatic intersections*. The case where there are no monochromatic intersections, i.e., intersections between segments having the same color, can be viewed as a special case of the general segment intersection problem. Thus, many authors worked on the variation where monochromatic intersections exist which is considered to be a more difficult problem. Agarwal<sup>2</sup> and Chazelle<sup>6</sup> showed that the  $k$ -bichromatic intersections can be reported in  $O(k+n^{4/3}\log^{O(1)}n)$  time using a partitioning technique called *cuttings* (see also Ref. 15). See Refs. 4, 9 and 13 for more information about this topic.

In this paper, we present a new variation of the bichromatic segment intersection problem in the case where monochromatic intersections exist. Instead of reporting all bichromatic intersections between two sets of colored segments, we are given two sets of colored points defining two sets of colored segments, and study the problem of reporting the set of segments of each color intersected by segments of the other color. More concretely,

**Problem.** Let  $R$  and  $B$  be two distinct sets of red and blue points in the plane. A line segment defined by two red points is a red segment, and that defined by two blue points is a blue segment. Let  $S_b$  be the set of blue segments that intersect at least one red segment, and let  $S_r$  be the set of red segments crossed by at least one blue segment. Report  $S_b$  and  $S_r$ .

Our main motivation to introduce this problem is to detect which segments are *conflict-free*, in the sense that no bichromatic intersection is generated by using them. In fact, to avoid intersections is one of the main difficulties in many geometric problems associated to red and blue segments. Sometimes it is not so important how many bichromatic intersections a given segment can have, but if in a configuration one single (forbidden) intersection is reported, then the whole configuration is considered to be not valid.

From another point of view, our problem can also be seen as a discrimination/separation problem of red and blue points, where the discrimination criterion is the existence of a witness (say a red segment given by two red points) which discretizes/separates some pairs of blue points, or in other words we want to obtain disjoint classes of blue points not interfered or being affected by pairs of red points.

Finally, it is worth to mention that our problem is also related to the problem of intersection searching. See Cheng and Janardan,<sup>8</sup> where space-efficient algorithms are presented for some geometric intersection searching problems such as ray-shooting, segment intersection searching, and triangle stabbing.

Let  $n_r$  and  $n_b$  be the sizes of the point sets  $R$  and  $B$ , respectively. Let  $S = R \cup B$  and  $n = n_r + n_b$ . By simplicity of arguments, we assume that the points are in general position, i.e., no three of them are collinear. Let  $s_b = |S_b|$  denote the size of  $S_b$ . Analogously,  $s_r = |S_r|$  is the size of  $S_r$ . Let  $\text{Sep}(S) = (S_b, S_r)$ . Figure 1 shows two point configurations with the values  $s_b$  and  $s_r$ . Note that  $s_b = 0$  if and only if  $s_r = 0$ . We shall assume that  $n \geq 3$  since otherwise the problem is trivial.<sup>a</sup>

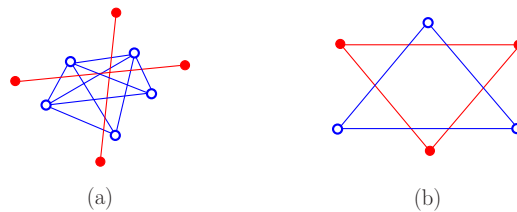


Fig. 1. (a)  $s_b = 9$  and  $s_r = 2$ , (b)  $s_b = 3$  and  $s_r = 3$ .

*Outline of the paper.* In Section 2, we present an  $O(n^2)$  time and space algorithm for computing  $\text{Sep}(S)$ . Section 3 is devoted to show the 3-Sum hardness of our problem. We also analyze the difference between computing  $\text{Sep}(S)$  and detecting the existence of at least one bichromatic intersection. Finally, since all the possible values of  $s_b$  and  $s_r$  are not achieved, in Section 4, we provide some results for illustrating their valid ranges.

## 2. Computing $\text{Sep}(S)$

In this section we present an efficient  $O(n^2)$  time algorithm for computing  $\text{Sep}(S) = (S_b, S_r)$ , that is, for reporting the set of segments of each color intersected by segments of the other color. This problem is equivalent to the problem of reporting the complementary set, i.e., the set of segments of each color that are not intersected by segments of the other color, which is reported by giving the subsets of endpoints of these segments. Additionally, we also can compute the pair  $(s_b, s_r)$ . We distinguish cases in order to report the set  $S_b$  according to the location of the endpoints of the blue segments regarding the convex hull of  $R$ , denoted by  $CH(R)$ . Similarly for  $S_r$ .

Let  $G = (V(G), E(G))$  be the geometric graph such that  $V(G) = B$  and  $E(G)$  is the set of blue segments intersecting no red segment, i.e.,  $E(G)$  is the complement

<sup>a</sup>In all the figures in this paper, red points are illustrated as solid red points, and blue points are depicted as hollow blue points.

of  $S_b$  with respect to the set of all possible edges in  $G$ . Let  $B_e$  and  $B_i$  denote, respectively, the subsets of blue points that are exterior and interior to  $CH(R)$ . Note that either set may be empty. Thus,  $G$  is the union of two disjoint graphs (both not connected in general): the *exterior graph*, denoted by  $G_e = (V(G_e), E(G_e))$ , with  $V(G_e) = B_e$ ; and the *interior graph*, denoted by  $G_i = (V(G_i), E(G_i))$  with  $V(G_i) = B_i$ .

**Lemma 1.**  $G_i$  is the union of disjoint complete graphs.

**Proof.** Let  $H$  be the complete geometric graph defined by the red points. Since the edges of  $G_i$  are not intersected by red segments, then every connected component of  $G_i$  is contained in one of the faces of the line-segment arrangement  $H'$  defined by the edges of  $H$ . Hence  $G_i$  is the union of disjoint graphs, and its components are complete graphs since the faces of  $H'$  are convex regions.  $\square$

Let  $P$  be a convex polygon, let  $Q$  be a set of exterior points to  $P$ , and  $p, q \in Q$ . Clearly, the segment  $\overline{pq}$  intersects  $P$  if and only if  $\overline{pq}$  intersects a side of  $P$ . Thus,  $\overline{pq}$  intersects  $P$  if and only if there exist two vertices  $u, v$  of  $P$  such that  $\overline{pq}$  and  $\overline{uv}$  intersect.

Consider now a caliper of  $P$  which is defined by two parallel supporting lines of  $P$  on antipodal vertices of  $P$ . The direction of the caliper is the direction of its parallel lines. See Ref. 17 for standard techniques using rotating calipers over a convex polygon.

**Lemma 2.** Let  $Q$  be a set of exterior points to a convex polygon  $P$ , and  $p, q \in Q$ . The segment  $\overline{pq}$  intersects  $P$  if and only if there exists a caliper of  $P$  through two vertices  $u, v$  of  $P$  that contains the points  $p$  and  $q$ , and  $\overline{uv}$  and  $\overline{pq}$  intersect.

**Proof.** It suffices to consider the caliper of  $P$  with direction given by the direction of the line through  $p$  and  $q$  (see Fig. 2(a)).  $\square$

**Lemma 3.** Let  $Q$  be a set of  $n$  exterior points to an  $n$ -sided convex polygon  $P$ . To decide whether any of the segments with endpoints in  $Q$  intersects  $P$  can be done in  $O(n \log n)$  time and  $O(n)$  space. Moreover, knowing the rotational ordering of the points of  $Q$  with respect to  $P$ , to decide whether any of the segments with endpoints in  $Q$  intersects  $P$  takes  $\Theta(n)$  time and space.

**Proof.** First, we obtain the *rotational ordering* of the points of  $Q$  with respect to  $P$  as follows: For each point  $p \in Q$ , compute the two directed tangents (or supporting lines) from  $p$  to  $P$ . These lines are called either right or left directed tangents depending on which half-plane  $P$  lies in. Then, sort the left directed tangents of all the points of  $Q$  by increasing slope starting from the horizontal slope. Analogously, sort the right directed tangents of all the points of  $Q$ . This process can be done in  $O(n \log n)$  time and  $O(n)$  space. Finally, in extra  $O(n)$  time, merge the two sorted

lists of directed tangents (left and right) into an ordered list which is called the rotational ordering of the points of  $Q$  with respect to  $P$ .

Now, we consider a horizontal caliper of  $P$ , and compute the points of  $Q$  inside it. By using the rotational ordering, we know which is the next point of  $Q$  entering or going out of the caliper as we rotate it a complete round over  $P$  (see Fig. 2(b)). Thus, the updates (insertions or deletions) of points inside the caliper can be done in constant time per point.

According to Lemma 2, it is clear that given a segment  $\overline{pq}$  with  $p, q \in Q$ , we can decide whether  $\overline{pq}$  intersects the  $n$ -sided convex polygon  $P$  in  $O(n \log n)$  time and  $O(n)$  space by rotating the caliper a complete round around  $P$ . It is also clear that if we know the rotational ordering above, then the decision can be done in  $\Theta(n)$  time and space.  $\square$

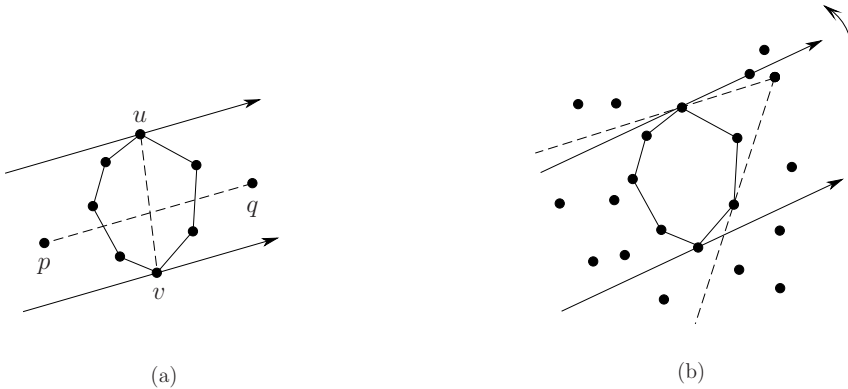


Fig. 2. (a) A caliper of  $P$  through two vertices  $u, v$  of  $P$  with direction given by the direction of the line through  $p$  and  $q$ , (b) a rotating caliper.

Lemma 1 says that  $G_i$  is the union of disjoint complete graphs and therefore,

$$S_b = E(\overline{G_e}) \cup E(\overline{G_i}) \cup \{\overline{uv} : u \in V(G_i), v \in V(G_e)\},$$

where  $E(\overline{G_e})$  (resp.  $E(\overline{G_i})$ ) denotes the set of edges of the complement graph of  $G_e$  (resp.  $G_i$ ). We shall use Lemma 3 and an incremental procedure described below to compute the sets  $E(\overline{G_e})$  and  $E(\overline{G_i})$  in  $O(n^2)$  time.

### 2.1. Splitting $CH(R)$ into convex regions

A key tool for computing  $E(\overline{G_i})$  in  $O(n^2)$  time is the next procedure which provides a partition of  $CH(R)$  into convex regions, each one is either empty or contains only blue endpoints whose segments are not intersected by red segments. This partition of  $CH(R)$  is generated in an incremental way, and has to be stored in some data structure such that we can update a constant number of new convex regions generated at each stage.

Let  $V(G_i) = \{b_1, \dots, b_m\}$  be the set of blue interior points to  $CH(R)$ , in the order given in the input. Consider the equivalence relation:  $b_j \sim b_k$  if and only the blue segment  $\overline{b_j b_k}$  crosses no red segment. In this case, we say that  $b_j$  and  $b_k$  are related. As initial stage, take the blue point  $b_1$ , and let  $C_1(b_1) := CH(R)$ . At stage  $k$ ,  $k \leq m$ , we will introduce the blue point  $b_k$ . Now, consider the current convex regions at stage  $k-1$  with  $2 \leq k-1 \leq m$ . The convex regions can be of two types:

- (1) Convex regions that contain all the blue points (at least one)  $b_s$  of  $V(G_i)$  with  $1 \leq s \leq k-1$  so that all are related among them. Each such region is associated with one of those blue points, say  $b_j$ , as its representative, and it is denoted by type  $C_{k-1}(b_j)$ .
- (2) Convex regions that contain no blue point  $b_s$  of  $V(G_i)$  with  $1 \leq s \leq k-1$ , denoted by type  $C_{k-1}(\emptyset)$ .

At stage  $k$ , the next blue point  $b_k$  is located into a convex region of type  $C_{k-1}(b_j)$  or  $C_{k-1}(\emptyset)$ <sup>b</sup> which is split into at most four new convex regions as follows. We distinguish three cases:

*Case 1.*  $b_k \in C_{k-1}(\emptyset)$ : update  $C_{k-1}(\emptyset) := C_k(b_k)$ . Note that only the type of the region has changed (see Fig. 3, stages 4 and 5, where  $b_5 \in C_4^2(\emptyset)$  and so the transition from stage 4 to 5 only requires updating  $C_4^2(\emptyset) = C_5(b_5)$ ).

*Case 2.*  $b_k \in C_{k-1}(b_j)$  and  $b_k \sim b_j$ : update  $C_{k-1}(b_j) := C_k(b_j)$ . Observe that  $b_k \in C_k(b_j)$  (see Fig. 3, stages 2 and 3, where  $b_3 \in C_2(b_1)$  which is updated as  $C_3(b_1)$ ).

*Case 3.*  $b_k \in C_{k-1}(b_j)$  and the blue segment  $\overline{b_k b_j}$  is crossed by at least one red segment: consider a red segment  $s_{jk}$  intersecting  $\overline{b_j b_k}$  and the line  $\ell_{jk}$  containing the segment  $s_{jk}$ . Assume first that the segment  $s_{jk}$  is strictly contained in  $C_{k-1}(b_j)$ , that is, its two red endpoints are located in the interior of  $C_{k-1}(b_j)$ . Thus,  $\ell_{jk}$  intersects  $C_{k-1}(b_j)$  in two edges since the points are in general position. Join the endpoints of  $s_{jk}$  with the endpoints of those edges. Therefore  $C_{k-1}(b_j)$  is split into four convex regions (see Fig. 3, stage 2):

- (a)  $C_k(b_k)$  which might contain blue points  $b_s$  with  $s \in \{k+1, \dots, m\}$  but there are no blue points  $b_s$  with  $s \in \{1, \dots, k-1\}$ .
- (b)  $C_k(b_j)$  containing the blue points  $b_s$  with  $s \in \{1, \dots, k-1\}$  so that  $b_s \sim b_j$ .
- (c) Two regions of type  $C_k(\emptyset)$  not containing the blue points  $b_1, \dots, b_k$ .

<sup>b</sup>Formally, the convex regions of type  $C_{k-1}(\emptyset)$  should be denoted as  $C_{k-1}^1(\emptyset, \{t, h\})$  and  $C_{k-1}^2(\emptyset, \{t, h\})$  indicating the current stage  $k-1$ , the blue points  $b_t, b_h$  which generate those convex regions at some moment of the process, and also distinguishing the at most two regions that can be generated by  $b_t, b_h$ . Nevertheless, for simplicity of notation we shall use either  $C_{k-1}^1(\emptyset)$  and  $C_{k-1}^2(\emptyset)$  (only distinguishing the two possible convex regions generated at stage  $k-1$ ) or  $C_{k-1}(\emptyset)$  (only specifying the region type).

The remaining convex regions are updated obtaining more convex regions of type  $C_k(\emptyset) = C_{k-1}(\emptyset)$  not containing the blue points  $b_1, \dots, b_k$ , and convex regions of type  $C_{k-1}(b_s)$  are updated as  $C_k(b_s)$  with  $s \in \{1, \dots, k-1\}$  and  $b_s \approx b_j$  (see Fig. 3). Note that in this case the blue points  $b_s$ ,  $b_k$  and  $b_j$  are located in different convex regions, i.e., the blue segments given by these points are intersected by red segments.

Suppose now that the segment  $s_{jk}$  is contained in  $C_{k-1}(b_j)$  and one or both of its endpoints belong to  $C_{k-1}(b_j)$ . We then proceed as above splitting  $C_{k-1}(b_j)$  into two or three convex regions respectively, missing at most the two classes of type  $C_k(\emptyset)$  (see Fig. 3, stages 3 and 4, where  $C_3(b_2)$  is split into the convex regions  $C_4(b_2)$  and  $C_4(b_4)$ ).

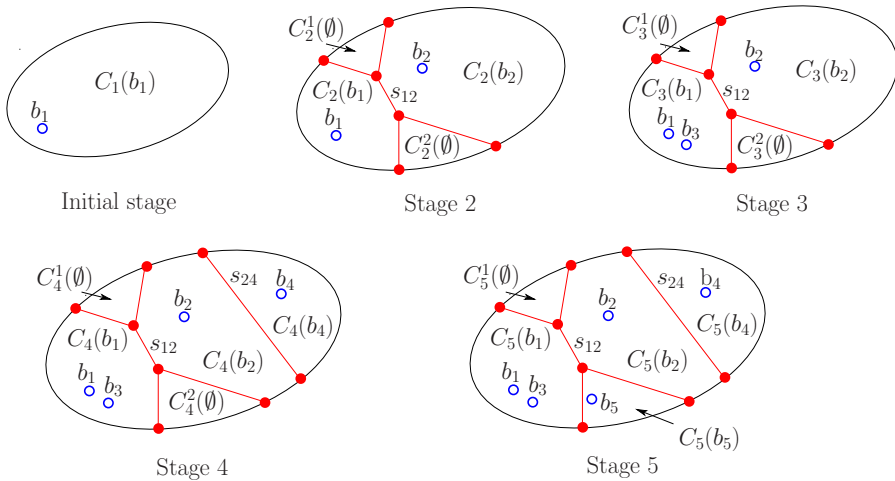


Fig. 3. An instance of convex regions created up to stage 5.

Finally, assume that the segment  $s_{jk}$  is not contained in  $C_{k-1}(b_j)$ , i.e., at least one endpoint of  $s_{jk}$  is located in a different convex region (see Fig. 4). Since  $C_{k-1}(b_j)$  is the convex region to be split we proceed analogously but considering, by using an artificial point (point  $q$  in Fig. 4), the part of the segment contained in the convex region to be split. Such point will not be taken into account in the next steps and will make no change in the time complexity of the next algorithm.

Note that the partition of  $CH(R)$  obtained by this incremental procedure (with insertions only) is a planar subdivision with linear complexity, where the cells are convex regions. The problem of the dynamic planar point location with insertions and deletions has been widely studied.<sup>16</sup> Nevertheless, in our problem we have to update the planar subdivision in an incremental way by splitting a convex region into at most four new convex regions inserting at most five new edges in each stage. Thus, we can use the linear-space data structure from Arge *et al.*<sup>1</sup> which in the

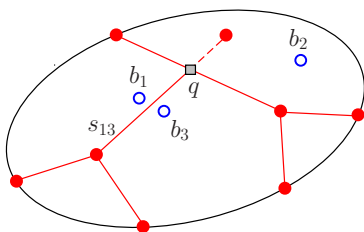


Fig. 4. The endpoints of  $s_{13}$  are located in different convex regions.

incremental case supports queries and edge insertions in  $O(\log n)$  time. Since in the construction of our planar subdivision the number of edge insertions and queries is  $O(n)$  we can establish the following result.

**Theorem 1.** *Computing the planar subdivision formed by the convex regions partitioning  $CH(R)$  can be done in  $O(n \log n)$  time and  $O(n)$  space.*

## 2.2. The $O(n^2)$ algorithm

As mentioned earlier, Lemma 3 and the procedure described in Subsection 2.1 are the key tools to design the following algorithm for reporting the sets of segments of each color intersected by segments of the other color.

PROCEDURE: RED-BLUE-INTERSECTED-SEGMENTS

**Input:**  $S = B \cup R$ ,  $|B| = |R| = n$

**Output:**  $\text{Sep}(S) = (S_b, S_r)$  and  $(s_b, s_r)$

- (1) Compute  $V(G_e)$  and  $V(G_i)$  in  $O(n \log n)$  time by determining the blue points which are *exterior* and *interior* to  $CH(R)$ , respectively.
- (2) “Computation of  $E(\overline{G_e})$ ”:
  - (a) In  $O(n \log n)$  time compute both the clockwise and counterclockwise rotational orders of the blue points of  $V(G_e)$  with respect to  $CH(R)$ .
  - (b) Compute the set  $E(\overline{G_e})$  of blue segments with endpoints in  $V(G_e)$  intersecting  $CH(R)$  by using a rotating caliper over  $CH(R)$ . The time complexity of this step is  $O(|E(\overline{G_e})| + n \log n)$ , i.e., depends on the cardinality of its output which is at most  $O(n^2)$ .
- (3) “Computation of  $E(\overline{G_i})$ ”: Let  $V(G_i) = \{b_1, \dots, b_m\}$  and  $C_1(b_1) := CH(R)$ .
  - (a) In  $O(mn) = O(n^2)$  time construct the dual arrangement  $\mathcal{A}$  of lines from the points of  $R \cup V(G_i)$ .
  - (b) **For**  $k = 2, \dots, m$  **do**
    - (i) In  $O(\log n)$  time compute the current convex region where  $b_k$  is contained.
    - (ii) **If**  $b_k$  is contained in  $C_{k-1}(\emptyset)$  **then** update  $C_{k-1}(\emptyset) := C_k(b_k)$  in  $O(1)$  time. **Else** “ $b_k$  is contained in  $C_{k-1}(b_j)$ ,  $j < k$ ” proceed as follows:

- (A) In  $O(n)$  time do the following: (1) use the arrangement  $\mathcal{A}$  to obtain the clockwise rotational order of the red points with respect to the point  $b_k$  ( $b_k$  is a line in the dual arrangement  $\mathcal{A}$ ; we traverse this line, in the order given by  $\mathcal{A}$ , discovering the order in which dual red lines cross it, thereby obtaining the clockwise rotational order of the red points with respect to  $b_k$ .), (2) analogously, use the arrangement  $\mathcal{A}$  to obtain the clockwise rotational order of the red points with respect to the point  $b_j$ , (3) merge both clockwise rotational orders to compute the clockwise rotational order of the red points with respect to the blue segment  $\overline{b_k b_j}$ .
  - (B) Using Lemma 3, in  $O(n)$  time determine whether there exists a red segment  $s_{jk}$  intersecting the blue segment  $\overline{b_j b_k}$ .
  - (C) **If** answer is the affirmative **then** split the convex region  $C_{k-1}(b_j)$  using the procedure described in Subsection 2.1. **Else** “ $b_k \sim b_j$ ” **then** update  $C_{k-1}(b_j) := C_k(b_j)$  in  $O(1)$  time.
- (iii) In  $O(1)$  time add  $b_k$  to the set of blue points in its definitive convex region.
- (c) Compute  $E(\overline{G_i})$  from the disjoint subsets of blue points contained in the non-empty convex regions.
- (4) In  $O(n^2)$  time compute the set  $S_b = E(\overline{G_e}) \cup E(\overline{G_i}) \cup \{\overline{uv} : u \in V(G_i), v \in V(G_e)\}$  and the value  $s_b$ .
- (5) Proceed analogously to compute the set  $S_r$  and the value  $s_r$ .

Note that Step 3(b) of the above described algorithm uses  $O(mn + m \log n + n \log n) = O(n^2)$  time. Observe in particular that the rotational orderings taken from the dual arrangement  $\mathcal{A}$  lets us apply Lemma 3, and Theorem 1 says that the total amortized time complexity of Step 3 is  $O(n \log n)$ . Therefore, the overall time complexity of Step 3 is  $O(n^2)$ . Thus, we have shown the following result.

**Theorem 2.**  $\text{Sep}(S) = (S_b, S_r)$  and  $(s_b, s_r)$  can be computed in  $O(n^2)$  time and space.

### 3. Hardness

In this section, we show that computing  $s_b$  and  $s_r$  is a 3-Sum hard problem which implies the 3-Sum hardness of the problem of reporting  $\text{Sep}(S)$ . The class of 3-Sum hard problems was introduced by Gajentaan and Overmars<sup>10</sup> where they list a number of problems for which they prove that they are at least as hard as the so-called *base problem*: *Given a set  $S$  of  $n$  integers, does there exist three elements of  $S$  that sum up to zero?* The best known algorithm for the base problem takes  $\Theta(n^2)$  time.

For the reduction, we shall use the following 3-Sum hard problem<sup>10</sup>: *Given a set of  $n$  points with integer coordinates on three horizontal lines  $y = 0$ ,  $y = 1$  and*

$y = 2$ , determine whether there exists a non-horizontal line containing three of the points.

For simplicity of notation, it can be assumed that the number of points on each line is the same and so we shall consider a set of  $3n$  points. Moreover, the points on each line shall be identified with its  $x$ -coordinate.

**Theorem 3.** *Computing  $s_b$  and  $s_r$  is 3-Sum hard.*

**Proof.** We are given a set of  $3n$  points with integer coordinates on three horizontal lines  $y = 0$ ,  $y = 1$  and  $y = 2$ , denoted  $A$ ,  $C$  and  $D$ , respectively. First, we transform each point  $c \in C$  into two points  $c - \frac{1}{4}$ ,  $c + \frac{1}{4}$ , called the *transformed points* of  $c$ . Now, all the points on  $A$  and  $D$  are colored red while all the  $2n$  points on  $C$  are colored blue. So far, a set of  $4n$  red and blue points has been obtained.

We next add a set of  $n - 1$  red points, say  $\Omega = \{(r_i, 1 + \varepsilon) \mid i = 1, \dots, n - 1\}$ . Figure 5 shows that there is exactly one of these points located between the transformed points of  $c_i, c_{i+1} \in C$  for  $i = 1, \dots, n - 1$ , that is,  $c_i + \frac{1}{4} < r_i < c_{i+1} - \frac{1}{4}$ , more concretely we take  $r_i = (c_i + c_{i+1})/2$ . Moreover, they are located at distance  $\varepsilon > 0$  to the line  $C$ . Thus, a set of  $5n - 1$  red and blue points has been obtained.

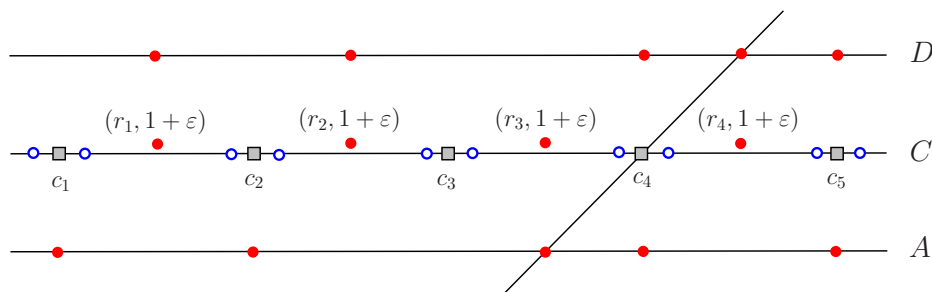


Fig. 5. The set of  $5n - 1$  red and blue points and a non-horizontal line through points on  $A$ ,  $C$  and  $D$  (color online).

Now, we shall guarantee that there exists a value of  $\varepsilon$  such that no blue segment with endpoints  $c - \frac{1}{4}$ ,  $c + \frac{1}{4}$  is intersected by a red segment with at least one endpoint in  $\Omega$ . As it will be shown below, this fact is the key for reducing our problem to the 3-Sum hard problem specified above.

Let us call the blue segment with endpoints  $c - \frac{1}{4}$ ,  $c + \frac{1}{4}$  the *segment of  $c$* . Obviously, the red segments with either both endpoints in  $\Omega$  or one endpoint in  $D$  and the other in  $\Omega$  intersect no segment of a point  $c \in C$ . So the conditions over  $\varepsilon$ , to guarantee that no segment of  $c \in C$  is intersected, will be imposed by the red segments with one endpoint in  $A$  and the other in  $\Omega$ .

Let  $a_1$ ,  $a_n$  and  $c_1$ ,  $c_n$  be the first and last points on  $A$  and  $C$ , respectively, in the left-right order. Let  $p = c_1 + 1$  and  $q = c_n - 1$  (these two points have integer

coordinates). Figure 6 shows an easy construction to obtain the two small triangles  $T_1, T_2$ . It can be easily checked that the distance to the line  $C$  of any point on the intersection region of the overlapping of  $T_1$  and  $T_2$  gives an adequate value for  $\varepsilon$ .

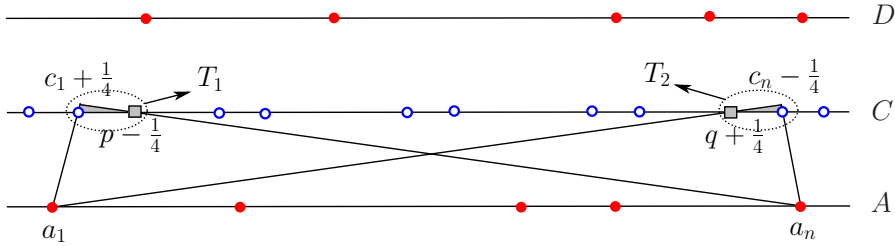


Fig. 6. Construction for obtaining an adequate value for  $\varepsilon$ .

We now prove that there exists a line containing three points  $a \in A$ ,  $c \in C$  and  $d \in D$  if and only if  $s_b > 2n(n-1)$ .

Note first that the  $\frac{2n(2n-1)}{2} - n$  blue segments with endpoints  $c_i \pm \frac{1}{4}$ ,  $c_j \pm \frac{1}{4}$  for  $c_i \neq c_j$  are intersected by a red segment (e.g.,  $\overline{a_1 r_{i+1}}$ ). Hence  $s_b \geq \frac{2n(2n-1)}{2} - n = 2n(n-1)$ . Therefore the result holds whenever at least one segment of a point  $c \in C$  is intersected. Clearly, if there is a line going through  $a \in A$ ,  $c \in C$  and  $d \in D$ , then the blue segment with endpoints  $c - \frac{1}{4}$ ,  $c + \frac{1}{4}$  is crossed by the red segment given by  $a, d$ , and so  $s_b > 2n(n-1)$  (see Fig. 5). It remains to prove the reverse.

Suppose that  $s_b > 2n(n-1)$ . This implies that all the blue segments are intersected but at most  $n-1$  of them. Hence, there is at least one segment with endpoints  $c - \frac{1}{4}$ ,  $c + \frac{1}{4}$  which is crossed by a red segment with endpoints, say  $a, d$ . Assume that the bichromatic intersection occurs at the point  $c + \delta$  for  $\delta \in (-\frac{1}{4}, \frac{1}{4})$  and so  $a + d = 2(c + \delta)$ . Since  $a, c$  and  $d$  are integer numbers then  $\delta = 0$  and  $a + d = 2c$ . This proves that there is a line going through the points  $a \in A$ ,  $c \in C$  and  $d \in D$ .  $\square$

**Corollary 1.** *Computing  $\text{Sep}(S)$  is 3-Sum hard.*

We have provided an algorithm for computing  $\text{Sep}(S)$  in  $O(n^2)$  time and space, and proved that the problem is 3-Sum hard. This prompts the question of which is the difference between computing the exact values of  $s_b$  and  $s_r$  and detecting the existence of at least one bichromatic intersection.

**Proposition 1.** *In  $O(n \log n)$  time it can be determined whether  $\text{Sep}(S)$  is empty, that is,  $s_b = s_r = 0$ .*

**Proof.** Deciding whether  $R$  and  $B$  are linearly separable can be done in  $O(n)$  time.<sup>14</sup> If so, then  $s_b = s_r = 0$ . If  $R$  and  $B$  are not linearly separable, then their

respective boundaries of the convex hulls  $CH(R)$  and  $CH(B)$  either intersect or form nested polygons. We proceed as follows: In  $O(n \log n)$  time, compute the boundaries of  $CH(R)$  and  $CH(B)$ , and check whether they intersect. If they do then  $s_b \geq 1$  and  $s_r \geq 1$ . Otherwise, they are nested polygons. Assume, for instance, that  $CH(R) \subset CH(B)$ . To decide which blue points are inside  $CH(R)$  can be done in  $O(n \log n)$  time. If there are blue points interior and exterior to  $CH(R)$  then  $s_b \geq 1$  and  $s_r \geq 1$ . Otherwise, all the blue points are exterior to  $CH(R)$  and, by Lemma 3, we can decide in  $O(n \log n)$  time whether any of the segments they define intersects  $CH(R)$ . Analogously if  $CH(B) \subset CH(R)$ .  $\square$

**Remark 1.** We conjecture (open problem) that the decision problem of determining whether there is at least one bichromatic intersection has a lower bound of  $\Omega(n \log n)$ . Given the  $O(n \log n)$  upper bound on the emptiness problem of Proposition 3.3 and the widely held conjecture of a lower bound of  $\Omega(n^2)$  for any 3-Sum-hard problem, it follows that there is a considerable gap in the computational complexities of the emptiness and counting problems for  $\text{Sep}(S)$ .

#### 4. The Range of Values of $s_b$ and $s_r$

In this section, we first prove that not all the possible values are achieved by the pair  $(s_b, s_r)$ . Then, some results are provided for illustration of the behavior of these two numbers. For simplicity, we assume that both point sets  $R$  and  $B$  have the same number of points, say  $n$ , since the techniques used in the proofs do not differ significantly when considering a different number of points.

**Proposition 2.** *If  $s_b \geq 1$  then  $n \leq s_b + s_r \leq n(n-1)$ , and both bounds are tight.*

**Proof.** Since  $s_b \geq 1$  then there is a red segment  $\overline{pq}$  intersecting a blue segment  $\overline{uv}$ . Consider the supporting half-lines from  $p$  and  $q$  to  $\overline{uv}$ . Figure 7(a) shows that these half-lines starting from either  $u$  or  $v$  together with the segment  $\overline{uv}$  split the plane into four regions (not bounded, open and disjoint) denoted by I, II, III and IV.

Clearly, if the  $n$  red points are located in regions I and II then there are at least  $n-1$  red segments crossed by  $\overline{uv}$ . This gives  $n \leq s_b + s_r$ . Suppose then that there is at least one red point  $r \in R - \{p, q\}$  in region III (analogously if  $r$  is in region IV). Let  $T$  be the triangle defined by  $p, q$  and  $r$ . Each blue point  $b \in B - \{u, v\}$  is either interior or exterior to  $T$  and so it is an endpoint of at least one intersected blue segment, either  $\overline{bu}$  or  $\overline{bv}$ . This easily implies that  $n \leq s_b + s_r$ .

Obviously, the upper bound of  $s_b + s_r$  is attained when all the blue segments are intersected by red segments and vice versa, i.e.,  $s_b + s_r = \frac{n(n-1)}{2} + \frac{n(n-1)}{2} = n(n-1)$ . Figures 7(b) and 7(c) show that both bounds are tight.  $\square$

We say that a monochromatic simple polygon  $P$  is a *dividing simple polygon* if there are at least two points of the other color, one interior and one exterior to  $P$ . Note that a dividing simple polygon splits the point set of the other color into two

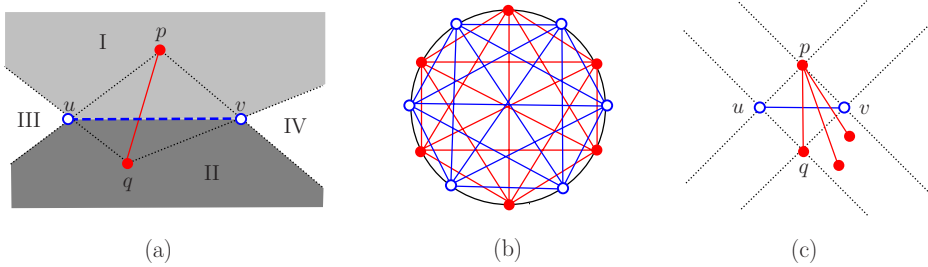


Fig. 7. (a) Regions I, II, III and IV defined by the segments  $\overline{uv}$  and  $\overline{pq}$ , (b) a point configuration where all the segments are intersected, (c) a point configuration where  $s_b + s_r = n$ .

subsets that can be viewed as the vertex set of a bipartite graph with edges the segments intersected by the polygon. Thus, the following result is straightforward.

**Proposition 3.** *If there exists a red dividing simple polygon then  $s_r \geq 1$  and  $s_b \geq n - 1$ . Analogously, if there exists a blue dividing simple polygon then  $s_b \geq 1$  and  $s_r \geq n - 1$ .*

**Proposition 4.** *If  $1 \leq s_b \leq n - 2$  then the following statements hold.*

- (i)  $CH(R)$  contains no blue point.
- (ii)  $s_r \geq n - 1$ , and this bound is tight.
- (iii) The value of  $s_r$  is the number of edges of a complete  $k$ -partite graph with  $n$  vertices and  $2 \leq k \leq \frac{(s_b^2 + s_b + 2)}{2}$ .
- (iv) If the red points are located in  $t \geq 2$  different regions of the arrangement defined by the lines containing the blue segments, then  $s_r \geq (t - 1)(n - \frac{t}{2})$ .

**Proof.** By Proposition 3, to prove statement (i), it suffices to assume on the contrary that all the blue points are located inside  $CH(R)$ . Since  $s_b \geq 1$  then there is at least one blue segment  $\overline{uv}$  intersecting a red segment  $\overline{pq}$ . Join  $p$  and  $q$  to at most two vertices of  $CH(R)$  (depending on whether either  $p$  or  $q$  is a vertex of  $CH(R)$ ) leaving  $u$  inside this red dividing simple polygon and  $v$  outside or vice versa. By Proposition 3 it follows that  $s_b \geq n - 1$  which is a contradiction.

Statement (ii) is shown assuming that either  $CH(R)$  and  $CH(B)$  intersect or they are nested polygons, since  $CH(R)$  contains no blue point (Fig. 8). Then, there is a blue segment  $\overline{uv}$  intersecting two segments of  $CH(R)$ , say  $\overline{p_1q_1}$  and  $\overline{p_2q_2}$ , which easily implies that there exists a blue dividing simple polygon (Fig. 8). By Proposition 3 we have  $s_r \geq n - 1$ . Figure 8(b) shows that this bound is tight.

To prove statement (iii), consider the arrangement defined by all the lines going through any two endpoints of the  $s_b$  intersected blue segments (Fig. 9). The number of regions (both bounded and unbounded) of the plane defined by this arrangement is  $\frac{(s_b^2 + s_b + 2)}{2}$ . Let  $k$  be the number of such regions containing at least one red point (see Fig. 9). Since  $s_b \geq 1$  then  $2 \leq k \leq \frac{(s_b^2 + s_b + 2)}{2}$ . Moreover, every red segment

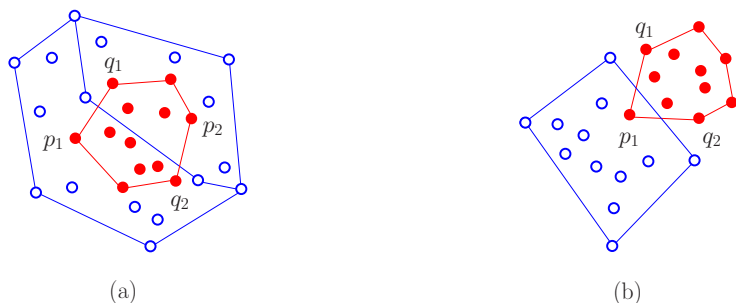


Fig. 8. (a)  $CH(R)$  and  $CH(B)$  are nested polygons, (b) the boundaries of  $CH(R)$  and  $CH(B)$  intersect.

whose endpoints are in different regions intersects at least one blue segment since  $CH(R)$  contains no blue point. Hence the result follows. As a consequence, arises statement (iv). In this case, the value  $s_r$  is the number of edges of a complete  $t$ -partite graph. This number is minimized when  $t - 1$  parts contain a single vertex and so  $s_r \geq \frac{(t-1)(t-2)}{2} + (n - t + 1)(t - 1)$ .  $\square$

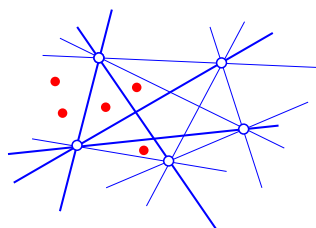


Fig. 9. Arrangement defined by all the lines through two endpoints of the blue segments intersected by red segments, and  $k = 4$ .

**Proposition 5.** If  $s_b = 1$  then  $n - 1 \leq s_r \leq \frac{n^2}{4}$ , and both bounds are tight. Moreover, if  $s_b = \frac{n(n-1)}{2}$  then  $\lceil \frac{(-1 + \sqrt{8n-7})}{2} \rceil \leq s_r \leq \frac{n(n-1)}{2}$ , and both bounds are tight.

**Proof.** Suppose first that  $s_b = 1$ . Then there is exactly one blue segment  $\overline{uv}$  intersecting a red segment  $\overline{pq}$ . By Proposition 3, it is easy to see that all the red points have to be distributed in regions I and II (both shown in Fig. 7(a)), say  $m_1$  and  $m_2 = n - m_1$ , respectively. A bipartite graph is obtained by considering the red points as the vertices and the  $m_1 m_2$  red segments as the edges. Since  $m_1 + m_2 = n$ , it follows that  $m_1 m_2 \leq \frac{n^2}{4}$  (the exact value  $\frac{n^2}{4}$  is attained for  $m_1 = m_2 = \frac{n}{2}$ ) and  $s_r = n - 1$  if and only if  $m_1 = 1$ . Hence  $n - 1 \leq s_r \leq \frac{n^2}{4}$  and both bounds are tight.

Suppose now that  $s_b = \frac{n(n-1)}{2}$ . Obviously,  $s_r \leq \frac{n(n-1)}{2}$  and so it suffices to check the lower bound. Proceeding as in the proof of Proposition 4, consider the arrangement defined by all the lines going through any two endpoints of the  $s_r$  intersected red segments. Since all the blue segments are crossed, then there is at most one blue point in each region defined by the arrangement. Hence,  $n \leq k \leq \frac{s_r^2 + s_r + 2}{2}$  where  $k$  denotes the number of regions containing at least one blue point. This implies the result. Figures 7(b) and 10 show that both bounds are tight.  $\square$

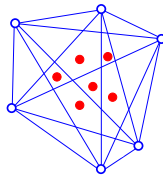


Fig. 10. A point configuration where  $s_b = \lceil \frac{-1 + \sqrt{8n-7}}{2} \rceil = 3$ .

### Relating $s_b$ and $s_r$ with combinatorial depths

An interesting issue to be considered is the connection between the numbers  $s_b$ ,  $s_r$  and the most usual combinatorial depths for point sets studied in the literature: the *convex depth*,<sup>11</sup> the *simplicial depth*<sup>12</sup> and the *Tukey depth*.<sup>18</sup> We first recall their definitions.

The *convex depth* of a point  $q$  with respect to a finite set of points  $\mathcal{P} \subset \mathbb{R}^2$ , denoted by  $d_{\mathcal{P}}^c(q)$ , is defined recursively as follows: If  $q \in CH(\mathcal{P})$  then  $d_{\mathcal{P}}^c(q) = 1$ , else

$$d_{\mathcal{P}}^c(q) = d_{\mathcal{P} \setminus CH(\mathcal{P})}^c(q) + 1.$$

The *simplicial depth* of  $q$  with respect to  $\mathcal{P}$  is the number of simplices generated by points in  $\mathcal{P}$  that contain  $q$ . Finally, the *Tukey depth* or *halfspace depth* of  $q$  with respect to  $\mathcal{P}$  is the minimum number of points in any closed half-plane bounded by a line through  $q$ .

By using the idea of dividing simple polygon, it is easy to establish a first relationship between  $s_b$ ,  $s_r$  and the depths above. For simplicity, let  $d_B(q)$  denote the depth of a red point  $q \in R$  with respect to the blue point set  $B$  for any of the depths.

**Proposition 6.** *For every two red points  $p, q \in R$  such that  $d_B(p) \neq d_B(q)$  it holds that  $s_b \geq n - 1$ . Moreover,  $s_r \geq (d_B(p) - d_B(q))^2$  for the convex depth and also for the Tukey depth, and  $s_r \geq |d_B(p) - d_B(q)|$  for the simplicial depth.*

**Proof.** Clearly, if  $d_B(p) \neq d_B(q)$  then there is a blue dividing simple polygon and so Proposition 3 says that  $s_b \geq n - 1$ .

Consider now two red points  $p, q$  with different simplicial depth with respect to  $B$ . Then, the segment  $\overline{pq}$  intersects at least  $|d_B(p) - d_B(q)|$  different blue triangles and hence  $s_r \geq |d_B(p) - d_B(q)|$ . In the case of the convex depth and the Tuckey depth, it can be easily seen that the segment  $\overline{pq}$  has to intersect at least  $|d_B(p) - d_B(q)|$  disjoint blue segments, and so  $\overline{pq}$  splits the extremes of those segments into two subsets  $B_1, B_2$  of size  $|d_B(p) - d_B(q)|$ , and all the blue segments with one extreme in  $B_1$  and the other in  $B_2$  are intersected by  $\overline{pq}$ . Therefore,  $s_r \geq (d_B(p) - d_B(q))^2$ .  $\square$

## 5. Conclusions

In this paper, we have introduced the following variation of the bichromatic segment intersection problem: *given two sets of colored points defining two sets of segments, report the set of segments of each color intersected by segments of the other color*. We have provided an  $O(n^2)$  time and space algorithm for solving the problem. We have also proved that the problem is 3-Sum-hard and analyzed the difference between computing the exact number of intersected segments of each color and deciding the existence of at least one bichromatic intersection. Finally, we have studied several point configurations as illustrative examples.

As a future work, it would be interesting to extend this problem to 3D, i.e., given two sets of points in 3D, consider the same problem but using monochromatic triangles instead of segments. The goal is to improve the trivial brute force algorithm for the problem. Also, it would be interesting to deepen the study regarding combinatorial depths.

## References

1. L. Arge, G. S. Brodal and L. Georgiadis, Improved dynamic planar point location, *Proc. 47th Annual IEEE Symp. Foundations of Computer Science (FOCS)* (2006), pp. 1–10.
2. P. K. Agarwal, Partitioning arrangements of lines: II. Applications, *Discr. Comput. Geom.* **5** (1990) 533–573.
3. I. Balaban, An optimal algorithm for finding segments intersections, *Proc. 11th Annual ACM Symp. Comput. Geom.* (1995), pp. 211–219.
4. J. Basch, L. J. Guibas and G. D. Ramkumar, Reporting red-blue intersections between two sets of connected line segments, *Proc. 4th Annual European Symp. Algorithms*, Lecture Notes Comput. Sci., Vol. 1136 (Springer-Verlag, 1996), pp. 302–319.
5. J. Bentley and T. Ottmann, Algorithms for reporting and counting intersections, *IEEE Trans. Comput.* **C-28** (1979) 643–647.
6. B. Chazelle, Cutting hyperplanes for divide-and-conquer, *Discr. Comput. Geom.* **9** (1993) 145–158.
7. B. Chazelle and H. Edelsbrunner, An optimal algorithm for intersecting line segments in the plane, *J. Assoc. Comput. Mach.* **39** (1992) 1–54.
8. S.-W. Cheng and R. Janardan, Algorithms for ray-shooting and intersection searching, *J. Algorithms* **13**(4) (1992) 670–692.
9. U. Finke and K. Hinrichs, Overlaying simply connected planar subdivisions in linear time, *Proc. 11th Annual ACM Symp. Comput. Geom.* (1995), pp. 119–126.

10. A. Gajentaan and M. Overmars, On the class of  $O(n^2)$  problems in computational geometry, *Comput. Geom.: Theor. Appl.* **5** (1995) 165–185.
11. P. J. Hubert, Robust statistics: A review, *Ann. Math. Stat.* **43**(3) (1972) 1041–1067.
12. R. Y. Liu, On a notion of data depth based on random simplices, *Ann. Stat.* **18**(1) (1990) 405–414.
13. H. G. Mairson and J. Stolfi, Reporting and counting intersections between two sets of line segments, *Theoretical Foundations of Computer Graphics and CAD*, NATO ASI, Vol. F40, eds. R. A. Earnshaw (Springer-Verlag, Berlin, 1988), pp. 307–325.
14. N. Megiddo, Linear-time algorithms for linear programming in  $\mathbb{R}^3$  and related problems, *SIAM J. Comput.* **12**(4) (1983) 759–776.
15. D. M. Mount, Geometric intersection, *Handbook of Discrete and Computational Geometry*, 2nd edn. (2004), pp. 857–876.
16. J. Snoeyink, Point location, *Handbook of Discrete and Computational Geometry*, 2nd edn. (2004), pp. 559–574.
17. G. T. Toussain, Solving geometric problems with the rotating calipers, *Proc. 2nd IEEE Mediterranean Electrotechnical (MELECON'83)* (1983), A 10.02/1–4.
18. J. Tukey, Mathematics and picturing of data, *Proc. Int. Congress of Mathematics*, Vol. 2 (1975), pp. 523–531.