

# Stabbing Segments with Rectilinear Objects

Mercè Claverol<sup>1</sup>, Delia Garijo<sup>2</sup>, Matias Korman<sup>3,4</sup>, Carlos Seara<sup>1</sup>,  
and Rodrigo I. Silveira<sup>1</sup> (✉)

<sup>1</sup> Universitat Politècnica de Catalunya, Barcelona, Spain  
`rodrigo.silveira@upc.edu`

<sup>2</sup> Universidad de Sevilla, Seville, Spain

<sup>3</sup> National Institute of Informatics, Tokyo, Japan

<sup>4</sup> JST, ERATO, Kawarabayashi Large Graph Project, Tokyo, Japan

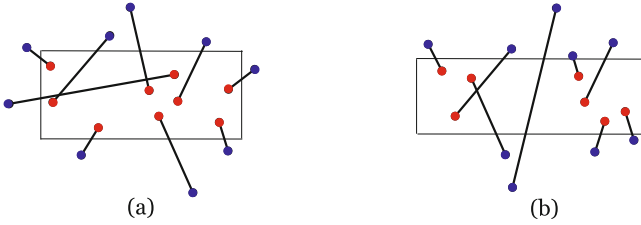
**Abstract.** We consider *stabbing regions* for a set  $S$  of  $n$  line segments in the plane, that is, regions in the plane that contain exactly one endpoint of each segment of  $S$ . Concretely, we provide efficient algorithms for reporting all combinatorially different stabbing regions for  $S$  for regions that can be described as the intersection of axis-parallel halfplanes; these are halfplanes, strips, quadrants, 3-sided rectangles, and rectangles. The running times are  $O(n)$  (for the halfplane case),  $O(n \log n)$  (for strips, quadrants, and 3-sided rectangles), and  $O(n^2 \log n)$  (for rectangles).

## 1 Introduction

Let  $S$  be a set of  $n$  line segments in the plane. We say that a region  $\mathcal{R} \subseteq \mathbb{R}^2$  is a *stabbing region* for  $S$  if  $\mathcal{R}$  contains exactly one endpoint of each segment of  $S$ ; see Fig. 1(a). Depending on the segment configuration, a stabbing region of a certain shape may not exist, as shown in Fig. 1(b). Our aim in this paper is to compute efficiently all stabbing regions  $\mathcal{R}$  for a given set of segments  $S$ , for regions  $\mathcal{R}$  that can be described as the intersection of axis-parallel (i.e., horizontal or vertical) halfplanes. Thus, the shapes here considered are halfplanes, strips, quadrants, 3-sided rectangles, and rectangles. This problem fits into the general framework of *classification or separability problems*, since any stabbing region implicitly classifies all endpoints of  $S$  into two sets: the ones inside  $\mathcal{R}$  (including those on its boundary) are the *red* points, and the ones outside  $\mathcal{R}$  are the *blue* points. Thus, we focus on computing the *combinatorially different* stabbing regions for  $S$ , which are those that provide a different classification of the endpoints of the segments in  $S$ .

Separability and classification problems have been widely investigated and arise in many diverse problems in computational geometry. In our context, perhaps the simplest stabbing region one can consider is a halfplane, whose boundary is defined by a line, and so it is equivalent to a line that intersects all segments (thus classifying their endpoints). Edelsbrunner et al. [13] presented an  $O(n \log n)$  time algorithm for solving the problem of constructing a representation of all stabbing lines (with any orientation) of a set  $S$  of  $n$  line segments.

When no stabbing halfplane exists, it is natural to ask for other types of stabbing regions. Claverol et al. [10] studied the problem of reporting all



**Fig. 1.** (a) A set of segments that has a stabbing rectangle. (b) A set of segments for which no stabbing rectangle exists.

combinatorially different stabbing *wedges* (i.e., the stabbing region defined by the intersection of two halfplanes) for a set  $S$  of  $n$  segments; see also [9]. Their algorithm runs in  $O(n^3 \log n)$  time and uses  $O(n^2)$  space. They also studied some other stabbers such as double-wedges and zigzags (see [10] for a table comparing the time and space complexities for these stabbers).

*Results.* Following this line of research, we introduce new shapes of stabbing regions and exploit their geometric structure to obtain efficient algorithms. Concretely, in Sect. 2, we study the case in which the stabbing region is formed by at most two halfplanes, that is, halfplanes, strips, and quadrants. Our approach partitions the plane into three regions: a red one that must be contained in any stabbing region, a blue one that must be avoided by any stabbing region, and a gray region for which we do not have enough information yet. The algorithms are based on iteratively classifying segments and updating the boundaries of these regions, a process that we call *cascading*. The running times obtained are  $O(n)$  (for the halfplane case), and  $O(n \log n)$  (for strips and quadrants). In Sect. 3 we show that the cascading approach can be extended to 3-sided rectangles, and that the number of combinatorially different solutions is still  $O(n)$ , resulting in an  $O(n \log n)$  algorithm. Finally, we focus on stabbing rectangles in Sect. 4, for which our algorithm runs in  $O(n^2 \log n)$  time. This is close to being worst-case optimal, since there can be  $\Theta(n^2)$  combinatorially different solutions.

Note that even though we present our algorithms for stabbing regions defined by axis-parallel halfplanes, they extend to any two fixed orientations by making an appropriate affine transformation.

*Other Related Work.* Díaz-Báñez et al. [12] considered a similar stabbing concept: a region  $\mathcal{R}$  stabs a collection of segments if *at least* one endpoint of each segment is in  $\mathcal{R}$ . In this setting, the endpoints of the segments are not necessarily classified. Moreover, existence of a stabber is always guaranteed (one can always find a large enough region that contains all segments). Thus, all studied problem focus on optimization. For instance, they search for a polygonal stabber with minimum perimeter or area. In their work, they show that the general problem is NP-hard and provide polynomial-time algorithms for some particular cases, like disjoint segments. Other relevant references on stabbing problems that focus

on optimization problems in two dimensions are [3, 6, 7, 15, 17–20, 22]. Variants of these problems have also been studied in three dimensions (e.g. [4, 8, 14, 16, 21]).

A more general formulation for the above-mentioned problems is using color-spanning objects. In this case, the input is a set of  $n$  colored points, with  $c$  colors, and the goal is to find an object (rectangle, circle, etc.) that contains *at least* (or *exactly*)  $k$  points of each color. Our setting is the particular case in which  $c = n/2$  and we want to contain *exactly* one point of each color class. The color-spanning objects (for the *at least* objective) that have been studied in the literature are strips, axis-parallel rectangles [1, 11], and circles [2]. All cases can be solved in roughly  $O(n^2 \log c)$  time. Less research has been done for the *exact* objective. Among others, we highlight the research of Barba et al. [5]. In this work they give algorithms that, in  $O(n^2c)$  time, compute disks, squares, and axis-aligned rectangles that contain exactly one element of each of the  $c$  color classes. The algorithm that we present in Sect. 4 (axis-aligned rectangle stabber) improves the result of [5] for the particular case in which  $c = n/2$  and one looks for an axis-aligned color spanning rectangle. Our algorithm is almost a linear factor faster, and also allows to report all possible solutions (whereas their algorithm only reports one region).

*Some Notation.* The input consists of a set  $S = \{s_1, \dots, s_n\}$  of  $n$  segments. For simplicity, we assume that there is no horizontal or vertical segment in  $S$ , and that all segments have non-zero length. The modifications needed to make our algorithms handle these special cases are straightforward, albeit rather tedious. For any  $1 \leq i \leq n$ , let  $p_i$  and  $q_i$  denote the upper and lower endpoint of  $s_i$ , respectively. Given a point  $p \in \mathbb{R}^2$ , we write  $x(p)$  (resp.  $y(p)$ ) for its  $x$ - (resp.  $y$ -) coordinate. Let  $y_b = \max_{s_i \in S} \{y(q_i)\}$  and  $y_t = \min_{s_i \in S} \{y(p_i)\}$ ; these values correspond to the  $y$ -coordinates of the highest bottom endpoint and the lowest top endpoint, respectively, of the segments in  $S$ . The segments attaining those values are denoted by  $s_b$  and  $s_t$ , respectively (i.e.,  $y(q_b) = y_b$  and  $y(p_t) = y_t$ ).

## 2 Stabbing with One or Two Halfplanes

This section deals with stabbing regions that can be described as the intersection of at most two halfplanes. That is, our aim is to obtain a halfplane, strip, or quadrant that contains exactly one endpoint of each segment of  $S$ . Note that such stabbing objects do not always exist.

### 2.1 Stabbing Halfplane

For completeness (since it will be used in the upcoming sections), we explain a straightforward algorithm for determining if a horizontal stabbing halfplane exists. That is, a horizontal line such that one of the (closed) halfplanes defined by the line contains exactly one endpoint of each segment. Observe that such a stabbing halfplane can be *perturbed* so that it has no endpoint of a segment on its boundary. In this case, the complement of a stabbing halfplane is also a

stabber. Thus, we are effectively looking for a horizontal line that intersects the interior of all segments. As we are dealing with horizontal stabbers, the problem becomes essentially one-dimensional, and it will ease our presentation to state the problem in that way.

All segments can be projected onto the  $y$ -axis, becoming intervals. Considering the set  $\bar{S} = \{\bar{s}_1, \dots, \bar{s}_n\}$  of projected segments, the question is simply whether all the intervals in  $\bar{S}$  have a point in common. Clearly, any horizontal line  $y := u$  stabbing  $S$  must have its  $y$ -coordinate between the values  $y_b$  and  $y_t$ , namely  $y_b \leq u \leq y_t$ . Therefore, such a line exists if and only if  $y_b \leq y_t$ . Moreover, whenever this condition happens, both the upper and lower halfplanes will be stabbing halfplanes (and any other horizontal halfplane will be equivalent to one of the two). This simple observation directly leads to a linear-time algorithm.

**Observation 1.** *All axis-aligned stabbing halfplanes of a set of  $n$  segments can be found in  $O(n)$  time.*

## 2.2 Stabbing Strips

We now consider the case in which the stabbing region is a horizontal strip. Note that the existence of stabbing halfplanes directly implies the existence of stabbing strips, but the reverse is not true. Thus, our aim is to compute all *non-trivial* stabbing strips; intuitively speaking, we are not interested in stabbing strips that can be extended to stabbing halfplanes. More formally, we say that a stabbing region  $\mathcal{R}$  is *non-trivial* if there is no stabbing region described as the intersection of fewer halfplanes that is combinatorially equivalent to  $\mathcal{R}$  (i.e., gives the same classification of the endpoints of the segments in  $S$ ).

As in Sect. 2.1, we can ignore the  $x$ -coordinates of the endpoints, project the points onto the  $y$ -axis, and work with the set  $\bar{S}$  instead. The endpoints of the classified segments can be seen in the projection onto the  $y$ -axis as a set of blue and red points. It follows that there is a separating horizontal strip for them if and only if the red points appear contiguously on the  $y$ -axis. More precisely, the points must appear on the  $y$ -axis in three contiguous groups, from top to bottom, first a blue group, then a red group, and then another blue group. We refer to the two groups of blue points as the *top* and *bottom* blue points, respectively. We denote the intervals of the  $y$ -axis spanned by them by  $B_t$  and  $B_b$ , respectively. The interval of the  $y$ -axis spanned by the group of red points is denoted by  $R$ . Since all points above  $B_t$  and below  $B_b$  must be necessarily blue, we extend  $B_t$  and  $B_b$  from  $+\infty$  and until  $-\infty$ , respectively. Thus, the  $y$ -axis is partitioned into three colored intervals and two uncolored (gray) intervals separating them. See Fig. 2(a).

**Observation 2.** *Let  $s_i$  and  $s_j$  be two segments in  $S$  such that  $\bar{s}_i$  is above  $\bar{s}_j$  (in particular, this implies that the projected segments  $\bar{s}_i$  and  $\bar{s}_j$  are disjoint). Then, any horizontal stabbing strip must contain  $q_i$  and  $p_j$ .*

**Lemma 1.** *Any non-trivial horizontal stabbing strip for  $S$  contains points  $q_b$  and  $p_t$ .*

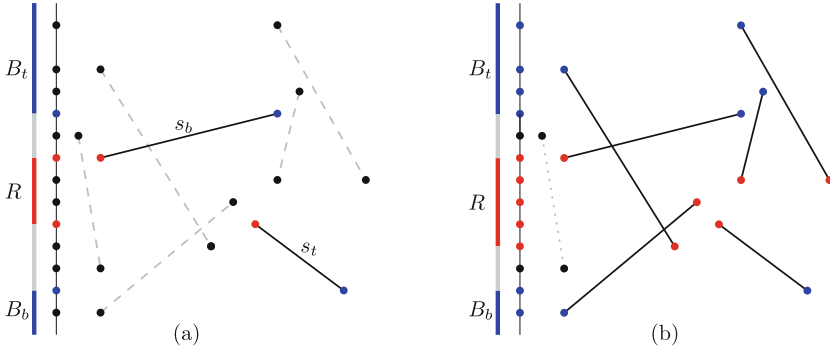
With the previous observations in place, we can present our algorithm. We give an intuitive idea in rather general terms because it will also be used in the upcoming sections. Our algorithm starts by classifying a few segments of  $S$  using some geometric observations (say, Lemma 1). As soon as some points are classified, we partition the plane into three regions: the *red* region (a portion of the plane that must be contained by any stabber), the *blue* region (a portion of the plane that cannot be contained in any stabber) and the *gray* region (the complement of the union of the two other regions; that is, regions of the plane for which we still do not know). If a segment of  $S$  has an endpoint in either the blue or red regions we can classify it (that is, if an endpoint is in a red region, that endpoint must be red, and its opposite endpoint blue). This coloring may enlarge either the red or blue regions, which may further allow us to classify other segments of  $S$ , and so on. We call this process the *cascading procedure*. This approach will continue until either we find a contradiction (say, the red and blue regions overlap), or we have classified all segments of  $S$  (and thus we have found a stabber). Thus, at any instant of time we partition  $S$  into three sets  $C$ ,  $W$ , and  $U$ . A segment is in  $C$  if it has been already classified, in  $W$  if it is waiting for being classified (there is enough information to classify it, but it has not been done yet), or in  $U$  if its classification is still unknown. The algorithm is initialized with  $C = W = \emptyset$ , and  $U = S$ .

*Regions.* In addition to the three sets of segments, the algorithm maintains red and blue candidate regions that are guaranteed to be contained or avoided in any solution, respectively. When we are looking for a horizontal strip, these regions will also be horizontal strips. Thus, it suffices to maintain the projection of the regions on the  $y$ -axis. The blue and red regions are represented by the intervals  $B_t, B_b$  and  $R$ : the *blue region* is  $B_t \cup B_b$ , and the *red region* is  $R$ . The complement of  $B_t \cup B_b \cup R$  is called the *gray region*. Note that the gray region, like the blue one, consists of two disjoint components. During the execution of the algorithm the regions will be updated as new segments become classified. See Fig. 2(a).

The algorithm starts by computing  $s_b$  and  $s_t$ . If a stabbing halfplane exists, one can find a stabbing strip and report it. Otherwise, by Lemma 1, we know how to classify both segments (i.e.,  $q_b$  and  $p_t$  are classified as red, and  $p_b$  and  $q_t$  as blue). Thus, we move them from  $U$  to  $W$ . See Fig. 2.

*Cascading Procedure.* The procedure iteratively classifies segments in  $W$  based on the red and blue regions. This is an iterative process in the sense that the classification of one segment can make the blue or red region grow, making other segments move from  $U$  to  $W$ .

As long as  $W$  is not empty, we pick any segment  $s \in W$ , assign the corresponding colors to its endpoints, and move  $s$  from  $W$  to  $C$ . If a newly assigned endpoint lies outside its corresponding zone, the red or blue area must grow to contain that point. Note that after the red or blue region grows, other segments can change from  $U$  to  $W$ . The process continues classifying segments of  $W$  until either: (i) a contradiction is found (the red region is forced to overlap with the blue region), or (ii) set  $W$  becomes empty.



**Fig. 2.** Computing a stabbing horizontal strip. (a) Result after classifying  $s_t$  and  $s_b$ . (b) Result after cascading; the red region has grown, and only one segment remains unclassified. In both figures, segments of  $U$  are shown dotted, those of  $W$  are dashed, and those of  $C$  are depicted with a solid line.

**Lemma 2.** *If the cascading procedure finds no contradiction, each remaining segment in  $U$  has an endpoint in each of the connected components of the gray region.*

*Proof.* By definition, when cascading procedure finishes, all segments still in  $U$  must have both endpoints in the gray region. Assume, for the sake of contradiction, that there exists a segment  $\bar{s}_i \in U$  whose both endpoints lie in the same gray component, say, the lower one. Recall that, by construction, the red region contains the interval  $[y_b, y_t]$ . In particular, we have  $y(p_i) < y_t$ , giving a contradiction with the definition of  $y_t$ .  $\square$

**Corollary 1.** *A horizontal stabbing strip exists for  $S$  if and only if the cascading procedure finishes without finding a contradiction.*

**Theorem 1.** *Determining whether a horizontal stabbing strip exists for a set of  $n$  segments can be done in  $O(n \log n)$  time and  $O(n)$  space.*

*Reporting All Horizontal Stabbing Strips.* The above algorithm can be modified to report all combinatorially different horizontal stabbing strips without an increase in the running time.

Once the cascading procedure has finished, we define  $\tau$  as the index of the segment of  $U$  whose upper endpoint is lowest (i.e., for any index  $i$  such that  $s_i \in U$ , it holds that  $y(p_i) \geq y(p_\tau)$ ). Likewise, we define  $\beta$  as the index whose lower endpoint is highest. Any stabbing strip will either contain: (i) all points in the upper component of the gray region, (ii) all points in the lower component of the gray region, or (iii) both  $p_\tau$  and  $q_\beta$ . The first two can be reported in constant time, and for the third case, it suffices to classify the two segments, cascade, and repeat the previous steps.

**Theorem 2.** *All the combinatorially different horizontal stabbing strips of a set of  $n$  segments can be computed in  $O(n \log n)$  time.*

### 2.3 Stabbing Quadrants

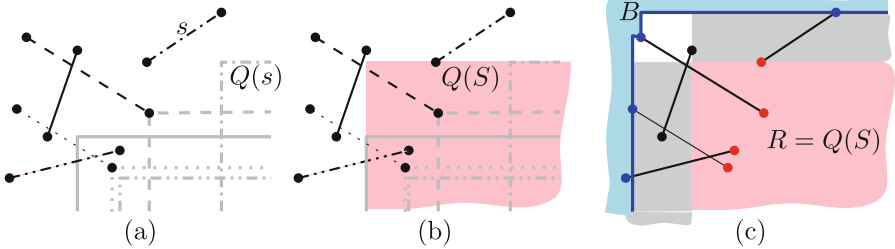
We now extend this approach for stabbing quadrants. There are four types of quadrants; without loss of generality, we concentrate on the bottom-right type. Thus throughout this section, the term *quadrant* refers to a bottom-right one. Other types can be handled analogously.

For a segment  $s = (p, q)$ , let  $Q(s)$  denote its *bottom-right quadrant*; that is, the quadrant with apex at  $(\max\{x(p), x(q)\}, \min\{y(p), y(q)\})$ . See Fig. 3(a).

**Observation 3.** *Any quadrant classifying a segment  $s \in S$  must contain  $Q(s)$ .*

Given the segment set  $S$ , the *bottom-right quadrant* of  $S$ , denoted by  $Q(S)$ , is the (inclusion-wise) smallest quadrant that contains  $\cup_{s \in S} Q(s)$ ; see Fig. 3(b). We now need the equivalent of Lemma 1 to create the initial partition of the plane into red, blue, and gray regions.

**Corollary 2.** *Any stabbing quadrant of  $S$  must contain  $Q(S)$ .*



**Fig. 3.** (a) A set  $S$  of five segments and their individual bottom-right quadrants. (b) Bottom-right quadrant  $Q(S)$  of the set of segments  $S$ . (c) Initial classification given by  $Q(S)$ , and the red, blue, and gray regions. Note that there is a region (white in the figure) that cannot contain endpoints of  $S$ .

*Regions.* Partition is as follows: the red region  $R$  is always defined as the inclusionwise smallest quadrant that contains all classified points and  $Q(S)$ . At any point in the execution, let  $a = (x_R, y_R)$  denote the apex of  $R$ . Any blue point  $b$  of a classified segment forbids the stabber to include  $b$  or any point above and to the left of  $b$  (i.e., in the top-left quadrant of  $b$ ). Moreover, if  $b$  satisfies  $y(b) \leq y_R$  or  $x(b) \geq x_R$ , a whole halfplane will be forbidden. Thus, the union of such regions is bounded by a staircase polygonal line (see Fig. 3(c)). Initially, we take the blue region  $B$  defined by a point at  $(-\infty, \infty)$ . We say that a point  $p = (x, y)$  is in the gray region if it is not in the red or blue region, and satisfies either  $x > x_R$  or  $y < y_R$  (see Fig. 3(c)). As in Sect. 2.2, observe that the gray region is the union of two connected components (which we call *right*, and *down* components). Note that, in this case, there is a region, which we call *white*, that is not contained in either the red, blue, or gray regions. However, our first observation is that no endpoint of an unclassified segment can lie in the white region.

**Observation 4.** *A segment  $s \in S$  containing an endpoint in the white region must contain its other endpoint in the red region.*

We initially classify all segments that have one endpoint inside  $Q(S)$  in  $W$  (and all the rest in  $U$ ). As before, we apply the cascading procedure until a contradiction is found (in which case we conclude that a stabbing quadrant does not exist), or set  $W$  eventually becomes empty. In this case we have a very strong characterization of the remaining unclassified segments.

**Lemma 3.** *If the cascading procedure finishes without finding a contradiction, each remaining segment in  $U$  has an endpoint in each of the gray components.*

Thus, if no contradiction is found we can extend  $R$  until it contains one of the two gray components to obtain a stabber. This can be done because, by construction, each of the connected components of the gray region forms a bottomless rectangle (i.e., the intersection of three axis aligned halfplanes) that shares a corner with the apex of  $R$ . As in the strip case, we can use this approach to report all combinatorially different quadrants in an analogous fashion to Theorem 1: any stabbing quadrant will either completely contain one of the two gray components, or it will contain at least a segment of each of the two components.

**Theorem 3.** *All the combinatorially different stabbing quadrants of a set of  $n$  segments can be computed in  $O(n \log n)$  time.*

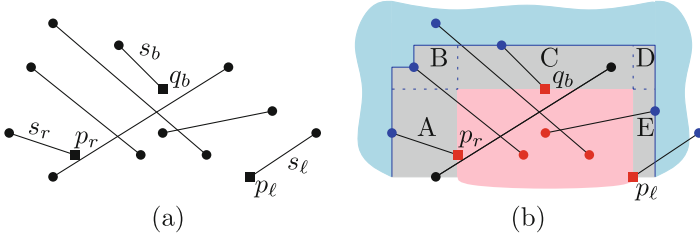
### 3 Stabbing with Three Halfplanes

We now consider the case in which the stabber is defined as the intersection of three halfplanes. As in the quadrant case, it suffices to consider those of fixed orientation. Thus, throughout this section, a *3-rectangle* refers to a rectangle that is missing the lower boundary edge, and that extends infinitely towards the negative  $y$ -axis (also called *bottomless rectangle*). Without loss of generality, we may assume that  $S$  cannot be stabbed with a halfplane, strip, or quadrant (since any of those regions can be transformed into a bottomless rectangle). As in the previous cases, we solve the problem by partitioning the plane into red-blue-gray regions. However, to generate all stabbing 3-rectangles we need a more involved sweeping phase that is combined with further cascading iterations.

#### 3.1 Number of Different Stabbing 3-Rectangles

First we analyze the number of combinatorially different stabbing 3-rectangles that a set of  $n$  segments may have. This analysis will lead to an efficient algorithm to compute them. We start by defining a region that must be included in any stabbing 3-rectangle. Recall that  $s_b$  is the segment of  $S$  with highest bottom endpoint, and  $q_b$  is its bottom endpoint. Analogously, we define  $s_r$  and  $s_\ell$  as the segments with leftmost right endpoint and rightmost left endpoint, respectively.





**Fig. 4.** (a) Example with points  $q_b$ ,  $p_r$ , and  $p_\ell$  highlighted as squares. (b) Red and blue regions after the initial cascading procedure has finished, and partition of the gray region into subregions  $A$ ,  $B$ ,  $C$ ,  $D$ ,  $E$  (Color figure online).

Their corresponding right and left endpoints are denoted by  $p_r$  and  $p_\ell$ . See Fig. 4. Finally, we define lines  $L_\ell$ ,  $L_r$  as the vertical lines passing through  $p_\ell$  and  $p_r$ , respectively; analogously,  $L_b$  is the horizontal line passing through  $q_b$ .

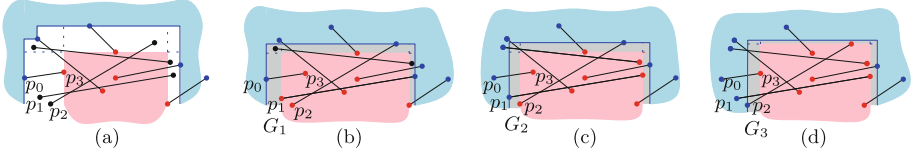
**Lemma 4.** *Any non-trivial stabbing 3-rectangle  $\mathcal{R}$  must contain the intersection points of lines  $L_\ell$ ,  $L_r$  and  $L_b$ .*

The above result allows us to initialize the red region and start the usual cascading procedure. If no contradictions are found, the classified segments define a red and a blue region which must be avoided and included in any solution, respectively. Specifically, the red region is the inclusionwise smallest 3-rectangle that contains all points classified as red. The blue region is the set of points whose inclusion would force a 3-rectangle to contain some point that has already been classified as blue. Finally, the area that is neither red nor blue is called *gray region*. Once the cascading procedure has finished, all remaining unclassified segments must have both of their endpoints in the gray region. It will be convenient to distinguish between different parts of the gray region, depending on their position with respect to the red region. We differentiate between five regions, named  $A$ ,  $B$ ,  $C$ ,  $D$ ,  $E$ , as depicted in Fig. 4(b). These five regions are obtained by drawing horizontal and vertical lines through the two corners of the red region. We say that the *type* of a segment  $s$  is  $XY$ , for  $X, Y \in \{A, B, C, D, E\}$  if  $s$  has one endpoint in region  $X$  and the other endpoint in region  $Y$ . The next lemma shows that, after a cascading, there are only a few possible types.

**Lemma 5.** *Any unclassified segment after the cascading procedure is of type  $AC$ ,  $AD$ ,  $AE$ ,  $BE$ , or  $CE$ .*

Now, consider the red and blue regions obtained after a cascading. Let  $p_1, \dots, p_k$  be the endpoints of segments of  $S$  inside region  $A$ . In addition, we define  $p_0$  as the blue point defining the blue boundary of region  $A$ , and  $p_{k+1}$  as the red point defining the red boundary of region  $A$ . See Fig. 5(a). Let  $G_i$ , for  $1 \leq i \leq k + 1$ , be the gray region obtained after classifying points  $p_1, \dots, p_{i-1}$  as blue,  $p_i, \dots, p_k$  as red, and performing a cascading procedure (see Fig. 5(b-d)). If any of those cascading operations results in a contradiction being found, we

simply set the corresponding  $G_i$  to be empty. Observe that, if for some  $i$ , the region  $G_i$  is not empty, any unclassified segment must be of type BE or CE. We say that a classification of the remaining segments is *compatible with  $G_i$*  if it is compatible with the classification of just classified segments. Next we bound the maximum number of combinatorially different solutions compatible with  $G_i$ .



**Fig. 5.** (a) Initial situation; region A contains points  $\{p_1, p_2\}$ . (b)-(d): sequence of gray regions  $G_1, G_2, G_3$  (Color figure online).

**Lemma 6.** *Let  $G_i$  be defined as above for some  $1 \leq i \leq k + 1$  such that  $G_i \neq \emptyset$ , and let  $n_i$  be the number of unclassified segments in  $G_i$ , i.e., segments with both endpoints in  $G_i$ . Then there are at most  $n_i$  combinatorially different solutions compatible with  $G_i$ .*

**Lemma 7.** *For any set  $S$  of  $n$  segments there are  $O(n)$  combinatorially different stabbing 3-rectangles.*

### 3.2 Algorithm

The previous results give rise to a natural algorithm to generate all combinatorially different stabbing 3-rectangles. The algorithm has two phases:

**Initial Cascading.** We initialize the red region with Lemma 4 and launch the usual cascading procedure. If this cascading finishes without finding a contradiction, we obtain a red and a blue region that must be included and avoided by any stabbing 3-rectangle, respectively.

**Plane Sweep of Region A.** We sweep the points in region A from left to right. In the  $i$ -th step of the sweep, we classify points  $p_1, \dots, p_{i-1}$  as blue, and points  $p_i, \dots, p_k$  as red. After each such step, we perform a cascading procedure. If the cascading gives no contradiction, we are left with a gray region  $G_i$  and a number of unclassified segments that must be of type BE or CE. Then we sweep the endpoints of the unclassified segments in region E from left to right (we call this the *secondary* sweep). At each step of the sweep, we fix those to the left of the sweep line as red, and those to the right of the sweep line as blue, and perform a cascading procedure. From the proof of Lemma 6, we know that each step of this second sweep, after the corresponding cascading procedure, can produce at most one different solution.

**Theorem 4.** *All combinatorially different stabbing 3-rectangles of a set of  $n$  segments can be computed in  $O(n \log n)$  time.*

## 4 Stabbing Rectangles

Applying the cascading approach of the previous sections to rectangles results in a rather involved case distinction, since segments with both endpoints in gray regions can have many different interdependences. Instead, we can use a simple approach based on the algorithm for 3-rectangles, which results in a running time that is close to optimal in the worst case (since it is easy to see that there can be  $\Theta(n^2)$  combinatorially different stabbing rectangles).

Any inclusionwise smallest stabbing rectangle  $R$  must contain one endpoint on each side, and in particular, an endpoint  $v$  of a segment in  $S$  must be on its lower boundary segment (otherwise, we could shrink it further). The key observation is that if we fix  $v$  (or equivalently, the lower side of a candidate stabbing rectangle), then we can reduce the problem to that of finding a 3-sided rectangle. In particular, by fixing the lower side of the rectangle we are forcing all points below it to be blue, and the point through the fixed side to be red. After a successful cascading procedure, we end up with a certain initial classification that can be completed to a solution to the rectangle if and only if a compatible stabbing 3-sided rectangle exists. Since there are  $2n$  candidates for  $v$  (each of the endpoints of the segments of  $S$ ), we have  $O(n)$  different instances that are solved independently using Theorem 4.

**Theorem 5.** *All the combinatorially different stabbing rectangles of a given set  $S$  of  $n$  segments can be computed in  $O(n^2 \log n)$  time.*

**Acknowledgments.** M. C., C. S., and R.S were partially supported by projects MINECO MTM2012-30951 and Gen. Cat. DGR2014SGR46. D. G. was supported by project PAI FQM-0164. R.S. was partially funded by the Ramón y Cajal program (MINECO).

## References

1. Abellanas, M., Hurtado, F., Icking, C., Klein, R., Langetepe, E., Ma, L., Palop, B., Sacristán, V.: Smallest color-spanning objects. In: Meyer auf der Heide, F. (ed.) ESA 2001. LNCS, vol. 2161, pp. 278–292. Springer, Heidelberg (2001)
2. Abellanas, M., Hurtado, F., Icking, C., Klein, R., Langetepe, E., Ma, L., Palop, B., Sacristán, V.: The farthest color Voronoi diagram and related problems. In: Proceedings of the 17th European Workshop on Computational Geometry, pp. 113–116 (2001)
3. Atallah, M., Bajaj, C.: Efficient algorithms for common transversal. *Inf. Process. Lett.* **25**, 87–91 (1987)
4. Avis, D., Wenger, R.: Polyhedral line transversals in space. *Discrete Comput. Geom.* **3**, 257–265 (1988)
5. Barba, L., Durocher, S., Fraser, R., Hurtado, F., Mehrabi, S., Mondal, D., Morrison, J., Skala, M., Wahid, M.A.: On  $k$ -enclosing objects in a coloured point set. In: Proc. of the 25th Canadian Conference on Computational Geometry, pp. 229–234 (2013)

6. Bhattacharya, B.K., Czyzowicz, J., Egyed, P., Toussaint, G., Stojmenovic, I., Urrutia, J.: Computing shortest transversals of sets. In: Proceedings of the 7th Annual Symposium on Computational Geometry, pp. 71–80 (1991)
7. Bhattacharya, B., Kumar, C., Mukhopadhyay, A.: Computing an area-optimal convex polygonal stabber of a set of parallel line segments. In: Proceedings of the 5th Canadian Conference on Computational Geometry, pp. 169–174 (1993)
8. Brönnimann, H., Everett, H., Lazard, S., Sottile, F., Whitesides, S.: Transversals to line segments in three-dimensional space. *Discrete Comput. Geom.* **34**, 381–390 (2005)
9. Claverol, M.: Problemas geométricos en morfología computacional. Ph.D. thesis, Universitat Politècnica de Catalunya (2004)
10. Claverol, M., Garijo, D., Grima, C.I., Márquez, A., Seara, C.: Stabbers of line segments in the plane. *Comput. Geom. Theor. Appl.* **44**(5), 303–318 (2011)
11. Das, S., Goswami, P.P., Nandy, S.C.: Smallest color-spanning objects revisited. *Int. J. Comput. Geom. Appl.* **19**(5), 457–478 (2009)
12. Díaz-Báñez, J.M., Korman, M., Pérez-Lantero, P., Pilz, A., Seara, C., Silveira, R.I.: New results on stabbing segments with a polygon. *Comput. Geom. Theor. Appl.* **48**(1), 14–29 (2015)
13. Edelsbrunner, H., Maurer, H.A., Preparata, F.P., Rosenberg, A.L., Welzl, E., Wood, D.: Stabbing line segments. *BIT* **22**, 274–281 (1982)
14. Fogel, E., Hemmer, M., Porat, A., Halperin, D.: Lines through segments in three dimensional space. In: Proceedings of the 29th European Workshop on Computational Geometry, Assisi, Italy, pp. 113–116 (2012)
15. Goodrich, M.T., Snoeyink, J.S.: Stabbing parallel segments with a convex polygon. In: Proceedings 1st Workshop Algorithms and Data Structures, pp. 231–242 (1989)
16. Kaplan, H., Rubín, N., Sharir, M.: Line transversal of convex polyhedra in  $\mathbb{R}^3$ . *SIAM J. Comput.* **39**(7), 3283–3310 (2010)
17. Lyons, K.A., Meijer, H., Rappaport, D.: Minimum polygon stabbers of isothetic line segments. Dept. of Computing and Information Science, Queen’s University, Canada (1990)
18. Mukhopadhyay, A., Kumar, C., Greene, E., Bhattacharya, B.: On intersecting a set of parallel line segments with a convex polygon of minimum area. *Inf. Process. Lett.* **105**, 58–64 (2008)
19. Mukhopadhyay, A., Greene, E., Rao, S.V.: On intersecting a set of isothetic line segments with a convex polygon of minimum area. *Int. J. Comput. Geom. Appl.* **19**(6), 557–577 (2009)
20. O’Rourke, J.: An on-line algorithm for fitting straight lines between data ranges. *Commun. ACM* **24**, 574–578 (1981)
21. Pellegrini, M.: Lower bounds on stabbing lines in 3-space. *Comput. Geom. Theory Appl.* **3**, 53–58 (1993)
22. Rappaport, D.: Minimum polygon transversals of line segments. *Int. J. Comput. Geom. Appl.* **5**(3), 243–256 (1995)