

# A 3-approximation algorithm for computing partitions with minimum stabbing number of rectilinear simple polygons

**Citation for published version (APA):**

Abam, M. A., Aronov, B., Berg, de, M. T., & Khosravi Dehkordi, A. (2011). A 3-approximation algorithm for computing partitions with minimum stabbing number of rectilinear simple polygons. In *Abstracts 27th European Workshop on Computational Geometry (EuroCG 2011, Morschach, Switzerland, March 28-30, 2011)* (pp. 103-106). ETH Zürich.

**Document status and date:**

Published: 01/01/2011

**Document Version:**

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

**Please check the document version of this publication:**

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

**General rights**

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

[www.tue.nl/taverne](http://www.tue.nl/taverne)

**Take down policy**

If you believe that this document breaches copyright please contact us at:

[openaccess@tue.nl](mailto:openaccess@tue.nl)

providing details and we will investigate your claim.

# A 3-Approximation Algorithm for Computing Partitions with Minimum Stabbing number of Rectilinear Simple Polygons

Mohammad Ali Abam\*    Boris Aronov†    Mark de Berg‡    Amirali Khosravi‡

## Abstract

Let  $P$  be a rectilinear simple polygon. The stabbing number of a partition of  $P$  into rectangles is the maximum number of rectangles stabbed by any axis-parallel segment inside  $P$ . We present a 3-approximation algorithm for finding a partition with minimum stabbing number. It is based on an algorithm that finds an optimal partition for histograms.

## 1 Introduction

Computing decompositions of simple polygons is one of the fundamental problems in computational geometry. When the polygon at hand is arbitrary then one typically wants a decomposition into triangles, and when the polygon is rectilinear one wants a decomposition into rectangles. Sometimes any such decomposition will do; then one can compute an arbitrary triangulation or, for rectilinear polygons, a vertical decomposition. In other cases one would like the decomposition to have certain properties. The property we are interested in is the so-called stabbing number—see below for a definition—of the decomposition.

Let  $P$  be a rectilinear simple polygon with  $n$  vertices. We call a decomposition of  $P$  into rectangles a *rectangular partition*. The stabbing number of a segment  $s$  with respect to a rectangular partition  $R$  is the number of rectangles intersected by  $s$ , and the (*rectilinear*) *stabbing number* of  $R$  is the maximum stabbing number of any axis-parallel segment  $s$  in the interior of  $P$ . De Berg and Van Kreveld [2] showed that any rectilinear polygon admits a rectangular partition with stabbing number  $O(\log n)$ . This bound is asymptotically tight in the worst case: any rectangular partition of a staircase polygon with  $n$  vertices has stabbing number  $\Omega(\log n)$ .

\*Department of Computer Science, TU Dortmund, 44221 Dortmund, Germany, mohammad-ali.abam@tu-dortmund.de

†Department of Computer Science and Engineering, Polytechnic Institute of NYU, Brooklyn, NY 11201-3840, USA, aronov@poly.edu. Work by Boris Aronov has been supported by grant No. 2006/194 from the U.S.-Israel Binational Science Foundation, by NSF Grant CCF-08-30691, and by NSA MSP Grant H98230-10-1-0210.

‡Department of Computing Science, TU Eindhoven, PO Box 513, 5600 MB Eindhoven, the Netherlands, {mdeberg, akhosrav}@win.tue.nl. Work by Amirali Khosravi has been supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 612.000.631.

The algorithm of De Berg and Van Kreveld [2] guarantees partitions which are tight in the worst case. However, some rectilinear polygons admit partitions with stabbing number  $O(1)$ . This leads to the topic of our paper: finding an *optimal rectangular* partition of a rectilinear polygon  $P$  whose stabbing number is minimum over all such partitions.

The problem is not known to be NP-complete. We present a 3-approximation algorithm for it. It is based on an algorithm that finds optimal rectangular partitions for histograms. Due to lack of space all the proofs are omitted in this version, and can be found in the full version of the paper.

**Related work.** Chazelle *et al.* [3] studied the stabbing number of convex decompositions of polytopes. Tóth showed that any partition of  $d$ -dimensional space ( $d \geq 2$ ) into  $n$  axis-aligned boxes has rectilinear stabbing number  $\Omega(\log^{1/(d-1)} n)$ , and he presented partitioning scheme achieving this bound [6]. Hershberger and Suri [4] gave an algorithm for triangulating a simple polygon with stabbing number  $O(\log n)$  which is worst-case tight. Considering triangulations of point sets, Agarwal, Aronov and Suri [1] proved that one can triangulate  $n$  points in  $\mathbb{R}^2$  and  $\mathbb{R}^3$  (using Steiner points) with stabbing number  $O(\sqrt{n} \cdot \log n)$ .

## 2 Optimal rectangular partitions

Let  $P$  be a rectilinear simple polygon with  $n$  vertices. We denote the interior of  $P$  by  $\text{int}(P)$  and its boundary by  $\partial P$ . In the remainder of paper, whenever we speak of partitions and stabbing numbers, we mean rectangular partitions and rectilinear stabbing numbers. We denote the stabbing number of a partition  $R$  by  $\sigma(R)$ . The *horizontal stabbing number* of  $R$ , denoted  $\sigma_{\text{hor}}(R)$ , is defined as the maximum stabbing number of any horizontal segment  $s \subset P$ . The *vertical stabbing number*, denoted  $\sigma_{\text{vert}}(R)$ , is defined similarly. Note that  $\sigma(R) = \max(\sigma_{\text{hor}}(R), \sigma_{\text{vert}}(R))$ .

We start by studying the properties of optimal partitions. Consider a partition  $R$  of  $P$ . The partition is induced by a set  $E(R)$  of *maximal edges*, that is, segments of maximal length that are the union of one or more rectangle edges that are not part of  $\partial P$ . A maximal edge is *anchored* if at least one of its endpoints is a vertex of  $P$ . We first show that there exists an optimal partition with only anchored edges.

**Lemma 1** Any rectilinear simple polygon  $P$  has an optimal partition  $R_{\text{opt}}$  in which all maximal edges are anchored.

A *binary space partition*, or BSP for short, of a rectilinear polygon  $P$  is a rectangular partitioning obtained by the following recursive process. First, the polygon is cut into two subpolygons with an axis-parallel segment contained in  $\text{int}(P)$ , and then the two subpolygons are partitioned recursively in the same way. A BSP is *anchored* if each of its cuts is anchored.

For so-called histograms we can show that there is always an optimal partition that is an anchored BSP. In fact, we show that any anchored partition of a histogram is a BSP. A (*vertical*) *histogram* is a rectilinear polygon  $H$  that has a horizontal edge seeing every point  $q \in \text{int}(H)$ . Here we say that an edge  $e$  sees a point  $q \in P$  if there is an axis-parallel segment  $s$  connecting  $e$  to  $q$  that is completely inside  $\text{int}(P)$  except possibly its endpoints. We call the horizontal edge that sees all points in the histogram the *base* of the histogram and denote it by  $\text{base}(H)$ .

**Lemma 2** Any anchored partition of a histogram is a BSP.

### 2.1 A 3-approximation algorithm

We present a 3-approximation algorithm for the problem of finding an optimal rectangular partition. First we split  $P$  into a set of histograms such that any axis-parallel segment inside  $P$  stabs at most three histograms. This can be done in  $O(n)$  time [5]. Then, we compute an optimal rectangular partition for each resulting histogram. By proving that this can be done in polynomial time we will have the following result.

**Theorem 3** Let  $P$  be a rectilinear simple polygon with  $n$  vertices. Then we can compute a rectangular partition of  $P$  with stabbing number at most  $3 \cdot \text{OPT}$  in polynomial time, where  $\text{OPT}$  is the minimum stabbing number of any rectangular partition of  $P$ .

Let  $H$  be a histogram. With a slight abuse of notation, we use  $n$  to denote the number of vertices of  $H$ . We assume without loss of generality that  $H$  is a vertical histogram lying above its base. By Lemmas 1 and 2,  $H$  admits an optimal partition that is an anchored BSP. We need the following properties.

**Lemma 4** There is an optimal partition  $R_{\text{opt}}$  for  $H$  that is an anchored BSP and such that for every rectangle  $r \in R_{\text{opt}}$  we have

- (i) the bottom edge of  $r$  is contained in either the top edge of a single rectangle  $r' \in R_{\text{opt}}$  or in  $\text{base}(H)$ , and
- (ii) the top edge of  $r$  contains an edge of  $H$ .

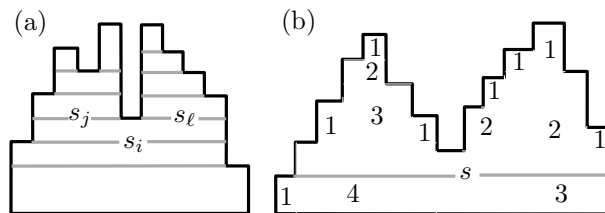


Figure 1: (a) Partitioning a histogram using canonical chords. (b) A partition with a unimodal labeling. The label sequence of the chord  $s$  is 4, 3 and the label sequence of the base is 1, 4, 3.

In the sequel all partitions have properties (i) and (ii) from Lemma 4 (but not all are anchored).

Our algorithm will do a binary search for the smallest value  $k$  such that  $H$  admits a partition with stabbing number  $k$ . Since there is always a partition with stabbing number at most  $2 \log_2 n$  [2], the binary search needs  $O(\log \log n)$  steps. It remains to describe our decision algorithm  $\text{HISTOGRAMPARTITION}(H, k)$ , which decides whether  $H$  has a partition with stabbing number at most  $k$ .

**Canonical chords.** A *chord* of  $H$  is a maximal horizontal segment contained in the interior of  $H$  except for its endpoints. A chord  $s$  partitions  $H$  into two parts. The part above  $s$  is a histogram, denoted by  $H(s)$ . Any partition  $R$  of  $H$  induces a partition of  $H(s)$ , denoted by  $R(s)$ . Now consider a partition of  $H$  obtained by adding a chord from each vertex of  $H$  for which this is possible. We call the resulting set of chords the *canonical chords* of  $H$  (see Figure 1(a)). We treat  $\text{base}(H)$  as a canonical chord.

The basic idea behind the algorithm is to treat the chords from top to bottom. Now consider a chord  $s_i$  with, say, two chords  $s_j$  and  $s_\ell$  immediately above it. Here we say that a chord  $s_i$  is *immediately above* another chord  $s_j$ , if we can connect  $s_i$  to  $s_j$  with a vertical segment that does not cross any other chord. One may hope that if we have optimal partitions for  $H(s_j)$  and  $H(s_\ell)$  then we can somehow “extend” these to an optimal partition for  $H(s_i)$ . Unfortunately this is not the case, since an optimal partition need not be composed of optimal subpartitions. The next idea is to compute all possible partitions for  $H(s_i)$ . These can be obtained by considering all combinations of a possible partition for  $H(s_j)$  and a possible partition for  $H(s_\ell)$ . Implementing this idea naively would lead to an exponential-time algorithm, however. Next we show how to compute a subset of all possible partitions that has only polynomial size and is still guaranteed to contain an optimal partition.

**Labeled partitions and label sequences.** We first introduce some notation and terminology. Let  $R$  be any partition of  $H$  (having the properties (i) and (ii) in Lemma 4). We say that a rectangle  $r \in R$  is *on*

top of a rectangle  $r' \in R$  if the bottom edge of  $r$  is contained in the top edge of  $r'$ . When the bottom edge of  $r$  is contained in  $\text{base}(H)$  then  $r$  is on top of the base. A *labeling* of  $R$  assigns a positive integer label  $\lambda(r)$  to each rectangle  $r \in R$ . We say that a labeled partition is *valid* (with respect to the stabbing number  $k$ ) if it satisfies these conditions:

- if  $r$  is on top of  $r'$  then  $\lambda(r) < \lambda(r')$ ;
- the vertical stabbing number of  $R$  equals the maximum label of any rectangle  $r \in R$ ;
- the stabbing number of  $R$  is at most  $k$ .

Observe that the first two conditions together imply that the stabbing number of  $R$  is equal to the maximum label assigned to any rectangle on top of  $\text{base}(H)$ . Also note that any partition with stabbing number  $k$  has a valid labeling: for example, one can set  $\lambda(r)$  to be equal to the maximum number of rectangles that can be stabbed by a vertical segment whose lower endpoint lies inside  $r$ . We will use the labelings to decide which partitions can be ignored and which ones we need to keep.

For a chord  $s$  of  $H$ , we define the *label sequence* of  $s$  with respect to a labeled partition  $R$  as the sequence of labels of the rectangles crossed by  $s$ , ordered from left to right; here we say that  $s$  crosses a rectangle  $r$  if  $s$  intersects  $\text{int}(r)$  or the bottom edge of  $r$ . We denote this sequence by  $\Sigma(s, R)$ ; see Figure 1(b). A label sequence is *valid* if it consists of at most  $k$  labels and the maximum label is at most  $k$ . Note that a labeled partition is valid if and only if the label sequence of each of its canonical chords is valid. A label sequence  $\lambda_1, \dots, \lambda_t$  is called *unimodal* if there is an index  $i$  such that  $\lambda_1 \leq \dots \leq \lambda_i$  and  $\lambda_i \geq \dots \geq \lambda_t$ . A labeling of a partition is unimodal if the label sequence of any chord is unimodal. A label sequence can be made unimodal using the following procedure.

MAKEUNIMODAL( $\Sigma$ )

Let  $\Sigma = \lambda_1, \dots, \lambda_t$ , and let  $\lambda_{i^*}$  be a maximum label in the sequence. For each  $i < i^*$  set  $\lambda_i := \max_{j \leq i} \lambda_j$ , and for each  $i > i^*$  set  $\lambda_i := \max_{j \geq i} \lambda_j$ .

The next lemma states that we can make the label sequences of all canonical chords unimodal, and still keep a valid sequence.

**Lemma 5** *Any anchored partition of  $H$  of stabbing number at most  $k$  admits a valid unimodal labeling.*

**Dominated and non-dominated sequences.** Next we explain how the labelings help us decide which partitions can safely be discarded. Consider an algorithm that handles the chords from top to bottom, and suppose that the algorithm reaches a chord  $s$ . Let  $R_1$  and  $R_2$  be two labeled partitions of  $H(s)$ . Suppose that  $\Sigma(s, R_1)$  is a subsequence of  $\Sigma(s, R_2)$ . Then there is no need to keep  $R_2$ : both partitions have stabbing

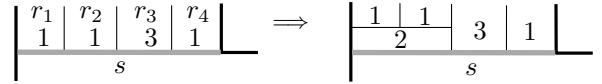


Figure 2: Merging two rectangles.

number at most  $k$  so far, and if we can complete  $R_2$  to a partition with stabbing number  $k$  of the full histogram  $H$  then we can do so with  $R_1$  as well. As another example in which we can disregard one of the two partitions, suppose that  $\Sigma(s, R_1) = 1, 1, 3, 1$  and  $\Sigma(s, R_2) = 1, 2, 3, 1$ , and let  $r_1, \dots, r_4$  be the four rectangle in  $R_1$  reaching the chord  $s$ . Then we could have merged  $r_1$  and  $r_2$  just before reaching  $s$ , that is, we could have terminated  $r_1$  and  $r_2$  and start a new rectangle with label “2”—see Figure 2. The new subsequence is then 2,3,1. This is a subsequence of  $\Sigma(s, R_2)$ , so we can disregard  $R_2$ .

To make this idea formal, we say that  $\Sigma(s, R_1)$  *dominates*  $\Sigma(s, R_2)$  if we can obtain a sequence  $\Sigma(s, R^*)$  that is a subsequence of  $\Sigma(s, R_2)$  by applying the following operation zero or more times to  $\Sigma(s, R_1)$ :

- Replace a subsequence  $\lambda_i, \dots, \lambda_j$  of  $\Sigma(s, R_1)$  by the single label “ $\max(\lambda_i, \dots, \lambda_j) + 1$ ”. Note that we can have  $i = j$ ; in this case the operation just adds 1 to the label  $\lambda_i$ .

Intuitively, if a label sequence dominates another label sequence, then the first sequence has postponed some merging operations that we can still do later on. Thus there is no need to maintain partitions with dominated label sequences. (Note that postponing a merging operation implies that the resulting partition may not be anchored. This means that in the algorithm presented below, we do not restrict ourselves to anchored partitions.) The next lemma gives a bound on the number of label sequences in the worst case.

**Lemma 6** *Let  $\mathcal{S}$  be any collection of valid unimodal sequences such that no sequence in  $\mathcal{S}$  dominates any other sequence in  $\mathcal{S}$ . Then  $|\mathcal{S}| = O(2^{3k/2}/\sqrt{k})$ .*

**The algorithm.** Our decision algorithm is as follows.

HISTOGRAMPARTITION( $H, k$ )

1. Compute the set of canonical chords of  $H$  and sort the chords by decreasing  $y$ -coordinate.
2. For each chord  $s$  in order, compute a collection  $\mathcal{R}(s)$  of labeled partitions of  $H(s)$ , as follows.
  - (i) If  $H(s)$  is a rectangle then set  $\mathcal{R}(s) := \{H(s)\}$ .
  - (ii) Otherwise  $s$  has one or more chords  $s_1, \dots, s_t$  immediately above it—see Figure 3. Find all valid unimodal partitions of  $H(s)$  that can be obtained from any combination of partitions in  $\mathcal{R}(s_1), \dots, \mathcal{R}(s_t)$  and whose label sequence is not dominated by the label sequence of any other such partition. (This will be explained below.)

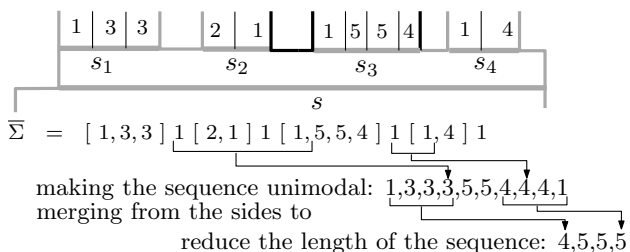


Figure 3: A chord  $s$ , the chords immediately above it, and the label sequence  $\bar{\Sigma}$  defined by the label sequences of the chords.

Let  $\mathcal{R}(s)$  be the set of all computed partitions. If  $\mathcal{R}(s)$  is empty, then report that no partition with stabbing number  $k$  exists for  $H$ , and exit.

3. Return any partition in  $\mathcal{R}(\text{base}(H))$ .

Next we explain how Step 2(ii) is performed. We assume that  $t > 1$ , that is, that  $s$  has several chords immediately above it; the case  $t = 1$  can be handled in a similar (but much simpler) way. In the sequel, we identify each partition with its label sequence and only talk about label sequences. Note that the operations we perform on the label sequences can be easily converted into the corresponding operations on the partitions. For every pair of label partitions  $R_1 \in \mathcal{R}(s_1), R_t \in \mathcal{R}(s_t)$  we proceed as follows.

- (a) For each  $1 < i < t$ , consider the set  $\mathcal{R}(s_i)$ . Note that the label sequences in  $\mathcal{R}(s_i)$  all have the same maximum value,  $M_i$ . This is because a label sequence dominates any sequence with larger maximum value. (The number of times the maximum label occurs can differ by at most one.) We pick an arbitrary label sequence  $\Sigma_i \in \mathcal{R}(s_i)$  for which  $M_i$  occurs the minimum number of times.
- (b) We now have, besides the partitions  $\Sigma_1 \in \mathcal{R}(s_1)$  and  $\Sigma_t \in \mathcal{R}(s_t)$ , picked a partition  $\Sigma_i$  from each  $\mathcal{R}(s_i)$  with  $1 < i < t$ . Let  $\bar{\Sigma}$  be the label sequence obtained by concatenating the sequences  $\Sigma_i$  in order, inserting a label “1” for any horizontal histogram edge incident to a chord  $s_i$ , see Figure 3. The labels “1” correspond to new rectangles whose top edge is the given histogram edge. We then make  $\bar{\Sigma}$  unimodal. This is done using a variant of the procedure MAKEUNIMODAL explained earlier: the difference is that if we give several consecutive labels the same value, then we merge them into a single new label—see Figure 3.
- (c) If the number of labels in  $\bar{\Sigma}$  is at most  $k$ , then we put  $\bar{\Sigma}$  into  $\mathcal{R}(s)$ . Otherwise  $\bar{\Sigma}$  is invalid because it contains too many labels, and we have to merge some rectangles. This is done as follows. Suppose that  $\bar{\Sigma}$  contains  $k + x$  labels  $\lambda_1, \dots, \lambda_{k+x}$ . Then we have to get rid of  $x$  labels by merging. Let  $x_{\text{left}}, x_{\text{right}}$  be integers such that  $x_{\text{left}} + x_{\text{right}} =$

$x + 2$  and both  $x_{\text{left}}, x_{\text{right}}$  are non-zero, or  $x_{\text{left}} + x_{\text{right}} = x + 1$  and one of  $x_{\text{left}}, x_{\text{right}}$  is zero. We merge  $x_{\text{left}}$  labels from the left into one new label, and  $x_{\text{right}}$  labels from the right into one, as in Figure 3. In other words, on the left side we replace  $\lambda_1, \dots, \lambda_{x_{\text{left}}}$  by a single new label  $\lambda_{x_{\text{left}}} + 1$  (and similarly on the right). If there are some labels immediately to the right of  $\lambda_{x_{\text{left}}}$  with the same value as  $\lambda_{x_{\text{left}}}$ , then we include them into the merging process. (We can do this for free, since it reduces the number of labels, without increasing the value of the new label.) If this merging process yields a new label whose value is more than the previous maximum label value, then we simply merge the entire sequence into a single new label. If the value of this label is  $k + 1$ , then we discard the sequence.

After applying the above steps to every pair  $R_1 \in \mathcal{R}(s_1), R_t \in \mathcal{R}(s_t)$ , we remove from  $\mathcal{R}(s)$  all partitions with a label sequence that is dominated by the sequence of some other partition. We do this by comparing every pair of partitions, and checking in  $O(k)$  time whether one dominates the other one. After handling  $s$ , and for any partition of  $H(s)$  denoted by  $R^*(s)$ , the following lemma shows that the set  $\mathcal{R}(s)$  contains at least a partition dominating  $\Sigma(s, R^*)$ .

**Lemma 7** HISTOGRAMPARTITION( $H, k$ ) returns a partition of  $H$  with stabbing number at most  $k$  if exists.

The next lemma follows from Lemma 6, and the fact that  $k \leq 2 \log_2 n$ .

**Lemma 8** HISTOGRAMPARTITION runs in polynomial time.

**References**

- [1] P.K. Agarwal, B. Aronov, and S. Suri. Stabbing triangulations by lines in 3D. *Proc. of 11th Symp. on Comp. Geom.* 267–276 (1995).
- [2] M. de Berg and M. van Kreveld. Rectilinear decompositions with low stabbing number. *Inf. Process. Lett.* 4:215–221 (1994).
- [3] B. Chazelle, H. Edelsbrunner, and L. J. Guibas, The complexity of cutting complexes. *Discr. Comput. Geom.* 6:139–181 (1989).
- [4] J. Hershberger and S. Suri. A pedestrian approach to ray shooting: shoot a ray, take a walk. *J. Alg.* 18:403-431 (1995).
- [5] C. Levcopoulos. *Heuristics for minimum decompositions of polygons*. Linköping Studies in Science and Technology, Dissertations, No. 55, 1987.
- [6] C.D. Tóth. Axis-aligned subdivisions with low stabbing numbers. *SIAM J. Discrete Math.* 22:1187–1204 (2008).