

# Approximating minimum-area rectangular and convex containers for packing convex polygons

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

de Berg, M. T., Alt, H., & Knauer, C. (2017). Approximating minimum-area rectangular and convex containers for packing convex polygons. *Journal of Computational Geometry*, 8(1), 1–10. <https://doi.org/10.20382/jocg.v8i1a1>

**DOI:**

[10.20382/jocg.v8i1a1](https://doi.org/10.20382/jocg.v8i1a1)

**Document status and date:**

Published: 01/01/2017

**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.

# APPROXIMATING MINIMUM-AREA RECTANGULAR AND CONVEX CONTAINERS FOR PACKING CONVEX POLYGONS\*

Helmut Alt,<sup>†</sup> Mark de Berg,<sup>‡</sup> Christian Knauer<sup>§</sup>

---

ABSTRACT. We investigate the problem of finding a minimum-area container for the disjoint packing of a set of convex polygons by translations. In particular, we consider axis-parallel rectangles or arbitrary convex sets as containers. For both optimization problems which are NP-hard we develop efficient constant factor approximation algorithms.

---

## 1 Introduction

Algorithms for efficiently packing objects into containers have important applications: In two dimensions, the problem occurs, e.g., in the context of cutting out a given set of patterns from a given large piece of material minimizing waste, typically in apparel fabrication or sheet metal processing. In three dimensions the problem occurs naturally in minimizing storage space or container space for transportation.

The problem has numerous variants. The most basic one is the decision problem whether a set of given objects can be packed into a given container. In sheet metal and apparel processing, mostly the problem of *strip packing* [12, 13, 5] occurs, i.e., a given set of objects needs to be packed inside a strip of a given fixed width using as short a piece of the strip as possible. Here, we will consider the problem of minimizing the area of the container.

Moreover, the shape of the objects to be packed is significant, e.g., in two dimensions arbitrary rectangles, axis-parallel rectangles, convex polygons, or simple polygons. Furthermore, the allowable transformations for placing the objects play an important role: Is it allowed to rotate them, i.e., apply *rigid motions*, or are only *translations* allowed? We will consider packing by translations, which is an important variant in practice. For example, in apparel production there are usually patterns of weaving or texture on the material so that the position where a piece should be cut out cannot be rotated arbitrarily.

Already the most simple versions of the problem are NP-hard, e.g., packing a given set of axis-parallel rectangles into a given axis-parallel container by translation, as can easily

---

\*Research was partially carried out at the International INRIA-McGill-Victoria Workshop on Problems in Computational Geometry, Barbados 2015. Research by Mark de Berg was also supported by the Netherlands' Organisation for Scientific Research (NWO) under project no. 024.002.003.

<sup>†</sup>*Institute of Computer Science, Freie Universität Berlin, Takustr. 9, 14195 Berlin, Germany, alt@mi.fu-berlin.de*

<sup>‡</sup>*Department of Computing Science, TU Eindhoven, P.O. Box 513, 5600 MB Eindhoven, the Netherlands, mdberg@win.tue.nl*

<sup>§</sup>*Universität Bayreuth, Institut für Angewandte Informatik, AG Algorithmen und Datenstrukturen (AI VI), 95440 Bayreuth, Germany, christian.knauer@uni-bayreuth.de*

be seen by reduction from PARTITION (see, e.g., [4]). So, only for a constant number of objects polynomial-time algorithms are known, see [1, 2, 14, 3, 8, 9, 10, 7]. Therefore, numerous approximation algorithms and heuristics have been investigated, mostly in the operations research and combinatorial optimization communities and mostly on axis-parallel rectangles under translation for a given container or strip. For a survey see [11].

Not much work has been done so far on approximation algorithms for the problem of finding *minimum area containers*, either convex ones or axis-parallel boxes. For objects that are axis-parallel, rectangles under translation, and arbitrary rectangles or convex polygons under rigid motion, approximation algorithms can be developed from the known ones for strip packing [13]. In fact, in [15] it is shown that packing axis-parallel rectangles under translation with an approximation factor of 2, and packing convex polygons under rigid motions with an approximation factor of 4 is possible efficiently.

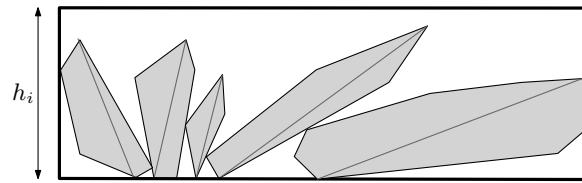
Thus the status of the packing problem in the translational case can be summarized as follows: All known algorithms that produce optimal results or have provable approximation ratios fall into one of two categories: They either pack only a very special kind of objects like line-segments or axis-parallel boxes, or they pack a constant number of polygons. Since the problem for a non-constant number of objects is NP-hard, this leaves the following question: can the packing problem for  $n$  objects that are not axis-parallel be efficiently approximated? We answer this question affirmatively for packing a set  $P$  of convex polygons into a minimum-area rectangular container. Using this result, we also show how to approximate the minimum-area convex container for  $P$ . Our algorithms run in  $O(n \log n)$  time.

We remark that the restriction to translational packing makes the problem harder. Indeed, if we allow rotations then we can compute a minimum-area oriented bounding box for each input polygon, rotate those boxes so that they become axis-parallel, and then pack the boxes using a known algorithm for axis-parallel boxes. Since the minimum-area oriented bounding box has area at most twice the area of the polygon itself, this can give a good approximation ratio. When we are not allowed to rotate the polygons this approach fails, since the area of the axis-parallel bounding box of a polygon can be arbitrarily much larger than the area of the polygon itself. Our result shows that we can still get a constant-factor approximation in the translational case. The approximation factors we obtain are fairly large, but to the best of our knowledge this is the first proof that these NP-hard optimization problems can be approximated at all.

## 2 Rectangular Containers

Let  $P := \{p_1, \dots, p_k\}$  be a set of convex polygons with  $n$  vertices in total. We call an axis-aligned box into which we can pack all polygons (without rotating them) a *container* for  $P$ . Our goal is to find a container for  $P$  of minimum area. Let  $b_{\text{opt}}$  be a minimum-area container for  $P$ , and let  $\text{OPT}$  be its area. Below we present an algorithm that finds a container of area at most  $17.45 \cdot \text{OPT}$ .

Define the *height* of a polygon  $p$ , denoted  $\text{height}(p)$ , to be the difference between its maximum and minimum  $y$ -coordinates, and let  $h_{\text{max}} := \max_{p \in P} \text{height}(p)$ . Furthermore,



**Fig. 1:** Example of a packing produced by our algorithm for a single height class.

define the *width* of a polygon  $p$ , denoted  $\text{width}(p)$ , to be the difference between its maximum and minimum  $x$ -coordinates, and let  $w_{\max} := \max_{p \in P} \text{width}(p)$ . We partition  $P$  into *height classes* using a parameter  $\alpha$ , with  $0 < \alpha < 1$  which is determined later to give the optimal approximation factor. More precisely,  $P$  is partitioned into subsets  $P_0, P_1, \dots$  according to the height: Polygons with height between  $h_{\max}$  and  $\alpha h_{\max}$  go into  $P_0$ , polygons with height between  $\alpha h_{\max}$  and  $\alpha^2 h_{\max}$  go into  $P_1$ , and so on. More precisely,  $P_i$  contains all polygons  $p \in P$  such that  $h_{i+1} < \text{height}(p) \leq h_i$ , where  $h_i = \alpha^i h_{\max}$ . Our general strategy is now as follows:

1. Pack each height class  $P_i$  separately into a container  $B_i$  of height  $h_i$ .
2. Replace each nonempty container  $B_i$  by a collection of axis-aligned *mini-containers* that are not too wide. Pack all mini-containers into a single container  $B$ .

Next we explain each of these steps in detail.

**Step 1: Packing polygons from one height class.** Consider the height class  $P_i$ , which contains all polygons whose height lies in the range  $(\alpha h_i, h_i]$ . Let  $\sigma := [0, \infty) \times [0, h_i]$  be a semi-infinite strip of height  $h_i$ . We place the polygons from  $P_i$  into  $\sigma$  in a greedy manner, as follows: For a polygon  $p$ , let  $s(p)$  be the segment connecting the lowest vertex of  $p$  to the highest vertex of  $p$ . If  $p$  has horizontal edges, then  $s(p)$  connects the bottom-left vertex to the top-right vertex. We call  $s(p)$  the *spine* of  $p$ . We sort the polygons in  $P_i$  according to the slopes of their spines and then we place them one by one into  $\sigma$ , where each polygon is pushed as far to the left as possible—that is, until it hits a previously placed polygon or the left edge of  $\sigma$ —while keeping its lowest vertex on the bottom edge of  $\sigma$ . Fig. 1 illustrates the process. After we have placed all polygons, we close the container  $B_i$ ; the right edge of  $B_i$  is defined by the vertical line through the rightmost vertex of any of the placed polygons.

**Lemma 2.1.** *The area of the container  $B_i$  computed for  $P_i$  satisfies*

$$\text{area}(B_i) \leq 2/\alpha \cdot \sum_{p \in P_i} \text{area}(p) + 2h_i \cdot \max_{p \in P_i} \text{width}(p).$$

*Proof.* Recall that we push each polygon (in order of the slope of the spines) to the left until either it hits the left edge of  $\sigma$  or until it hits a previously placed polygon. We define a polygon  $p$  to be *relevant* if

- (i) it is the last polygon that hits the left edge of  $\sigma$ , or

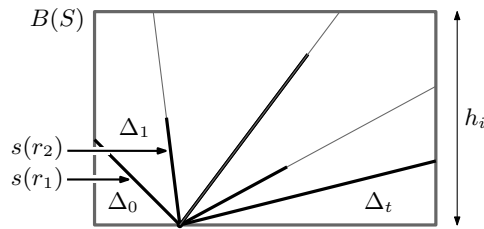


Fig. 2: Illustration for the proof of Lemma 2.1.

- (ii) there is a previously placed relevant polygon  $r'$  such that  $p$  is the last polygon that hits  $r'$ .

By definition, there is a relevant polygon touching the left edge of  $B_i$ . There must also be a relevant polygon touching the right edge of  $B_i$ . Indeed, assume otherwise and let  $p$  be the relevant polygon placed last. Then either the right edge of  $B_i$  is determined by the rightmost vertex of some irrelevant polygon  $p'$  placed before  $p$  or there are polygons placed after  $p$ . Let us consider the first case: Since  $p'$  reaches the right side of  $B_i$  and all polygons are placed on the base of  $B_i$ ,  $p'$  must be *taller* than all polygons placed after it. Consequently  $p$  (which is one of those polygons placed after  $p'$ ) cannot be connected to the left side of  $B_i$  via a chain of relevant polygons, a contradiction. In the second case, none of the polygons placed after  $p$  can touch  $p$ , because otherwise one of them would become relevant, as well. Let  $p'$  be the one placed directly after  $p$ . On its left,  $p'$  must touch some (irrelevant) polygon  $p''$  placed before  $p$ . But then, the union of  $p'$  and  $p''$  would separate  $p$  from the left edge of  $B_i$ . Hence,  $p$  cannot be relevant, a contradiction.

We conclude that in our analysis we can safely restrict our attention to the relevant polygons. Now let  $P_i^*$  be the set of relevant polygons in  $P_i$ . We will prove that

$$\text{area}(B_i) \leq 2/\alpha \cdot \sum_{p \in P_i^*} \text{area}(p) + 2h_i \cdot \max_{p \in P_i^*} \text{width}(p), \quad (1)$$

which obviously implies the lemma.

Let  $p_1, p_2, \dots, p_t$  be the polygons in  $P_i^*$ , ordered according to the slope of their spines, and let  $S := \{s(p_1), \dots, s(p_t)\}$  be the set of spines of the relevant polygons. Imagine placing each spine with its lower endpoint in the origin. Let  $B(S)$  be the box of height  $h_i$  and minimum width containing all the spines placed in this manner, see Figure 2. Note that

$$\text{area}(B(S)) \leq 2h_i \cdot \max_{p \in P_i^*} \text{width}(p). \quad (2)$$

Partition  $B(S)$  into pieces by extending the spines until they hit the boundary of  $B(S)$ . Let  $\Delta_0, \Delta_1, \dots, \Delta_t$  be the resulting set of pieces, where the numbering is in clockwise order around the origin.

To bound the area of  $B_i$ , we first partition  $B_i$  into regions (which are either triangles, quadrilaterals, or 5-gons) by extending the spines of the polygons in  $P_i^*$  until they hit the boundary of  $B_i$ . Let  $Q_0, Q_1, \dots, Q_t$  be these regions ordered from left to right. Thus  $Q_j$  lies

between spines  $s(p_j)$  and  $s(p_{j+1})$ , except for the first and last region which lie before the first spine and after the last spine, respectively. We claim (and will prove below) that

$$\text{area}(Q_j) \leq 2/\alpha \cdot \left( \text{area}(p_j^r) + \text{area}(p_{j+1}^l) \right) + \text{area}(\Delta_j), \tag{3}$$

where  $p^l$  and  $p^r$  are the right and left hand side, respectively, into which a polygon  $p$  is split by its spine, and, with a slight abuse of notation (since  $p_0$  and  $p_{t+1}$  do not exist) we define  $\text{area}(p_0) = \text{area}(p_{t+1}) = 0$ . From the claim we derive

$$\begin{aligned} \text{area}(B_i) &= \sum_j \text{area}(Q_j) \\ &\leq \sum_j \left( 2/\alpha \cdot \left( \text{area}(p_j^r) + \text{area}(p_{j+1}^l) \right) + \text{area}(\Delta_j) \right) \\ &\leq 2/\alpha \cdot \sum_{p \in P_i^*} \text{area}(p) + \text{area}(B(S)), \end{aligned} \tag{4}$$

which, using (2), proves the lemma.

It remains to prove claim (3). To simplify the presentation we assume  $0 < j < t$ ; it is easily verified that a similar argument applies when  $j = 0$  and when  $j = t$ . To get a bound on the area of  $Q_j$ , let  $s_p$  be the segment parallel to  $s(p_j)$  that splits  $Q_j$  and passes through the point  $p$  where  $p_j$  touches  $p_{j+1}$ . Let  $q$  be the lower endpoint of  $s_p$ . There are two cases, as illustrated in Fig. 3:

*Case A: The point  $q$  lies to the left of the lower endpoint of  $s(p_{j+1})$ .* Now connect  $q$  to the opposite boundary of  $B_i$  with a segment,  $s_q$ , which is parallel to  $s(p_{j+1})$ . The segments  $s_p$  and  $s_q$  partition  $Q_j$  into three pieces  $Q_j^1$ ,  $Q_j^2$ , and  $Q_j^3$ . First consider the left piece,  $Q_j^1$ , which is bounded from the left by  $s(p_j)$  and from the right by  $s_p$ . Let  $A$  be the parallelogram between the lines spanned by  $s(p_j)$ ,  $s_p$ , and the upper and lower edges of  $B_i$ . Note that  $Q_j^1$  can be smaller than  $A$ , because unlike  $A$  it can be bounded by the right or left edge of  $B_i$ . Because  $\text{height}(p_j) \geq \alpha h_i$ , the area of the triangle  $T$  defined by  $s(p_j)$  and  $p$  is at least  $\text{area}(A) \cdot \alpha/2$  which is at least  $\text{area}(Q_j^1) \cdot \alpha/2$ . Since also the area of  $T$  is at most  $\text{area}(p_j^r)$ , we have

$$\text{area}(Q_j^1) \leq 2/\alpha \cdot \text{area}(p_j^r). \tag{5}$$

A similar argument shows that the area of  $Q_j^3$ , the rightmost piece, is at most  $2/\alpha \cdot \text{area}(p_{j+1}^l)$ . The middle piece,  $Q_j^2$ , is bounded by  $s(p_j)$  and  $s(p_{j+1})$ , so  $\text{area}(Q_j^2) \leq \text{area}(\Delta_j)$ . This finishes the claim for Case A.

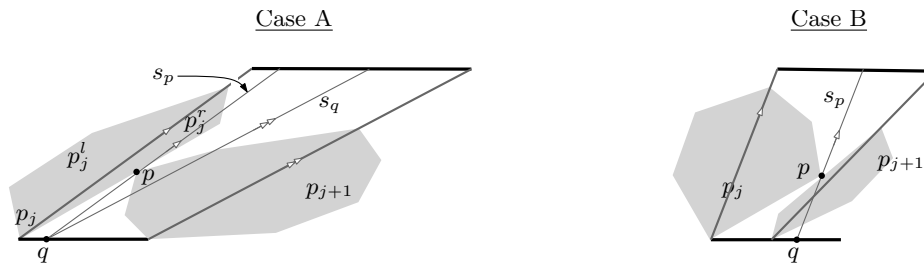


Fig. 3: The two cases in the proof of Lemma 2.1.

*Case B:* The point  $q$  lies to the right of or coincides with the lower endpoint of  $s(p_{j+1})$ . As before we can bound the area between  $s(p_j)$  and  $s_p$  by  $2/\alpha \cdot \text{area}(p_j^r)$ . Note that this area may not be completely contained in  $Q_j$ , but this does not matter. The only part of  $Q_j$  we have not accounted for is the part between  $s_p$  and the line through  $s(p_{j+1})$  above the intersection point of  $s_p$  and  $s(p_{j+1})$ . The area of this part is bounded by the area of  $\Delta_j$ . This finishes the proof of claim (3) for Case B and, hence, finishes the proof of the lemma.  $\square$

**Step 2: Generating and packing mini-containers.** Step 1 results in a collection of containers  $B_i$  of various lengths  $l_i$ , each containing all polygons from the height class  $P_i$ . We replace each  $B_i$  by mini-containers of equal lengths as follows. Recall that  $w_{\max}$  is the maximum width of any polygon in  $P$ . First, partition  $B_i$  into boxes of width  $cw_{\max}$ —we will determine a suitable value for  $c$  later—and height  $h_i$ , except for the last box which may have width smaller than  $w_{\max}$ . Now assign each polygon  $p \in P_i$  to the box  $b$  containing its leftmost point. (If the leftmost point lies on the boundary between two boxes we assign it to the righthand box.) We now generate a mini-container from each box  $b$  by extending  $b$  to the right until its width is exactly  $(c+1)w_{\max}$ . Note that (the extended)  $b$  contains all polygons assigned to  $b$ . This results in a collection  $\overline{R}_i$  of at most  $l_i/(cw_{\max}) + 1$  mini-containers each having width exactly  $(c+1)w_{\max}$ . Since the height of  $B_i$  and of each mini-container is  $h_i$ , we have

$$\sum_{b \in \overline{R}_i} \text{area}(b) \leq \left(1 + \frac{1}{c}\right) \cdot \text{area}(B_i) + (c+1)w_{\max}h_i. \quad (6)$$

Let  $\overline{R} := \bigcup \overline{R}_i$  be the collection of all mini-containers obtained in this manner. Packing these mini-containers can trivially be done without any loss of area: Since all mini-containers in  $\overline{R}$  have the same width we can simply stack them on top of each other to obtain our final container  $B$ .

We can now state our result about packing polygons.

**Theorem 2.2.** *Let  $P$  be a set of polygons in the plane with  $n$  vertices in total. We can pack  $P$  in  $O(n \log n)$  time into an axis-aligned rectangular container  $B$  such that  $\text{area}(B) \leq 17.45 \cdot \text{OPT}$ , where  $\text{OPT}$  is the minimum area of any axis-aligned rectangular container for  $P$ .*

*Proof.* Let  $B$  be the container computed by our algorithm. Observe that  $\text{OPT} \geq$

$\sum_{p \in P} \text{area}(p)$  and  $\text{OPT} \geq w_{\max} h_{\max}$ . We have

$$\begin{aligned}
 \text{area}(B) &= \sum_{b \in \bar{R}} \text{area}(b) \\
 &\leq \sum_i \left\{ \left(1 + \frac{1}{c}\right) \cdot \text{area}(B_i) + (c+1)w_{\max}h_i \right\} && \text{by (6)} \\
 &\leq \left(1 + \frac{1}{c}\right) \sum_i \left\{ 2/\alpha \cdot \sum_{p \in P_i} \text{area}(p) + 2h_i \cdot \max_{p \in P_i} \text{width}(p) \right\} \\
 &\quad + \frac{1}{1-\alpha} \cdot (c+1)w_{\max}h_{\max} \\
 &\hspace{15em} \text{(by Lemma 2.1 and because } h_i = \alpha^i h_{\max} \text{)} && (7) \\
 &\leq \left(1 + \frac{1}{c}\right) \cdot \left(2/\alpha \cdot \sum_{p \in P} \text{area}(p) + 2/(1-\alpha) \cdot w_{\max}h_{\max}\right) \\
 &\quad + \frac{1}{1-\alpha}(c+1) \cdot \text{OPT} \\
 &\leq \left( \left(1 + \frac{1}{c}\right) \left(\frac{2}{\alpha} + \frac{2}{1-\alpha}\right) + \frac{c+1}{1-\alpha} \right) \cdot \text{OPT}.
 \end{aligned}$$

The factor in front of OPT simplifies to

$$f(c, \alpha) := \left(1 + \frac{1}{c}\right) \cdot \frac{2 + c\alpha}{\alpha - \alpha^2} \quad (8)$$

In order to minimize the approximation factor, we determine the optimal values for  $c$  and  $\alpha$  by setting the partial derivatives to zero:  $\frac{\partial f}{\partial c} = 0$  and  $\frac{\partial f}{\partial \alpha} = 0$ .

The first equation yields the identity  $\alpha = 2/c^2$ . Using this in the second equation gives that  $c$  is obtained by  $c^3 - 4c - 2 = 0$  which has the solution  $c = 2.214\dots$  which gives  $\alpha = 0.407\dots$  and the approximation factor  $f(c, \alpha) = 17.449\dots$

In order to get the desired runtime, we first observe that partitioning the polygons in  $P$  into height classes takes linear time and sorting the ones in each class by the slopes of their spines takes a total time of  $O(n \log n)$ .

In order to pack each polygon  $p$  efficiently in Step 1, we maintain a balanced binary search tree  $T$ . It contains, ordered by  $y$ -coordinate, the set  $E$  of vertices of the set  $P'$  of polygons already packed which are visible from the right (see the dashed lines in Figure 4). In order to find the point where  $p$  first hits a polygon of  $P'$  when moved to the left, we search in  $T$  for the  $y$ -coordinates of the set  $L$  of vertices of  $p$  visible from the left, and thereby identify the corresponding candidate edges in  $P'$  (see the dotted arrows pointing to the left in Figure 4). This requires  $O(\log n)$  time per vertex in  $L$ . Vice versa, for the set  $B$  of vertices of  $P'$  in the  $y$ -range of  $p$  we find the corresponding candidate edges of  $p$  (see the dashed arrows pointing to the right in Figure 4). This can be done in  $O(\log n)$  time per vertex of  $B$ , assuming that the vertices of  $p$  are available sorted clockwise in an array. For each vertex in  $B \cup L$  we now compute the horizontal distance to its candidate edge. The vertex-edge pair with the smallest distance determines where  $p$  first hits a polygon of  $P'$  when moved to the left. After  $p$  has been packed, no vertex in  $B$  is visible from the right anymore, and the set  $R$  of vertices of  $p$  visible from the right has now become a part of  $E$ . Consequently, the data structure  $T$  has to be updated:  $B$  has to be removed from, and  $R$  has to be inserted into  $T$ . These updates require  $O(\log n)$  time per vertex. Obviously, each vertex is inserted into and deleted from the data structure at most once, taking  $O(n \log n)$  time in total.  $\square$



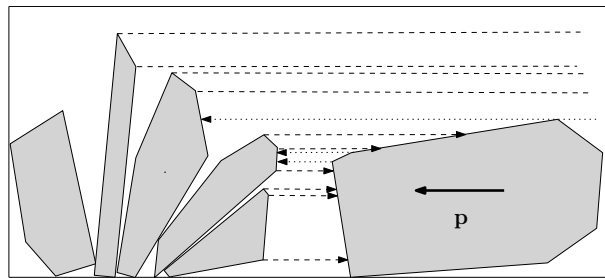


Fig. 4: The data structure for inserting polygons.

### 3 Convex Containers

Next, we consider the problem of finding an arbitrary convex container of minimal area for a set of convex polygons. In other words, we want to pack the input polygons by translation such that the area of the convex hull of their union is minimized. We will give an approximation algorithm based on the one for minimum enclosing boxes from before.

The idea is to find a suitable orientation  $\phi^* \in S^1$ , determine the approximately optimal bounding box  $B$  with that orientation using the algorithm from Section 2 and then return  $B$  as the approximate solution.

Given a set  $P$  of polygons, we choose  $\phi^*$  to be that orientation minimizing  $h_{max}(\phi)w_{max}(\phi)$ , where  $h_{max}(\phi)$  is the maximal extent of any polygon in direction perpendicular to  $\phi$  and  $w_{max}(\phi)$  the extent in direction  $\phi$ . The orientation  $\phi^*$  can be determined in  $O(n \log n)$  time by using rotating calipers simultaneously around all the polygons from  $P$ .

To see this, observe that the functions  $h_{max}(\phi)$  and  $w_{max}(\phi)$  are composed piecewise of functions of the form  $a \sin(\phi + b)$  for some constants  $a, b \in \mathbb{R}$ . The number of these functions equals the number of pairs of antipodal vertices in all polygons, so it is  $O(n)$ . Two of these functions can intersect at most once, so by an algorithm of Hershberger [6] their upper envelope can be constructed in  $O(n \log n)$  time and consists of  $O(n\alpha(n))$  pieces of the functions above ( $\alpha(n)$  being the inverse Ackermann function). The upper envelope of  $h_{max}(\phi)w_{max}(\phi)$  is a piecewise trigonometric function as well, where the pieces result from merging the pieces of the upper envelopes of  $h_{max}(\phi)$  and  $w_{max}(\phi)$ . Therefore, it also consists of  $O(n\alpha(n))$  pieces and it can be constructed and its minimum determined in  $O(n \log n)$  time.

To get an estimate on the quality of the solution, let us consider the optimal solution which is some convex polygon  $C_{opt}$ . Consider a bounding box  $B_{opt}$  of  $C_{opt}$  that has a side parallel to a longest line segment  $\overline{pq}$  contained in  $C_{opt}$ . We claim that

$$\text{area}(B_{opt}) \leq 2 \cdot \text{area}(C_{opt}). \quad (9)$$

To see this, observe that  $B_{opt}$  is partitioned into two rectangles  $B_1$  and  $B_2$  (one of which can be empty) by  $\overline{pq}$ . The triangle in  $B_1$  formed by  $\overline{pq}$  and the opposite tangent point

has half the area of  $B_1$ . The same holds for the corresponding triangle in  $B_2$ . Since the union of both triangles is contained in  $C_{opt}$ , claim (9) follows.

Now, let  $h_{opt}$  and  $w_{opt}$  be the height and width of  $B_{opt}$ , respectively. Then by the choice of  $\phi^*$ ,

$$h_{max}(\phi^*)w_{max}(\phi^*) \leq h_{opt}w_{opt} = \text{area}(B_{opt}) \leq 2 \cdot \text{OPT}. \quad (10)$$

Observe that now OPT denotes the area of the smallest enclosing convex container rather than bounding box. Therefore, using (10) the derivation in (7) can be replaced by

$$\begin{aligned} \text{area}(B) &= \sum_{b \in \overline{R}} \text{area}(b) \\ &\leq \sum_i \left\{ \left(1 + \frac{1}{c}\right) \cdot \text{area}(B_i) + (c+1)w_{\max}h_i \right\} \\ &\leq \left(1 + \frac{1}{c}\right) \sum_i \left\{ 2/\alpha \cdot \sum_{p \in P_i} \text{area}(p) + 2h_i \cdot \max_{p \in P_i} \text{width}(p) \right\} \\ &\quad + 1/(1-\alpha) \cdot (c+1)w_{\max}h_{\max} \\ &\leq \left(1 + \frac{1}{c}\right) \cdot \left( 2/\alpha \cdot \sum_{p \in P} \text{area}(p) + 2/(1-\alpha) \cdot w_{\max}h_{\max} \right) \\ &\quad + 2/(1-\alpha) \cdot (c+1) \cdot \text{OPT} \\ &\leq \frac{c+1}{c} \left( \frac{2}{\alpha} + \frac{4}{1-\alpha} + \frac{2c}{1-\alpha} \right) \cdot \text{OPT}. \end{aligned} \quad (11)$$

Let  $f(c, \alpha)$  denote the factor in front of OPT in the last line. The partial derivatives are:

$$\begin{aligned} \frac{\partial f}{\partial c} &= \frac{-1}{c^2} \cdot \left( \frac{2}{\alpha} + \frac{4+2c}{1-\alpha} \right) + \frac{c+1}{c} \cdot \frac{2}{1-\alpha} \\ \frac{\partial f}{\partial \alpha} &= \frac{c+1}{c} \cdot \left( -\frac{2}{\alpha^2} + \frac{4+2c}{(1-\alpha)^2} \right) \end{aligned}$$

As can be easily verified, both expressions evaluate to zero for  $c = 2$  and  $\alpha = 1/3$ , so we obtain the optimal approximation factor  $f(2, 1/3) = 27$ , if we choose these values in our algorithm. We obtain:

**Theorem 3.1.** *Let  $P$  be a set of convex polygons in the plane with  $n$  vertices in total. We can pack  $P$  in  $O(n \log n)$  time into a convex polygon  $B$  such that  $\text{area}(B) \leq 27 \cdot \text{OPT}$ , where OPT is the minimum area of any convex container for  $P$ .*

## Acknowledgement

We would like to thank an anonymous referee for a useful hint by which we could improve the approximation factors.

## References

- [1] H. Ahn, H. Alt, S. W. Bae, and D. Park. Bundling three convex polygons to minimize area or perimeter. In *Algorithms and Data Structures - 13th International Symposium, WADS 2013, London, ON, Canada, August 12-14, 2013. Proceedings*, pages 13–24, 2013.

- [2] H.-K. Ahn and O. Cheong. Aligning two convex figures to minimize area or perimeter. *Algorithmica*, 62(1-2):464–479, 2012.
- [3] H. Alt and F. Hurtado. Packing convex polygons into rectangular boxes. In *JCDCG*, pages 67–80, 2000.
- [4] R. J. Fowler, M. Paterson, and S. L. Tanimoto. Optimal packing and covering in the plane are np-complete. *Inf. Process. Lett.*, 12(3):133–137, 1981.
- [5] R. Harren, K. Jansen, L. Prädél, and R. van Stee. A  $(5/3 + \epsilon)$ -approximation for strip packing. *Comput. Geom.*, 47(2):248–267, 2014.
- [6] J. Hershberger. Finding the upper envelope of  $n$  line segments in  $o(n \log n)$  time. *Inf. Process. Lett.*, 33(4):169–174, 1989.
- [7] H.-C. Lee and T. C. Woo. Determining in linear time the minimum area convex hull of two polygons. *IIE Transactions*, 20(4):338–345, 1988.
- [8] V. Milenkovic. Translational polygon containment and minimal enclosure using linear programming based restriction. In *STOC*, pages 109–118, 1996.
- [9] V. Milenkovic. Multiple translational containment. part ii: Exact algorithms. *Algorithmica*, 19(1/2):183–218, 1997.
- [10] V. Milenkovic. Rotational polygon containment and minimum enclosure. In *Symposium on Computational Geometry*, pages 1–8, 1998.
- [11] G. Scheithauer. *Zuschnitt- und Packungsoptimierung: Problemstellungen, Modellierungstechniken, Lösungsmethoden*. Studienbücher Wirtschaftsmathematik. Vieweg+Teubner Verlag, 2008.
- [12] I. Schiermeyer. Reverse-fit: A 2-optimal algorithm for packing rectangles. In *Algorithms - ESA '94, Second Annual European Symposium, Utrecht, The Netherlands, September 26-28, 1994, Proceedings*, pages 290–299, 1994.
- [13] A. Steinberg. A strip-packing algorithm with absolute performance bound 2. *SIAM J. Comput.*, 26(2):401–409, Apr. 1997.
- [14] K. Tang, C. C. L. Wang, and D. Z. Chen. Minimum area convex packing of two convex polygons. *Int. J. Comput. Geometry Appl.*, 16(1):41–74, 2006.
- [15] L. von Niederhäusern. Packing polygons. Master’s thesis, EPFL Lausanne, FU Berlin, 2014.