Cutting out polygon collections with a saw[☆]

Adrian Dumitrescu[a], Anirban Ghosh[a], Masud Hasan[b]

[a]*Department of Computer Science, University of Wisconsin–Milwaukee, USA*
[b]*Department of Computer Science, Taibah University, Madina Munawwara, Saudi Arabia*

Abstract

(I) Given a segment-cuttable polygon P drawn on a planar piece of material Q, we show how to cut P out of Q by a (short) segment saw with a total length of the cuts no more than 2.5 times the optimal. We revise the algorithm of Demaine et al. (2001) so as to achieve this ratio.

(II) We prove that any collection $\mathcal{R}$ of n disjoint[1] axis-parallel rectangles drawn on a planar piece of material Q is cuttable by at most $4n$ rays and present an algorithm that runs in $O(n \log n)$ time for computing a suitable cutting sequence. In particular, the same result holds for cutting with an arbitrary segment saw (of any length).

(III) Given a collection $\mathcal{P}$ of segment-cuttable polygons drawn on a planar piece of material such that no two polygons in $\mathcal{P}$ touch each other, $\mathcal{P}$ is always cuttable by a sufficiently short segment saw. We also show that there exist collections of disjoint polygons that are uncuttable by a segment saw.

(IV) Given a collection $\mathcal{P}$ of disjoint polygons drawn on a planar piece of material Q, we present a polynomial-time algorithm that computes a suitable cutting sequence to cut the polygons in $\mathcal{P}$ out of Q using ray cuts when $\mathcal{P}$ is ray-cuttable and otherwise reports $\mathcal{P}$ as uncuttable.

Keywords: cuttable polygon, cuttable collection, separability, line cut, ray cut, segment cut, cutting sequence

1. Introduction

The problem of efficiently cutting out a simple polygon P drawn on a planar piece of material (such as wood, paper, glass) Q, was introduced by Overmars and Welzl in their seminal paper [20] from 1985. Since then, the problem has attracted the interest of many computational geometers.

A saw cut may split (divide) Q into a number of pieces—those that lie left of the cut and those that lie right of the cut. In some situations, the saw may stop short of splitting Q, in which case, the material remains as one solid piece. In any case we do not allow a cut to run through the interior of P. Several variants have been studied, primarily depending on the cutting tools used [1, 3, 5, 7, 10–14, 20, 23]: line cuts, ray cuts and segment cuts; they are described subsequently. In saw cutting, i.e., in all three models above, turns are impossible. The type of tool used in cutting determines the class of polygons that can be cut within that model.

[1]For brevity, a *collection of disjoint polygons* refers to a collection of polygons with pairwise disjoint interiors.

[☆]A preliminary version of this paper appeared in the *Proceedings of the International Conference on Algorithms and Discrete Applied Mathematics*, (CALDAM 2015), IIT Kanpur, India, February 2015, Vol. 8959 of LNCS, pp. 58–68.

Email addresses: `dumitres@uwm.edu` (Adrian Dumitrescu), `anirban@uwm.edu` (Anirban Ghosh), `mhares@taibahu.edu.sa` (Masud Hasan)

The measures of efficiency commonly considered in polygon cutting are the total length of the cuts and the total number of cuts. Polygon cutting problems are useful in industry applications such as metal sheet cutting, paper cutting, furniture manufacturing and numerous other areas of engineering, where smart cutting techniques with high efficiency may result in the reduction of production costs. For instance, reducing the total length of the cuts may result in lesser power requirement and extend the life of the cutting tool. Similarly, reducing the total number of cuts may save cutting time and extend the life of the cutting tool.

1.1. Line cuts, ray cuts, and segment cuts

A *line cut* (also called a *guillotine cut*) is a line that does not go through P and divides the current piece of material containing P (initially Q) into two pieces. For cutting P out of Q by line cuts, P must be convex. An instance of a cutting sequence using line cuts appears in Figure 1. The most studied efficiency measure for line cutting is the total length of the cuts and several approximation algorithms have been obtained [1, 3, 5, 7, 10–13, 20, 23], including a PTAS proposed by Bereg et al. [3].

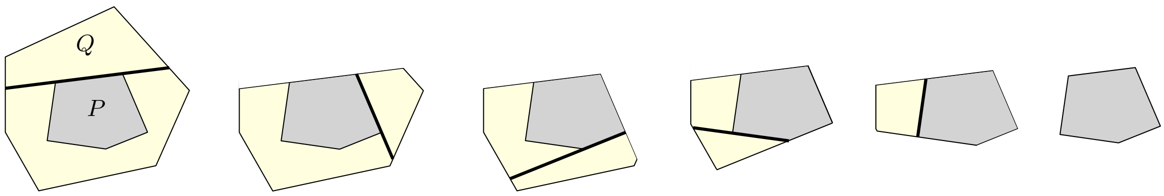

Figure 1: Cutting a convex polygon P out of Q using 5 line cuts.

A *ray cut* comes from infinity and can stop at any point outside P, again, not necessarily splitting the piece of material into pieces. Ray cuts are usually used to cut out non-convex polygons; however, not all non-convex polygons can be cut by ray cuts. The following observation gives a necessary and sufficient condition for ray-cuttability, in the case of a single polygon; see Figure 2. Several approximation algorithms can be found in [7, 10, 11, 23].

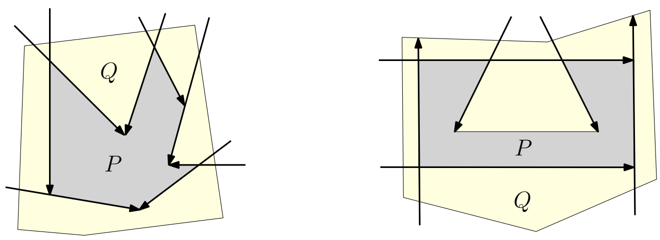

Figure 2: Left: a ray-cuttable polygon. Right: a polygon which is not ray-cuttable or segment-cuttable.

Observation 1. *A polygon P drawn on a planar material is ray-cuttable if and only if every edge of P that has some material adjacent to it, can be extended to infinity from one of its endpoints without passing through the interior of P.*

A *segment cut* is similar to a ray cut, but is not required to start at infinity, it may start at some finite point. The *segment saw* (also referred to as *circular saw* in [11]) is abstracted as a line segment, which cuts through material when moved along its supporting line. Before executing a segment cut, the saw needs to be placed. A small example of a cutting sequence appears in Figure 3. Recall that saw turns are impossible during a cut; however, if a small free space within Q

is available, a segment cut can be initiated there by maneuvering (i.e., rotating) the saw. The space required for maneuvering the saw is proportional to the length of the saw. The problem of cutting a polygon by a segment saw was introduced by Demaine et al. [11] in 2001. The authors gave a characterization of the class of polygons cuttable by a (possibly short) segment saw: a polygon is cuttable in this model, i.e., by a sufficiently short segment saw, if and only if it does not have two adjacent reflex vertices (with interior angle $> \pi$).

Note that ray-cuttability is not equivalent with segment-cuttability. For instance the polygon P in Figure 3 is segment-cuttable but not ray-cuttable; indeed, the condition specified by Observation 1 is not fulfilled by two edges of P at the bottom of the pocket.

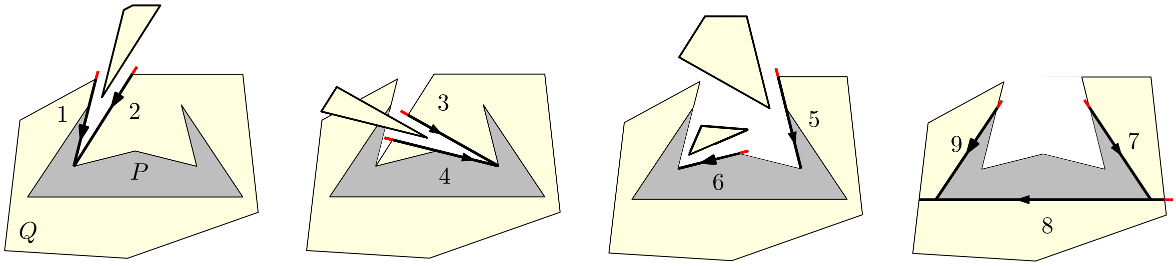

Figure 3: A cutting sequence with a segment saw (in red) consisting of 9 cuts. The polygon is not ray-cuttable.

For ease of analysis the length of the saw is assumed to be arbitrarily small, i.e., the segment abstracting the saw is as short as needed. Consequently, a segment cut can be initiated from an arbitrarily small available free space. In this model, several parts may result after a cut is made, and any of them can be removed (lifted) from the original plane. Moreover, free space may appear within the pieces of material from where future segment cuts can be initiated. The cutting process may continue independently on any of the separated pieces of material, if resulting parts contain subcollections of a larger collection to be cut out.

1.2. Our results and related work

Demaine et al. [11] presented an algorithm for cutting P out of its convex hull using segment cuts with a total number of cuts and total length of the cuts within constant factors of the respective optima. With regard to the total number of cuts, this number is within 2.5 times the respective optimum. Dumitrescu and Hasan [14] improved the approximation ratio on the total number of cuts from 2.5 to 2. Moreover, the new approximation guarantee is in a stronger sense than that offered by Demaine et al. [11]. While theirs achieves ratio 2.5 for cutting out P from its convex hull, the algorithm in [14] achieves ratio 2 for cutting out P from any enclosing polygon Q.

With regard to the total length measure of the cuts, Demaine et al. [11] reported that their algorithm achieves the same ratio, 2.5, as for the total number of cuts. Here we point out that the approximation factor on the total length of the cuts for their algorithm (as implied by their proof) is in fact 3. In Section 2 we show that with a little care, one can recover the claimed ratio 2.5; it requires a small change in their algorithm to obtain:

Theorem 1. *Given a segment-cuttable polygon P, drawn on a planar piece of material Q, P can be cut out of Q by an arbitrarily short segment saw with the total length of the cuts within 2.5 times the optimum.*

We now proceed to our main results regarding collections of polygons. In the conclusion of their paper, Demaine et al. [11] left several open questions. Here we focus on one of them: "What collections of nonoverlapping polygons in the plane can be simultaneously cut out by a circular saw?"[2] The authors remarked that the problem is nontrivial when some of the polygons share edges. We study the case of axis-parallel rectangles in Section 3, where it is shown that the answer is always positive. Moreover, Section 4 gives evidence that the problem is nontrivial even if the polygons do not share edges.

A collection $\mathcal{P}$ of disjoint polygons drawn on a planar piece of material is *segment-cuttable* (or cuttable by segments) if there exists a sequence of segment cuts after which every polygon in the collection is cut out (along its sides) as a separate piece. Otherwise, we say that the collection is *uncuttable* by segments. Cuttability with other tools (such as arbitrarily short segment saw, rays or lines) are defined similarly. The following observations are in order.

1. If a collection of polygons is cuttable by rays, then it is also cuttable by a segment saw of any length.
2. If a collection of polygons is cuttable by lines, then it is also cuttable by rays.
3. If a collection of polygons is cuttable by a segment saw (segment) s of length $|s|$, then it is also cuttable by any segment saw of smaller length.

It is easy to draw collections of disjoint convex polygons (even with axis-parallel rectangles) that are uncuttable by line cuts. Motivated by this state of affairs, Pach and Tardos have studied the problem of separating a large subfamily from a given family of pairwise disjoint compact convex sets on a sheet of glass, using line cuts [21]. For the case of axis-parallel rectangles, the authors show how to separate a subcollection with $\Omega(n/\log n)$ members out of given n. From the other direction, there exist instances of n rectangles such that at most cn of them can be separated in this model, where $c < 1$ is a positive constant. Far weaker guarantees, sublinear in n, can be made for arbitrary convex polygons. For other related results see [2, 8, 9, 18, 22]. In Section 3 we prove:

Theorem 2. *Given a collection $\mathcal{R}$ of n disjoint axis-parallel rectangles drawn on a planar piece of material, $\mathcal{R}$ is cuttable by rays, so in particular by a segment saw of any length. The cutting sequence can be computed in $O(n \log n)$ time and uses at most $4n$ ray cuts, which is optimal in the worst case.*

In Section 4 we exhibit some uncuttable collections of disjoint polygons.

Theorem 3. *There exist collections of disjoint polygons that are uncuttable by any segment saw. Such collections can be realized with convex or not necessarily convex polygons, and even with rectangles, or triangles.*

On the other hand we have the following positive result (in Section 4).

Theorem 4. *Given a collection $\mathcal{P}$ of segment-cuttable polygons drawn on a planar piece of material such that no two polygons in $\mathcal{P}$ touch each other, $\mathcal{P}$ is always cuttable by a sufficiently short segment saw.*

In Section 5 we prove:

Theorem 5. *Consider a collection $\mathcal{P}$ of k disjoint polygons with n vertices in total drawn on a planar piece of material Q. Then there exists an algorithm that computes in $O(n^6)$ time a suitable sequence of ray cuts for cutting the polygons in $\mathcal{P}$ out of Q when $\mathcal{P}$ is ray-cuttable and otherwise reports $\mathcal{P}$ as uncuttable.*

The same algorithm can be adapted to the case of line cuts resulting in a faster running time of $O(n^4)$. We conclude in Section 6 with some open problems.

[2]Recall, this means cuttable by some (possibly short) segment saw.

2. Cutting out a single polygon using a segment saw

In this section we prove Theorem 1. Let OPT and ALG denote the lengths of an optimal cutting sequence and that of a given algorithm being analyzed. We first show that the approximation algorithm of Demaine et al. [11] for cutting out a polygon achieves ratio 3 in the length measure. Moreover, this ratio cannot be improved as long as one uses the trivial lower bound on OPT given by the perimeter of P.

The algorithm cuts the polygon P out of its convex hull, $\mathtt{conv}(P)$, by following the boundary of P (in a chosen fixed direction, clockwise or counterclockwise) and removing the material in the pockets of P (the maximal connected components of $\mathtt{conv}(P) \setminus P$). The pockets of P are thereby cut out sequentially; we refer the reader to [11] for details. Note that reflex vertices in a pocket K correspond to convex vertices of the target polygon P, and vice versa. By the characterization of Demaine et al. [11], P is cuttable by a (short) segment saw if and only if no two reflex vertices of P are consecutive; equivalently, no two convex vertices of any pocket K are consecutive.

Let a_i and b_i (in this order) be the two edges incident to a convex vertex of K, along the chosen direction. By the above characterization, any two terms a_i and b_j are disjoint, i.e., they denote distinct edges of K. Let $i = 1, \ldots, k$ be the sequence of convex vertices along the same direction. Write $A = \sum_{i=1}^{k} |a_i|$ and $B = \sum_{i=1}^{k} |b_i|$. The cutting algorithm (illustrated in [11, Figure 5, p. 74]) gives a cost arbitrarily close to $2A + 3B$, while obviously $\mathsf{OPT} \geq A + B$ (the trivial lower bound on OPT). If $A \to 0$ and $A \ll B$ then the ratio can be arbitrarily close to 3:

$$\frac{\mathsf{ALG}}{\mathsf{OPT}} \leq \frac{2A + 3B}{A + B} = 2 + \frac{B}{A + B} \to 3.$$

Moreover, such a polygon (with $A \to 0$ and $A \ll B$) can be constructed by choosing small $|a_i|$

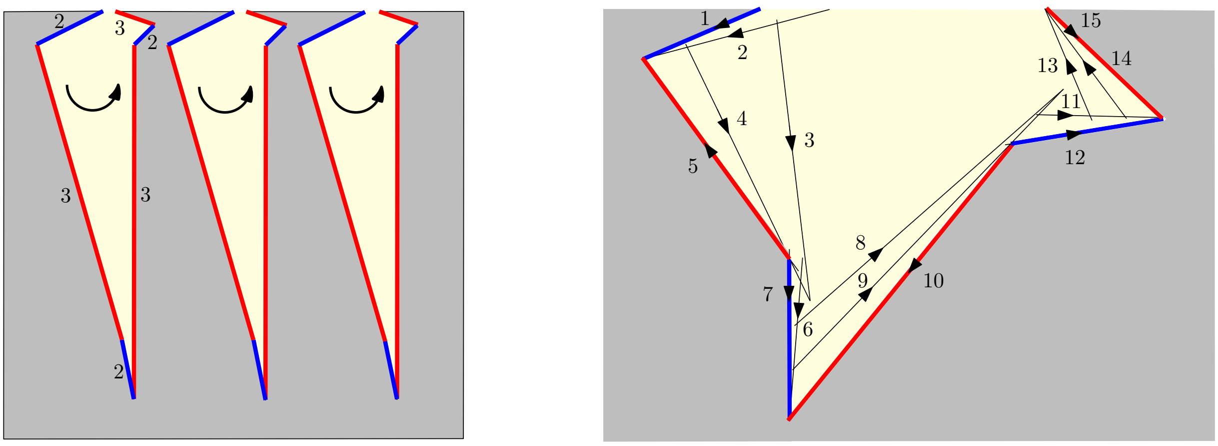

Figure 4: Left: a polygon with many thin pockets for which cutting length is close to 3 times the perimeter. The B-parts (which generate cuts of length close to $3B$) are drawn in red lines; the pockets are processed in the order indicated by the arrows. The number of cuts (2 or 3) corresponding to each edge is specified. Right: Details of cutting out one of the pockets; for clarity the internal angles and edge-lengths have been altered.

and large $|b_i|$ in the pockets and minimizing other parts on the perimeter in comparison with the pockets. See Figure 4.

The revised algorithm chooses the best direction for cutting out each pocket K from the two possible, clockwise or counterclockwise, namely the direction for which $A \geq B$. Recall that the cutting algorithm gives a cost arbitrarily close to $2A + 3B$, while $\mathsf{OPT} \geq A + B$. Hence the ratio is arbitrarily close to

$$\frac{\mathsf{ALG}}{\mathsf{OPT}} \leq \frac{2A + 3B}{A + B} = 2 + \frac{B}{A + B} \leq 2.5,$$

as desired.

Saw Length. As remarked by Demaine et al. [11], it would be interesting to compute the length of the largest segment saw that can be used to cut out a cuttable polygon. While this question remains unresolved, the following observation gives an easy upper bound.

Observation 2. *Let P be a segment-cuttable polygon drawn on a piece of material Q (in its interior) and E be the set of its edges. Let $l_1(e)$ and $l_2(e)$ be the lengths of the two extensions of $e \in E$ at both endpoints until they intersect the interior of P and let $L(e) = max\{l_1(e), l_2(e)\}$. If an extension does not intersect the interior, we consider its length to be infinity. If L is the maximum length of a segment saw that can be used to cut P out of Q, then*

$$L \leq \min_{e \in E} L(e).$$

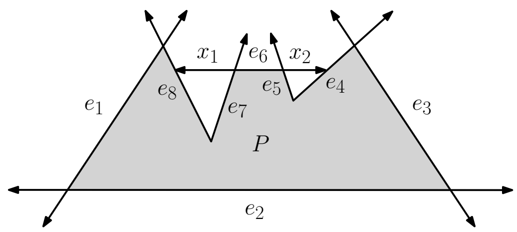

Figure 5: A line represents the two end extensions and a ray represents one end extension of an edge. If $x_1 \geq x_2$, $L \leq x_1$ is needed.

Proof. For an illustration refer to Figure 5. To remove the material adjacent to an edge e, we need to place the saw on one of the extensions of e and the maximum length of the saw that can be used for e is the maximum of the lengths of the two extensions, i.e., $L(e)$. Since a fixed length saw is used for every edge of P, we need to take the minimum of all $L(e)$, $e \in E$. □

3. Cutting out a collection of axis-parallel rectangles using a segment saw

In this section we prove Theorem 2. As it turns out, the problem of cutting out a collection of axis-parallel rectangles by a segment saw is very much related to the problem of separating such a collection by moving the rectangles, one at at time, to infinity, using translations. We start by recalling the following classical result of Guibas and Yao [17] concerning translations of rectangles in a common direction.

Lemma 1 (Guibas and Yao [17]). *Let $\mathcal{R}$ be any set of n disjoint axis-parallel rectangles in the plane, and θ be any direction. Then there is an ordering $R_1, \ldots, R_n$ of the rectangles such that R_i can be moved continuously to infinity in direction θ without colliding with the rectangles R_j, $1 \leq i < j \leq n$. Such an ordering can be computed in $O(n \log n)$ time.*

The reader can verify that the proof of Lemma 1 in [17] implies the following stronger result; see also [15, Lemma 3].

Lemma 2 (Guibas and Yao [17])**.** *For any set of n disjoint axis-parallel rectangles in the plane, there is an ordering $R_1, \ldots, R_n$ of the rectangles such that R_i can be moved continuously to infinity in any direction between 0 and $\pi/2$ without colliding with the rectangles R_j, $1 \le i < j \le n$. Such an ordering can be computed in $O(n \log n)$ time.*

In Lemma 2 the set of directions under discussion make the closed interval $[0, \pi/2]$. It is now convenient to reformulate this lemma in terms of our interest.

Notation. Given two points $p, q \in \mathbf{R}^2$, *p dominates q* if the inequalities $x(p) > x(q)$ and $y(p) > y(q)$ hold among their x- and y-coordinates. Given two axis-parallel rectangles R', R'', write $R'' >_x R'$ if there exists a vertical line that separates R' and R'', so that R' lies in the left (closed) halfplane and R'' lies in the right (closed) halfplane determined by the line.

Lemma 3. *For any set of n disjoint axis-parallel rectangles in the plane, there is an ordering $R_1, \ldots, R_n$ of the rectangles such that R_i is unblocked in any direction between 0 and $\pi/2$ by any of the rectangles R_j, $1 \le i < j \le n$. Such an ordering can be computed in $O(n \log n)$ time.*

To be precise, R_i is *unblocked* in any direction between 0 and $\pi/2$ by any of the rectangles R_j, $i < j \le n$, if and only if no vertex of such a rectangle dominates the lower left corner of R_i; see Figure 6.

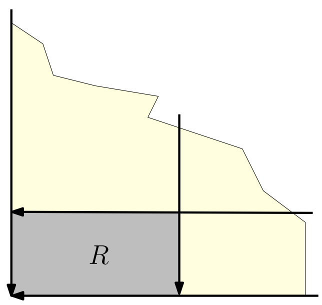

Figure 6: R is unblocked hence it is cuttable.

The two-step algorithm from [17] for computing the order is as follows.

Step 1: Sort the rectangles by decreasing order of their y-coordinate of the top side; let $R_1, \ldots, R_n$ be the resulting order.

Step 2: Start with an empty list L and add the rectangles R_i, for $i = 1, \ldots, n$, in this order. Place each new rectangle R in the first (i.e., leftmost) position in L consistent to the constraint that $R >_x S$ for every rectangle S following R in L.

An illustration of the ordering produced appears in Figure 7. An obvious implementation of Step 2 takes $O(n^2)$ time, however, Guibas and Yao [17] showed that it is possible to use balanced trees in a non-trivial way to reduce the time to $O(n \log n)$. Consequently, the two-step algorithm runs in $O(n \log n)$ time. Using the ordering guaranteed by Lemma 3, we get our desired result.

Proof of Theorem 2. We use the ordering provided by Lemma 3 and cut out rectangles one by one in this order; there are n iterations, $i = 1, \ldots, n$. The following invariant is maintained: in iteration i, rectangles $R_1, \ldots, R_{i-1}$ have been cut out (i.e., each has detached on a separate piece of material), and the current piece of material contains the subcollection $R_i, \ldots, R_n$. Observe that in iteration i, rectangle R_i is unblocked in any direction between 0 and $\pi/2$, hence it is cuttable as follows; refer again to Figure 6.

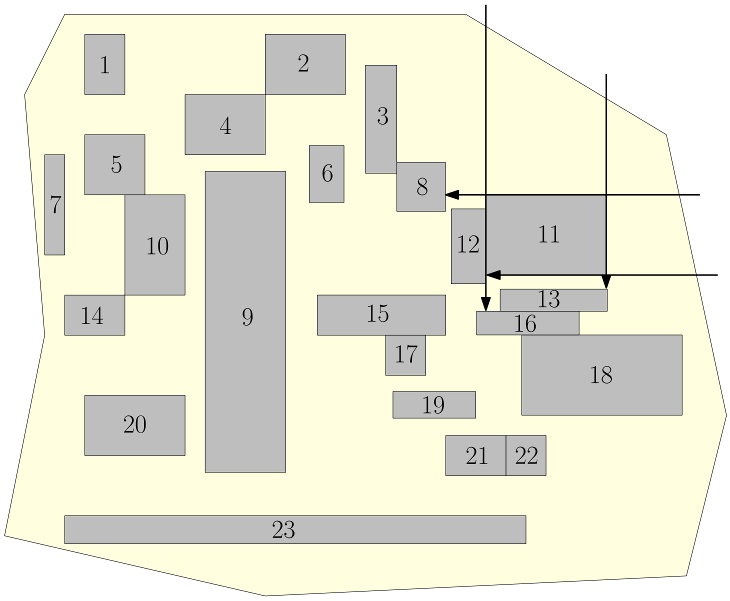

Figure 7: STEP 2 of the algorithm. After $R_1, \ldots, R_{23}$ are successively inserted into L, the list is $11, 13, 12, 16, 18, 22, 8, 3, 2, 6, 15, 17, 19, 21, 4, 9, 1, 5, 10, 14, 20, 7, 23$. The first rectangle to be cut out is 11.

We execute two ray cuts from infinity (or from the boundary of the material): one vertical (going down) and one horizontal (going left); the two cuts meet at the lower left corner of R_i. Further, each ray cut is extended until it hits the interior of a rectangle or the boundary of the current piece of material. The effect is detaching R_i from the piece of material containing the remaining rectangles R_j, $i < j \leq n$. The piece containing R_i contains no other rectangles, so two more cuts along two sides of R_i suffice to completely separate R_i completely, for a total of at most 4 cuts per rectangle. The process is continued until all rectangles are cut out in this way, with at most $4n$ cuts overall.

Clearly some collections require at least 4 cuts for each rectangle, e.g., if no two rectangle sides are collinear, for a total of at least $4n$ cuts. Hence the number of cuts executed in the algorithm is worst-case optimal.

Remark. There are cases when as few as $\Theta(\sqrt{n})$ cuts suffice, for instance when the rectangles are arranged in a square grid formation with their sides aligned.

4. Cuttable and uncuttable collections by a segment saw

It is easy to exhibit collections of disjoint polygons that are uncuttable by a segment saw, especially if one uses non-convex polygons or convex polygons with many sides. The problem becomes more interesting when one restricts the number of sides of the polygons or their shape. In this section we prove Theorem 3. An uncuttable collection of rectangles (in arbitrary orientations) is shown in Figure 8 (left); this example has $n = 12$ rectangles, 6 drawn on the sides of a regular hexagon in its exterior, and 6 drawn in its interior. A similar pattern can be realized for every $n \geq 10$ (with $\lfloor n/2 \rfloor$ rectangles on the outer boundary of the union and the remaining $\lceil n/2 \rceil$ inside). Observe that every possible cut that can be initiated from outside is blocked by one of the small rectangles; moreover, all intersecting cuts are incident to some tangency point between two consecutive outer rectangles. Hence none of the rectangles can be separated. In contrast, the similar looking construction with $n = 7$ rectangles shown in Figure 8 (right) is cuttable by rays,

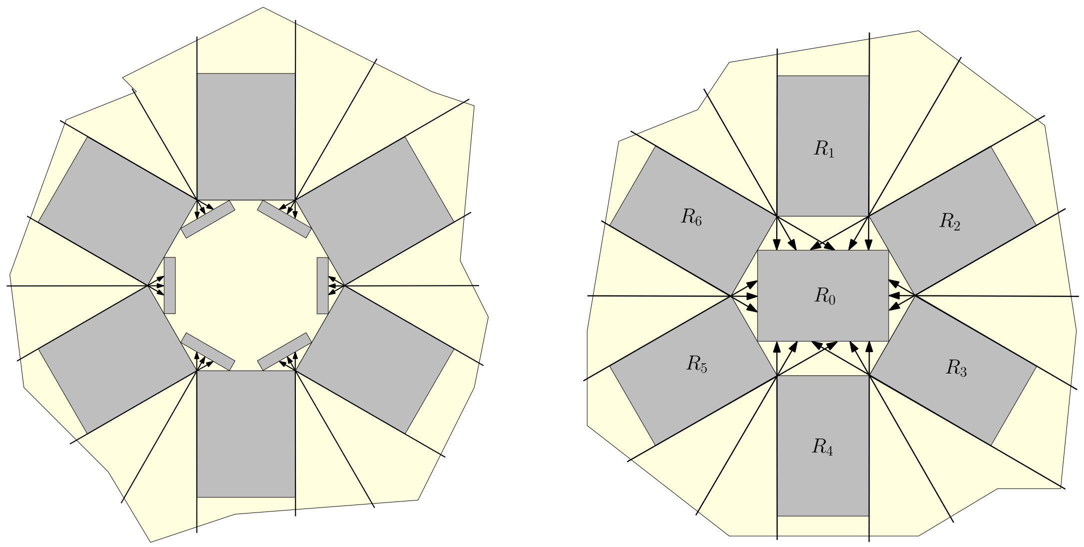

Figure 8: Left: an uncuttable collection of $n = 12$ rectangles; arrows represent saw cuts that fail in cutting out any rectangle. Right: a cuttable collection of $n = 7$ rectangles.

hence in particular, by a segment saw of any length. This particular example has $n = 7$ rectangles, 6 drawn on the sides of a regular hexagon in its exterior, and one drawn in its interior. Start by separating R_1 by using two ray cuts aligned with the long sides of its left and right neighbor rectangles. Observe that these two rays cross each other, so the piece of material containing the top rectangle can be detached. Once one of the rectangles has been cut out separately, the rest can be easily cut out one by one, and so can the entire collection.

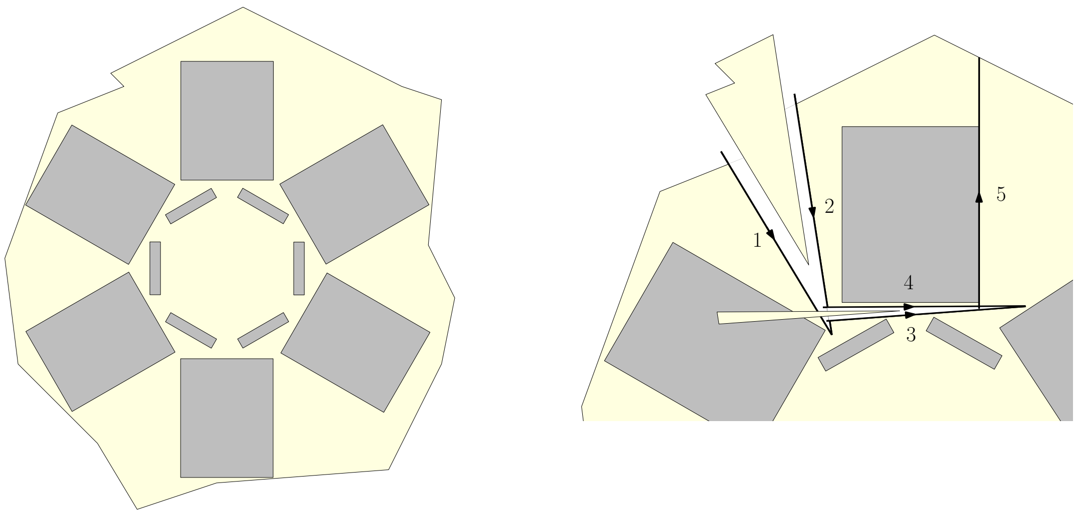

Figure 9: Left: a collection of $n = 12$ rectangles where no two rectangles touch each other. The collection is now cuttable by a (sufficiently short) segment saw. Right: a magnified view.

If we slightly modify the placements of the rectangles in Figure 8 (left) so that no two rectangles touch each other, as shown in Figure 9 (left), then the resulting collection becomes cuttable using a segment saw. Indeed, every outer rectangle can be separated out as shown in Figure 9 (right). Next, using a similar approach, the inner rectangles can also be separated out. After separation, it is easy to remove any material adjacent to an edge of a rectangle.

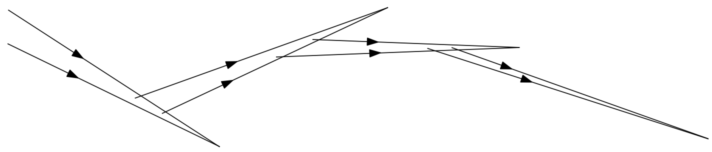

Figure 10: Double cuts (pairs of segment cuts of about the same length) can be used to cut along any polygonal line achieving separation of two or more polygons.

This approach can be applied to any collection of segment-cuttable polygons where no two polygons touch each other. Separation can be achieved by using double cuts as shown in Figure 10. After separation, the individual polygons can be cut out using the algorithm from [11]. We have thereby proved Theorem 4.

An uncuttable collection of triangles is shown in Figure 11 (left). Such collections can be also realized with a larger number of triangles. For any $k \geq 4$, it is straightforward to draw uncuttable k-gon collections using $k+1$ polygons $P_0, P_1, \dots, P_k$. Constructions for $k = 4$ and $k = 6$ are shown in Figure 11 (middle and right).

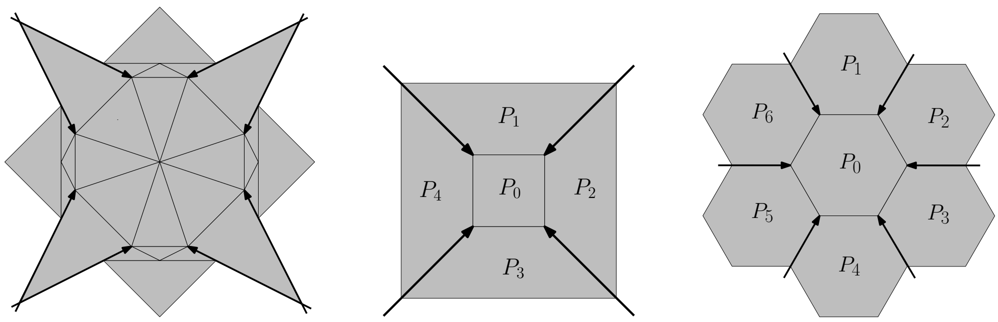

Figure 11: Uncuttable k-gon collections. Left: construction with triangles ($k = 3$). Middle: $k = 4$. Right: $k = 6$. Arrows represent the only possible useful cuts in the collection.

Remarks. It is interesting to observe that separability by translations in a single direction holds for any collection of disjoint convex bodies; see also [4, Theorem 1], [15, Lemma 1], [19, Theorem 8.7.2]. Theorem 6 below appears in the work of Fejes Tóth and Heppes [16], but it can be traced back to de Bruijn [6]; the algorithmic aspects of the problem have been studied by Guibas and Yao [17].

Theorem 6. [6, 16, 17] *Any set of n convex objects in the plane can be separated via translations all parallel to any given fixed direction, with each object moving only once. If the top and bottom points of each object are given, an ordering of the moves can be computed in $O(n \log n)$ time.*

In general, separability via translation does not imply cuttability. Observe that the collections in Figures 8 (left) and 11 are not cuttable by a segment saw although they can be separated via translations along any fixed direction as stated in Theorem 6. On the other hand, the broader variant of separability for axis-aligned rectangles stated in Lemma 2 (so that R_i can be translated along any direction in the first quadrant) finally allows cuttability by rays for any family of axis-parallel rectangles.

5. Cutting out a collection of polygons using ray cuts

In this section we prove Theorem 5. Consider a collection $\mathcal{P}$ of k disjoint polygons $P_1, \dots, P_k$, with n vertices in total, drawn on a planar piece of material Q. Recall that a ray cut comes from infinity and can stop at any point. We may assume that each P_i is ray-cuttable by itself; otherwise the collection is not ray-cuttable. (Note that a collection of convex polygons is not necessarily ray-cuttable; see Figure 11.)

We present a polynomial-time algorithm that computes a suitable cutting sequence to cut the polygons in $\mathcal{P}$ out of Q using ray cuts when $\mathcal{P}$ is ray-cuttable and otherwise reports $\mathcal{P}$ as uncuttable.

A subcollection of polygons $\mathcal{P}' \subset \mathcal{P}$ drawn on a piece of material $Q' \subset Q$ is called *separated* if Q' is already detached from Q after executing a sequence of ray cuts. Observe that a ray cut can produce multiple (separated) subcollections; in that case, we say that the ray cut *achieves some separation*. In order to cut out the polygons in $\mathcal{P}$, it is necessary to separate out every polygon in the given collection.

When two ray cuts r, r' meet at a point p, the non-reflex angle between the ray cuts at p is called their *internal angle* and is denoted by $\theta(r, r')$. The wedge-shaped piece of material enclosed by r, r' when $\theta(r, r')$ is considered is denoted by $W(r, r')$; see Figure 12.

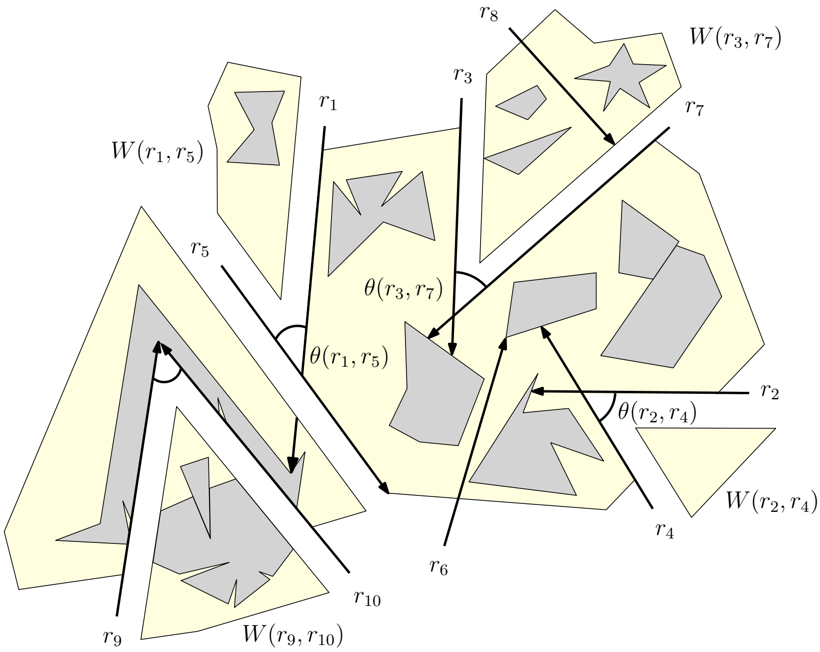

Figure 12: Let $\mathcal{S} = r_1, \ldots, r_{10}$ be a sequence of executed ray cuts. (r_3, r_7) and (r_9, r_{10}) are the only separating pairs; r_5 and r_8 are the only separating rays. However, (r_1, r_5) is not a separating pair since r_5 is a separating ray.

By slightly abusing the notation, we refer to the current piece of material as Q and the subcollection present on Q as $\mathcal{P}$. Consider a sequence $\mathcal{S}$ of ray cuts executed on Q. We assume that the detached pieces of Q, with or without polygons, are removed immediately after separation and are handled independently. Furthermore, we extend any ray cut $r \in \mathcal{S}$ until it hits the interior of a polygon in $\mathcal{P}$ or the boundary of Q.

A *separating ray* is a ray cut $r \in \mathcal{S}$ executed on Q such that its endpoint is on the boundary of Q and each side of r contains a proper polygon subcollection of $\mathcal{P}$. A *separating pair (of rays)* is a pair of meeting ray cuts r, r' in $\mathcal{S}$ executed on Q such that $W(r, r')$ contains a proper polygon subcollection of $\mathcal{P}$ and r, r' are not separating rays. Refer to Figure 12 for an illustration.

The following lemma shows that there is always a separating ray or a separating pair in any sequence $\mathcal{S}$ of ray cuts that achieves some separation. The ray cuts in $\mathcal{S}$ are executed until the first separation occurs. After separation each piece of material is handled independently.

Lemma 4. *Let $\mathcal{S} = r_1, \ldots, r_s$ be a sequence of ray cuts executed in this order and achieving some separation. Then either there is a separating ray or a separating pair in $\mathcal{S}$.*

Proof. Consider the subsequence $\mathcal{S}' = r_1, \ldots, r_t$ where $t \leq s$, such that the first separation occurs when r_t is executed. If r_t is a separating ray, we are done. Now assume that r_t is not a separating ray. Since separation is achieved only after r_t is executed, then it must be the case that r_t forms a separating pair with some previously executed ray cut $r_i \in \mathcal{S}'$, with $i < t$, i.e., $W(r_i, r_t) \cap \mathcal{P}$ is not empty. □

A ray is called *canonical* if it passes through at least two polygon vertices (not necessarily of the same polygon). Clearly, there are $O(n^2)$ canonical rays in the given configuration. Any valid ray cutting sequence will repeatedly partition the given polygon collection using separating pairs or separating rays until each polygon has been separated out. A cutting sequence need not to contain canonical ray cuts. However, the following lemma shows that any valid separation can be achieved by using canonical ray cuts only. This observation is key for designing a polynomial-time ray cutting algorithm, based on finding appropriate canonical ray cuts for separation.

Lemma 5. *Let (r_1, r_2) be a separating pair. Then there exists a pair of canonical ray cuts (r_1'', r_2'') that achieves the same effect as (r_1, r_2) in terms of separability.*

Proof. We transform the pair (r_1, r_2) into the pair (r_1'', r_2'') so that the two rays intersect at all times during the transformation; refer to Figure 13. Let p be the intersection point of r_1, r_2 such that the pair separates $\mathcal{P}' \subset \mathcal{P}$. Translate r_1 towards $\mathcal{P}'$ until it touches a polygon vertex, say a, and refer to this ray as r_1'. Similarly, translate r_2 towards $\mathcal{P}'$ and obtain the ray r_2' passing through c. Then (r_1', r_2') is still a separating pair, where r_1', r_2' intersect at p'.

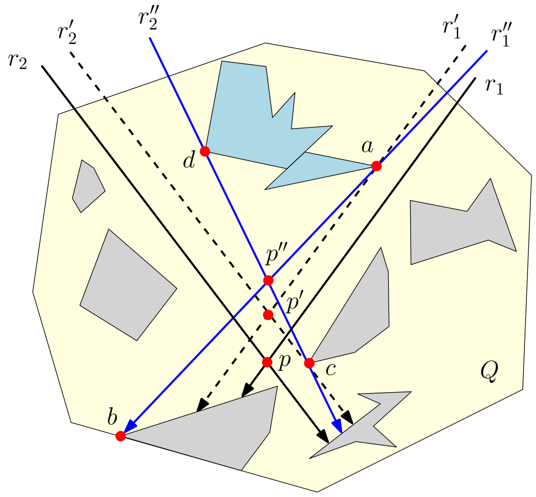

Figure 13: Separating the set of polygons $\mathcal{P}' \subset \mathcal{P}$ (in light blue) using the canonical pair (r_1'', r_2'').

For a given a ray r, let $\pi_{\text{left}}(r)$ denote the open half plane to the left of r and $\pi_{\text{right}}(r)$ denote the open half plane to the right of r. If r_1' passes through a only, we have the following two cases where r_1'' is obtained by rotating r_1' around a until it passes through a second polygon vertex b.

Case 1: If $a \in \pi_{\text{left}}(r_2')$, rotate r_1' clockwise around a.
Case 2: If $a \in \pi_{\text{right}}(r_2')$ or $a \in r_2'$, rotate r_1' counter-clockwise around a.

Similarly if r_2' passes through c only, we obtain r_2'' passing through vertices c and d by rotating r_2' around c in the following analogous way.

CASE 1: If $c \in \pi_{\text{left}}(r'_1)$ or $c \in r'_1$, rotate r'_2 clockwise around c.
CASE 2: If $c \in \pi_{\text{right}}(r'_1)$, rotate r'_1 counter-clockwise around c.

The canonical rays r''_1, r''_2 meet at p'' and can also separate P'. □

A rather straightforward argument also yields:

Lemma 6. *Let r be a separating ray. Then there exists a canonical separating ray that achieves the same partition.*

Algorithm for cutting. According to Lemmas 4,5,6, the existence of a canonical separating ray is first determined. If at least one exists, an arbitrary ray is chosen and the corresponding cut is executed. If no canonical separating ray exists, the existence of a canonical separating pair is determined. If at least one exists, an arbitrary pair is chosen and the corresponding cuts are executed. After separation the algorithm continues independently and recursively on each of the newly formed pieces; it terminates when every polygon has been separated from the rest. If no canonical separating ray or canonical separating pair exists at some point during the algorithm, $\mathcal{P}$ is reported as uncuttable.

As mentioned earlier, each ray cut is extended until it hits the interior of a polygon or the boundary of the current piece of material. (In some instances separation can be only achieved with separating rays, e.g., for a collection of stacked congruent axis-aligned rectangles.)

After each polygon in $\mathcal{P}$ has been separated out, it is cut out using ray cuts. This step can be achieved by considering one end extensions of the edges that have some material attached to them, to form the necessary ray cuts (refer to Observation 1).

Analysis. There are $O(n^2)$ canonical ray cuts and $O(n^4)$ canonical pairs of ray cuts in any given configuration. Hence we have $O(n^4)$ ways to cut in total. Verifying whether a canonical pair is a separating pair and a canonical ray is a separating ray can be done in $O(n)$ time. In the worst case, we need to execute $O(n)$ canonical separating pairs and rays. After each execution, the bookkeeping of separation and creation of new pieces can be performed in $O(n)$ time. Thus, separation of the polygons can be achieved in $O(n^6)$ time. After the separation step, in $O(n)$ time we can remove the pieces of material adjacent to the edges of the polygons. Hence the overall running time of the algorithm is $O(n^6)$. This completes the proof of Theorem 5. □

Line Cuts. The above algorithm can be easily adapted for cutting out polygon collections using line cuts instead of ray cuts. Similar to the approach described above, we will use only canonical separating lines and since there are $O(n^2)$ of them, the algorithm would run in $O(n^4)$ time.

6. Open problems

1. The obvious remaining open problem is devising an algorithm, which, given a collection of disjoint polygons in the plane determines whether it is cuttable by a segment saw, and computes a suitable cutting sequence if it is. We conjecture that the problem admits a polynomial-time algorithm.
2. Can the cutting algorithm presented in Section 5 (or its analysis) be substantially improved? Is there a substantially faster algorithm?

References

[1] S. I. Ahmed, M. A. Islam, and M. Hasan, Cutting a cornered convex polygon out of a circle, *Journal of Computers* **5(1)** (2010), 4–11.

[2] N. Alon, M. Katchalski, and W. R. Pulleyblank, Cutting disjoint disks by straight lines, *Discrete & Computational Geometry* **4** (1989), 239–243.

[3] S. Bereg, O. Dăescu, and M. Jiang, A PTAS for cutting out polygons with lines, *Algorithmica* **53** (2009), 157–171.

[4] S. Bereg, A. Dumitrescu, and J. Pach, Sliding disks in the plane, *International Journal of Computational Geometry & Applications* **15(8)** (2008), 373–387.

[5] J. Bhadury and R. Chandrasekaran, Stock cutting to minimize cutting length, *European Journal of Operations Research* **88** (1996), 69–87.

[6] N. G. de Bruijn, Aufgaben 17 and 18 (in Dutch), Nieuw Archief voor Wiskunde **2** (1954), 67.

[7] R. Chandrasekaran, O. Dăescu, and J. Luo, Cutting out polygons, *Proceedings of the 17th Canadian Conference on Computational Geometry* (2005), pp. 183–186.

[8] J. Czyzowicz, E. Rivera-Campo, and J. Urrutia, Separating convex sets in the plane, *Discrete & Computational Geometry* **7** (1992), 189–195.

[9] J. Czyzowicz, E. Rivera-Campo, and J. Urrutia, Separation of convex sets, *Discrete Applied Mathematics* **51(3)** (1994), 325–328.

[10] O. Dăescu and J. Luo, Cutting out polygons with lines and rays, *International Journal of Computational Geometry & Applications* **16(2-3)** (2006), 227–248.

[11] E. D. Demaine, M. L. Demaine, and C. S. Kaplan, Polygons cuttable by a circular saw, *Computational Geometry: Theory & Applications* **20** (2001), 69–84.

[12] A. Dumitrescu, An approximation algorithm for cutting out convex polygons, *Computational Geometry: Theory & Applications* **29** (2004), 223–231.

[13] A. Dumitrescu, The cost of cutting out convex n-gons, *Discrete Applied Mathematics* **143** (2004), 353–358.

[14] A. Dumitrescu and M. Hasan, Cutting out polygons with a circular saw, *International Journal of Computational Geometry & Applications* **23(2)** (2013), 127–139.

[15] A. Dumitrescu and M. Jiang, On reconfiguration of disks in the plane and other related problems, *Computational Geometry: Theory & Applications* **46(3)** (2013), 191–202.

[16] L. Fejes Tóth and A. Heppes, Über stabile Körpersysteme (in German), *Compositio Mathematica* **15** (1963), 119–126.

[17] L. Guibas and F. F. Yao, On translating a set of rectangles, in *Computational Geometry*, F. Preparata (ed.), pp. 61–67, Vol. 1 of *Advances in Computing Research*, JAI Press, London, 1983.

[18] K. Hope and M. Katchalski, Separating plane convex sets, *Mathematica Scandinavica* **66** (1990), 44–46.

[19] J. O'Rourke, *Computational Geometry in C*, Cambridge University Press, 2nd edition, 1998.

[20] M. H. Overmars and E. Welzl, The complexity of cutting paper, *Proceedings of the 1st Annual ACM Symposium on Computational Geometry* (1985), pp. 316–321.

[21] J. Pach and G. Tardos, Cutting glass, *Discrete & Computational Geometry*, **24** (2000), 481–495.

[22] J. Pach and G. Tardos, Separating convex sets by straight lines, *Discrete Mathematics* **241(1-3)** (2001), 427–433.

[23] X. Tan, Approximation algorithms for cutting out polygons with lines and rays, *Proceedings of the 11th Annual International Computing and Combinatorics Conference*, Vol. 3595 of *LNCS*, Springer, (2005), pp. 534–543.