On Collections of Polygons Cuttable with a Segment Saw

Adrian Dumitrescu, Anirban Ghosh, Masud Hasan

Presentation by

Anirban Ghosh

anirban@uwm.edu

CALDAM 2015

Polygon Cutting

Introduced by Overmars and Welzl (1985).

Objective.

To cut out efficiently a polygon P or a collection of polygons, drawn on a piece of planar material Q.

Efficiency measures.

1. Total length of the cuts
2. Total number of the cuts

Applications.

Cutting out polygonal objects in manufacturing industries, such as metal sheet cutting, paper cutting etc.

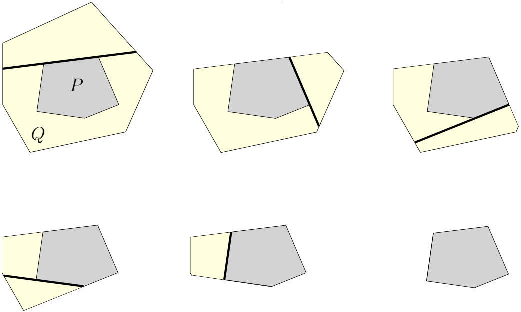
Cutting tool variants

1. Line cuts
2. Ray cuts
3. Segment cuts

Fact.

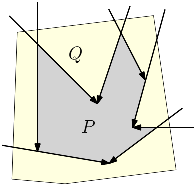
Cutting tool variant determines the class of cuttable polygons.

Line Cuts

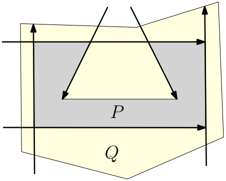


- A line cut is a line that does not go through P and cuts the current piece of material into two or more pieces.
- For cutting P out of Q by line cuts, P *must be convex*.

Ray Cuts



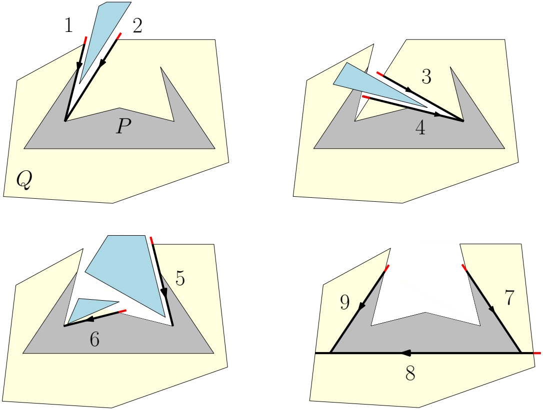
Ray-cuttable



Ray-uncuttable

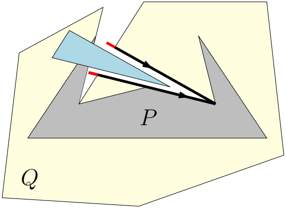
- A ray cut comes from ∞ and can stop at any point outside P (not necessarily cutting the material into pieces).
- Objective is to cut out *non-convex* polygons. However, not all non-convex polygons can be cut by ray cuts.

Introduced by E. D. Demaine, M. L. Demaine and Kaplan (2001).



- A segment cut is like a ray cut, but *may not* come from ∞. The saw is abstracted as a line segment, which cuts if moved along its supporting line.

Segment Cuts

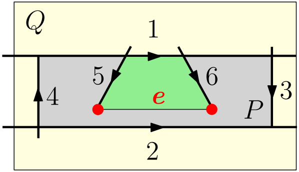


- If a small free space within Q is available, a saw cut can be initiated there by maneuvering the saw.
- The length of the saw was assumed to be arbitrarily small, and its width to be 0. Thus, a cut can be initiated from an arbitrarily small free space.
- Turns are not possible when making cuts.

Existing results on Segment Cuts

- Demaine et al. (2001) presented an algorithm for cutting P out of its convex hull, with the total number of cuts and the total length of the cuts, both within 2.5 times their respective optimums. (*We show that the approximation factor for the length measure is in fact* 3.)

- They also showed that P is segment saw cuttable *iff* P *does not* have 2 consecutive reflex vertices (interior angle $> \pi$).



- Dumitrescu and Hasan (2011), improved the approximation guarantee for the total number of cuts from 2.5 to 2.

Our results on Segment Cuts

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

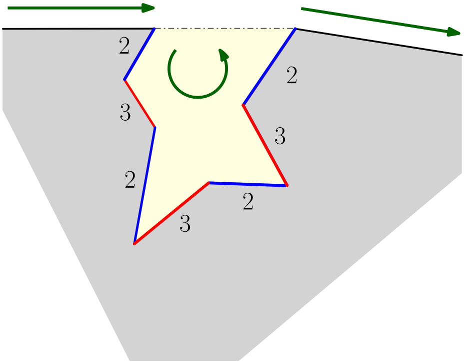
Question asked by Demaine et al. (2001).

What collections of non-overlapping polygons in the plane can be simultaneously cut out by a segment saw?

2. Any collection of n interior-disjoint axis-parallel rectangles is cuttable by at most $4n$ ray cuts (in particular segment cuts) and there is an algorithm that runs in $O(n \log n)$ time for computing a suitable cutting sequence.

3. There exist uncuttable collections of interior-disjoint rectangles (in arbitrary orientations). We also present various uncuttable collections of interior-disjoint polygons, including triangles.

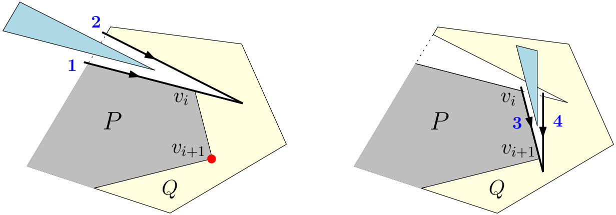
Cutting out a Single Polygon [by Demaine et al.]



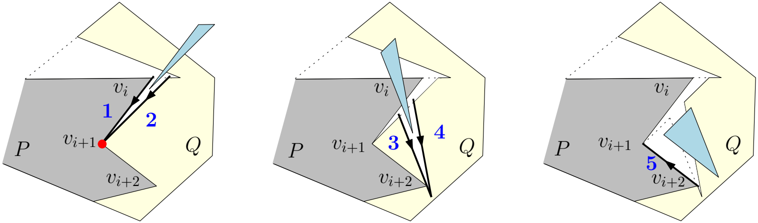
The number of required cuts (2 or 3) corresponding to each edge is specified.

- Pockets are cut out sequentially (clockwise or anticlockwise).

Cutting out a Single Polygon [by Demaine et al.]

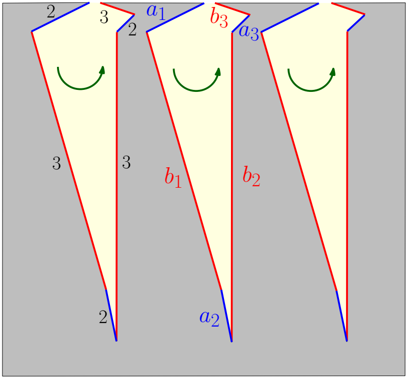


Case 1 : v_{i+1} is convex, cost for the edge $v_i v_{i+1} \approx 2|v_i v_{i+1}|$.



Case 2 : v_{i+1} is reflex, cost for the edges $v_i v_{i+1}$ and $v_{i+1} v_{i+2} \approx 2|v_i v_{i+1}| + 3|v_{i+1} v_{i+2}|$.

An instance where the ratio $\to 3$



- The authors claim that their algorithm achieves ratio 2.5 on the length measure.
- $A = a_1 + a_2 + a_3$,
  $B = b_1 + b_2 + b_3$.
- We choose small value for A and large value for B in the pockets.
- ALG $\approx 2A + 3B$, OPT $\geq A + B$

If $A \to 0$ and $A \ll B$ then,

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

A revised 2.5–Approximation Algorithm

We revise the algorithm as follows.

Modification.

Choose the best direction for cutting out each pocket (clockwise/anticlockwise), namely the direction for which $A \geq B$.

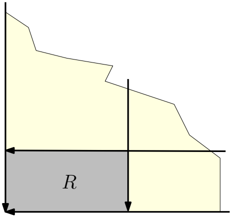
If $A \geq B$ then,

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

Cutting out a Collection of Axis-parallel Rectangles

Reasons for non-triviality.

- A segment cut cannot be initiated unless there is a free space.
- Computation of a valid cuttable sequence of rectangles.



R is unblocked since there is no rectangle in the yellow region and hence it is cuttable using 4 cuts.

Idea.

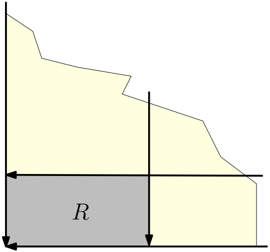
Given any configuration, $\exists$ an *unblocked* rectangle.

Cutting out a Collection of Axis-parallel Rectangles

Our approach is based on a classical result by Guibas and Yao (1983) regarding separability of objects.

Lemma.

For any set of n interior-disjoint axis-parallel rectangles in $\mathbf{R}^2$, $\exists$ an ordering $R_1, ..., R_n$ of the rectangles s.t. R_i is unblocked in any direction $\in [0, \pi/2]$ by any $R_j, 1 \leq i < j \leq n$. Such an ordering can be computed in $O(n \log n)$ time.

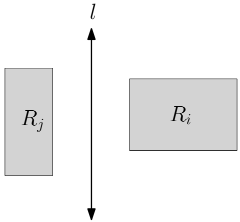


R is unblocked.

Implication.

R_i is *unblocked* in any direction $\in [0, \pi/2]$ by any of the rectangles R_j, $i < j \leq n$, *iff* no vertex of such a rectangle dominates the lower left corner of R_i.

Cutting out a Collection of Axis-parallel Rectangles



Notation.

$R_i >_x R_j$ if $\exists$ a vertical line l such that R_j lies to the left of l and R_i to the right.

The Algorithm.

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.

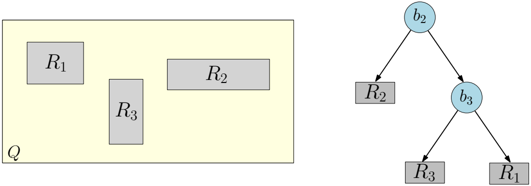
2. Start with an empty list L and add the rectangles R_i, for $i = 1, \ldots, n$, in this order. Place each new rectangle R in the first (leftmost) position in L such that $R >_x S$ for every rectangle S following R in L.

Cutting out a Collection of Axis-parallel Rectangles

- We will use height balanced tree (for example **AVL**) for step 2 in a way s.t. the inorder traversal of the leaves gives a valid cutting sequence of the rectangles. (*Naive approach takes $O(n^2)$ time*.)

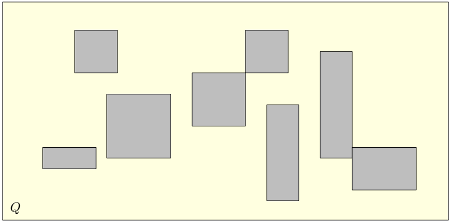
- At each internal node, we keep the rightmost boundary of all the right boundaries of the descendant rectangles.

  b_i refers to the x-coordinate of the right boundary of R_i.

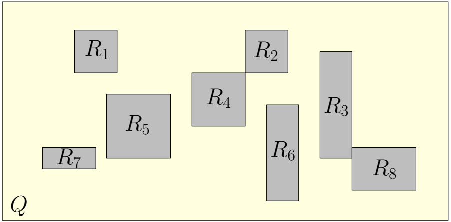


- Each insertion requires $O(\log n)$ time. Total running time of the algorithm : $O(n \log n)$.

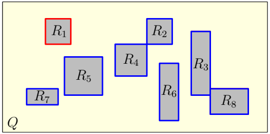
An input instance



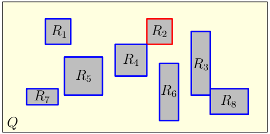
Sort the rectangles

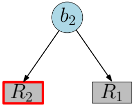


Insert R_1

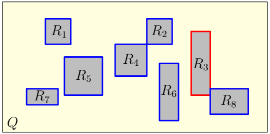


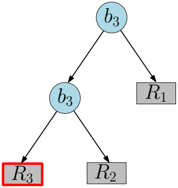
Insert R_2



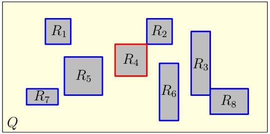


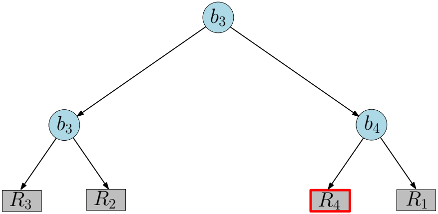
Insert R_3



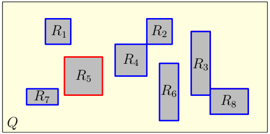


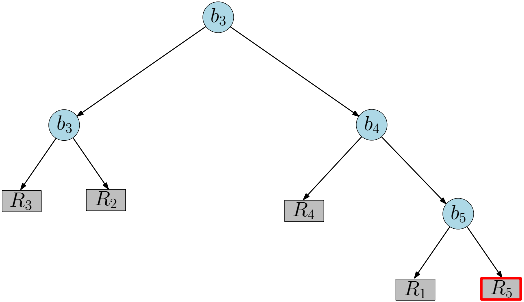
Insert R_4



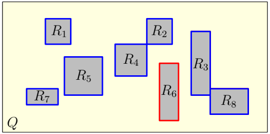


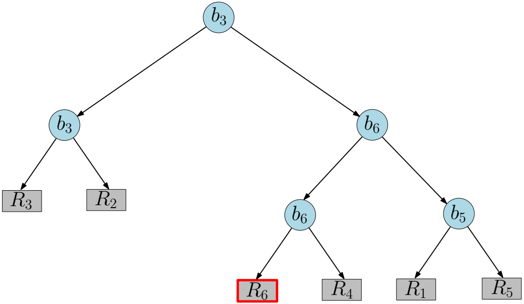
Insert R_5



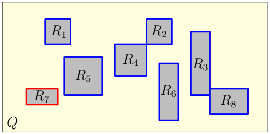


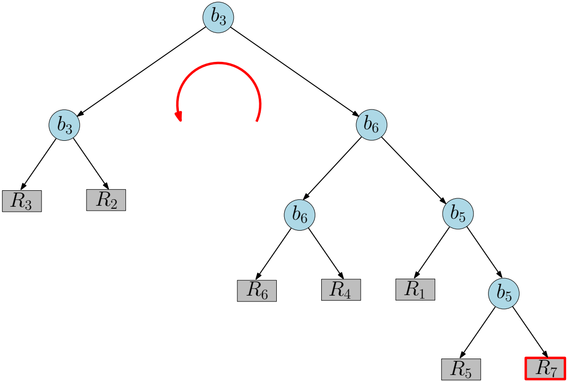
Insert R_6



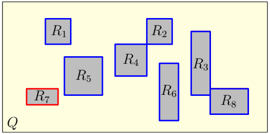


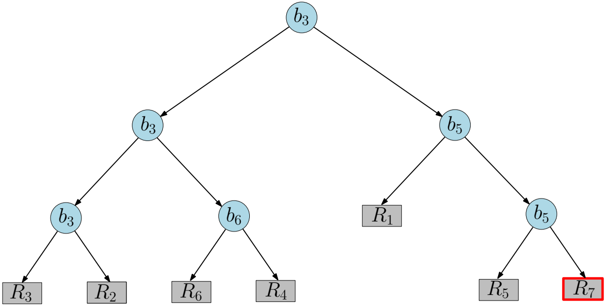
Insert R_7, Balance



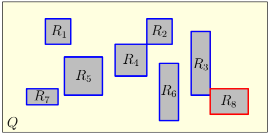


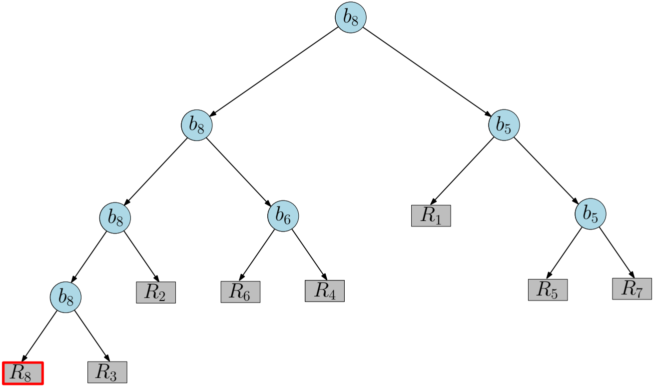
Insert R_7



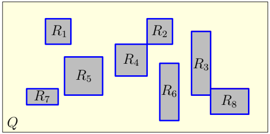


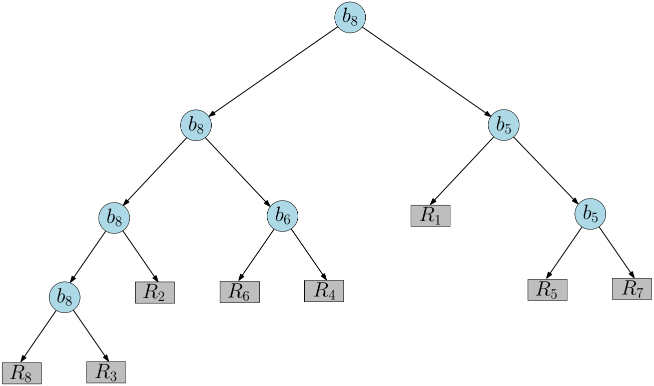
Insert R_8





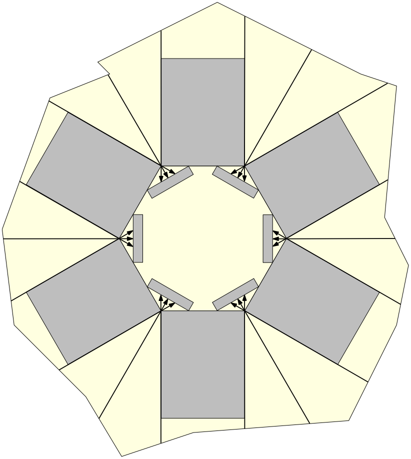
Final Cutting Sequence



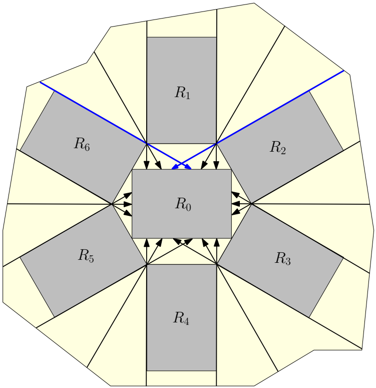


Cutting sequence : $R_8, R_3, R_2, R_6, R_4, R_1, R_5, R_7$

Uncuttable Rectangle Collections

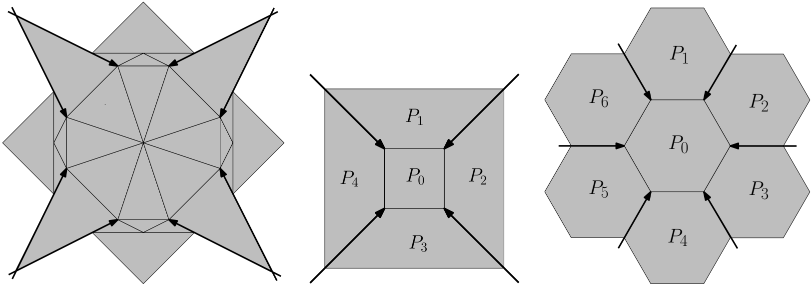


Uncuttable Collection



Cuttable Collection

Uncuttable k-gon Collections



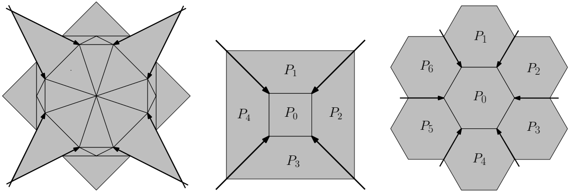
$k = 3$ $\quad$ $k = 4$ $\quad$ $k = 6$

For any $k \geq 4$, uncuttable k-gon collections can be drawn using $k + 1$ number of k-gons $P_0, P_1, \ldots, P_k$.

Translability does not guarantee Cuttability

Theorem. [Fejes Tóth and Heppes (1963)]

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 once only.



Observation.

The theorem holds true for the presented collections but they are not cuttable using a segment saw.