Rectangular Partitioning of Rectilinear Polygons

Exercise 15

Alexandre Ros

November 7, 2023

Introduction to the problem

Exercise 15

Decompose a rectilinear polygon into rectangles, using segments aligned with the edges of the polygon. The algorithm must produce the **minimum number of pieces**, and the output must give a complete description of the partition.

Input

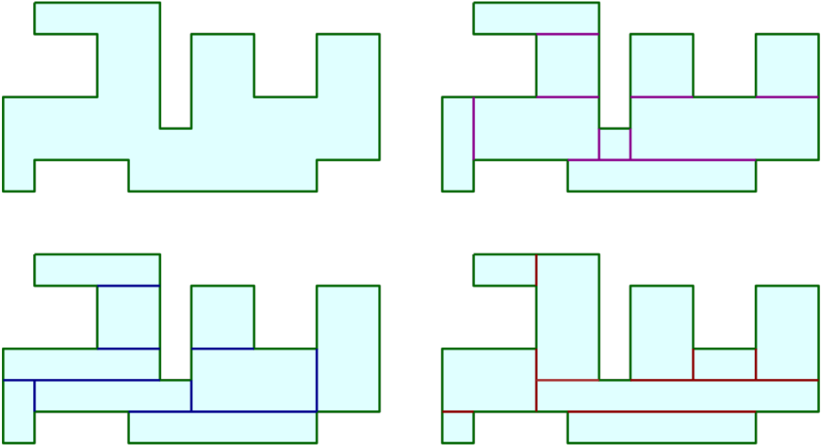
A polygon whose sides meet at right angles.

Output

A **partition** of the polygon into rectangles, with no overlaps.

We will call this problem *MNC* (minimal nonoverlapping cover).

Examples



Multiple partitions may exist

Preliminaries

Concave and convex vertices

In a rectilinear polygon, we can distinguish between concave and convex vertices.

- A vertex is said to be **concave** if its interior angle is over 180°.
- A vertex is said to be **convex** if its interior angle is under 180°.

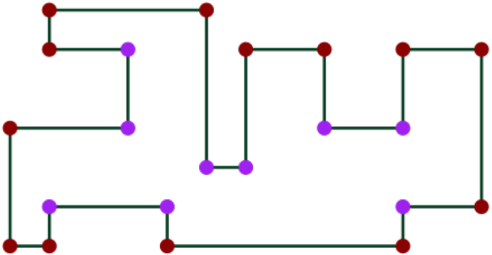


Figure 1: Convex (red) and concave (purple) vertices.

Co-vertical, co-horizontal and chords

- Two vertices (x_1, y_1), (x_2, y_2) that share no edge are **co-vertical** $\iff y_1 = y_2$ and **co-horizontal** $\iff x_1 = x_2$.
- A **chord** is a segment fully contained inside the polygon that connects two co-horizontal or co-vertical vertices.

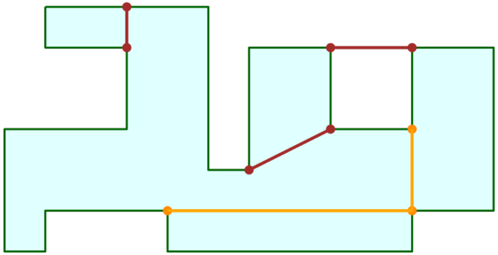


Figure 2: Examples of chords (orange) and **not** chords (red).

Chord-less polygons

Concave vertices observations

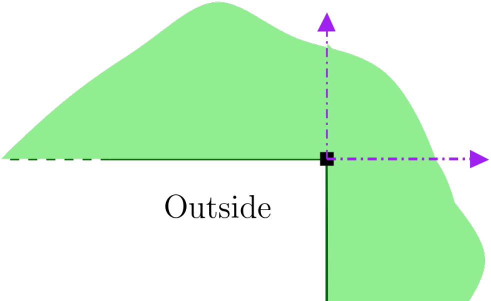
Suppose, for now, that there are no chords in polygon P.

Observation #1

P is a rectangle $\iff$ P has exactly four convex vertices.

Observation #2

A concave vertex must be a vertex of two rectangles of the partition. See example below.



Algorithm for chord-less polygons

Observation #3

If P is not a rectangle then it must have at least one concave vertex.

Idea to solve the problem

For each concave vertex, extend its vertical edge until it hits another edge. *If it hits a vertex, then that extension is a chord!*

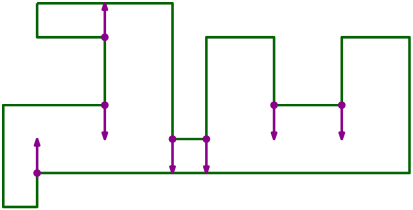


Figure 4: Chord-less polygons optimal solution.

Algorithm for chord-less polygons

Algorithm 1 Chord-less MNC in $O(n \log n)$

procedure $\mathrm{MNC}(P)$

$E \leftarrow \emptyset$; $R \leftarrow \emptyset$

$L \leftarrow$ sorted vertices of P by x-coordinate[1].

for each vertex v in P **do**

Insert v to E in position, delete position if duplicate.

if v is concave **then**

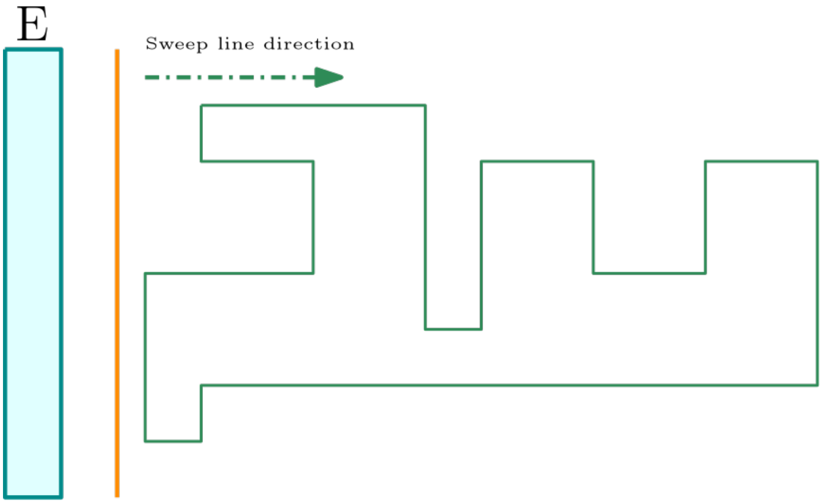
Extend vertical edge.

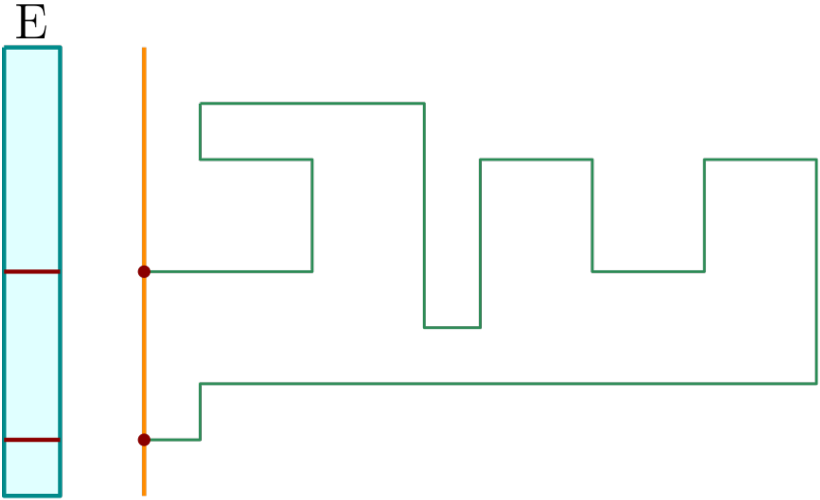
Insert to R the vertical edge.

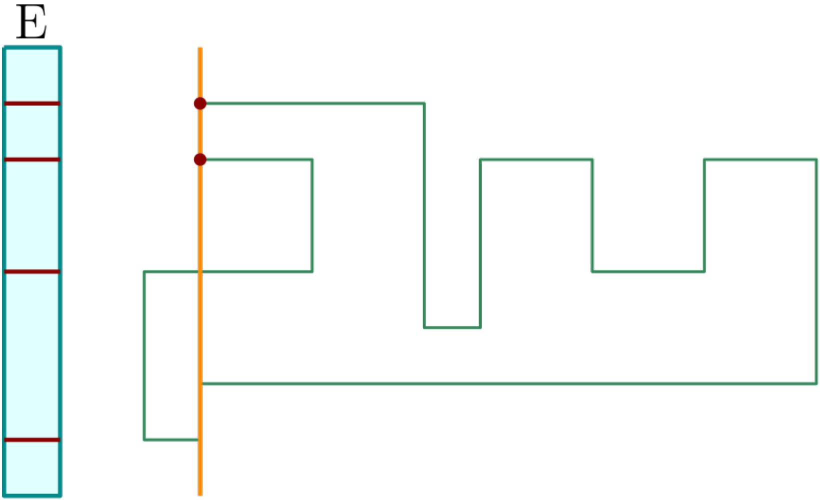
Return R as the paritioning edges of P.

[1] If using y-coordinate as tiebreaker, we can read pairs of vertices at a time (vertical edges).

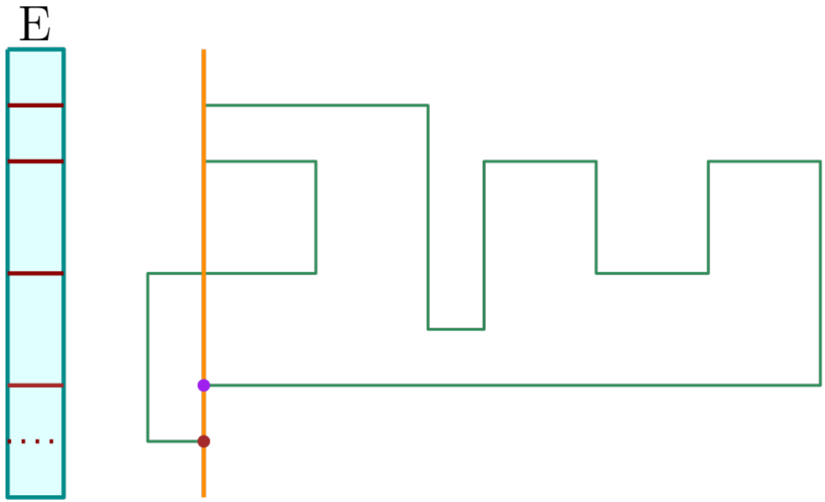
Algorithm for chord-less polygons



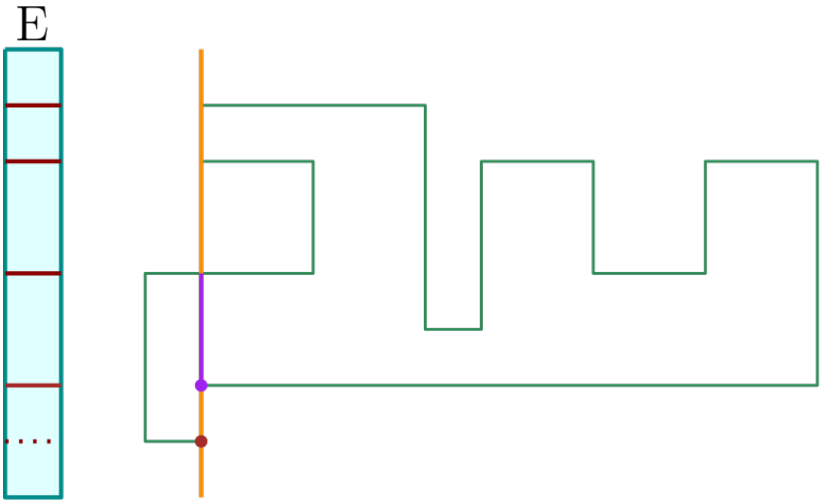

E


E

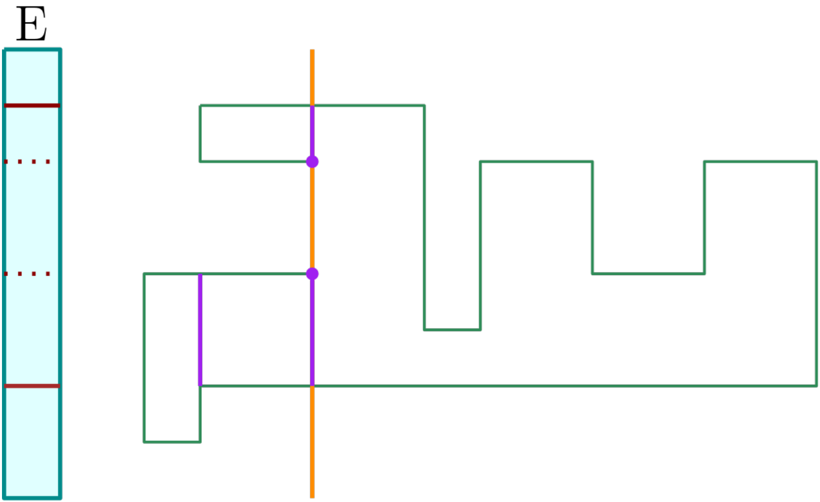
Algorithm for chord-less polygons



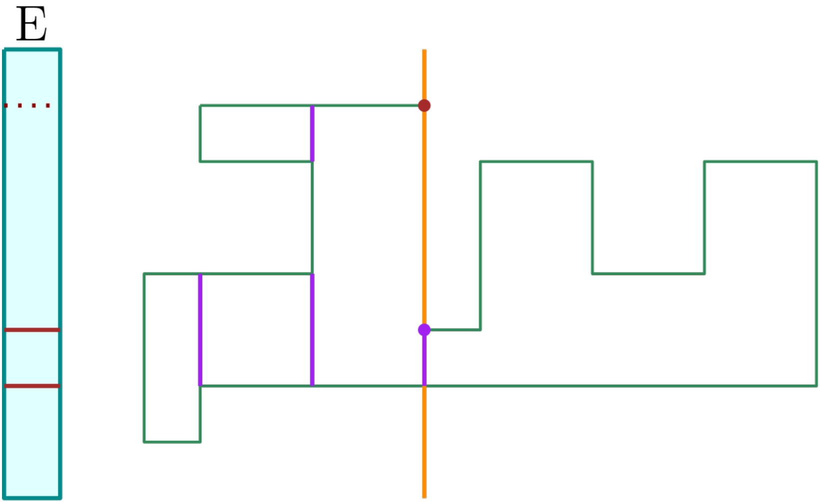
Algorithm for chord-less polygons



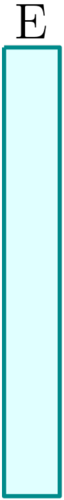
Algorithm for chord-less polygons

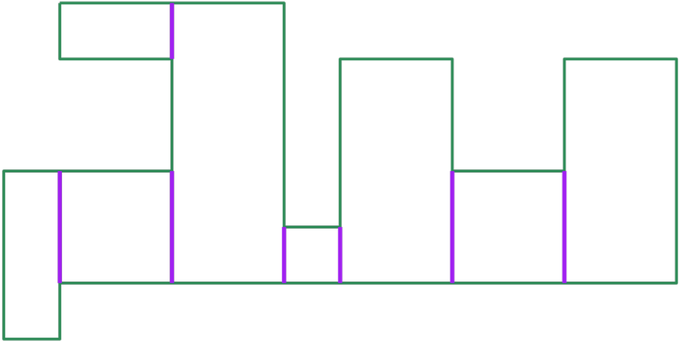


Algorithm for chord-less polygons

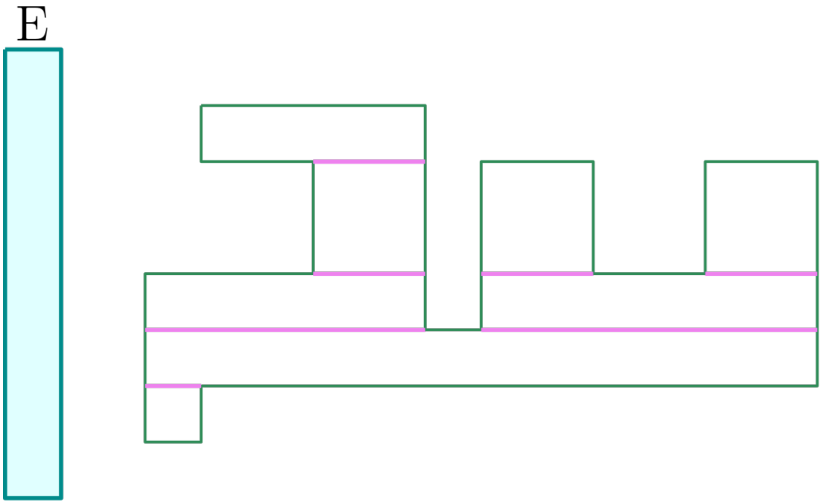


Algorithm for chord-less polygons





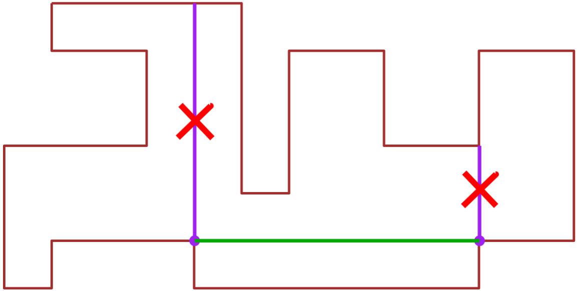
Algorithm for chord-less polygons



Polygons with chords

Problem #1

Suppose there are two co-horizontal vertices which form a chord. We may prefer joining that chord rather than extending the vertical edge.

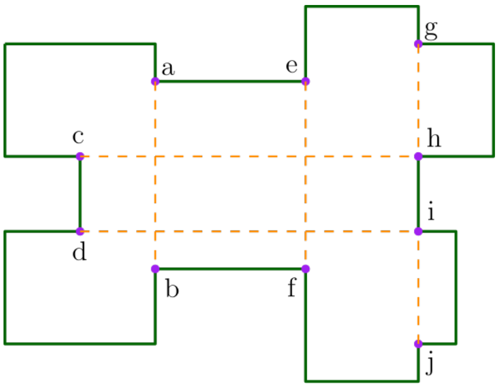


We would end up with one more rectangle!

Polygons with chords

Problem #2

If we drew all chords, there may be intersections.

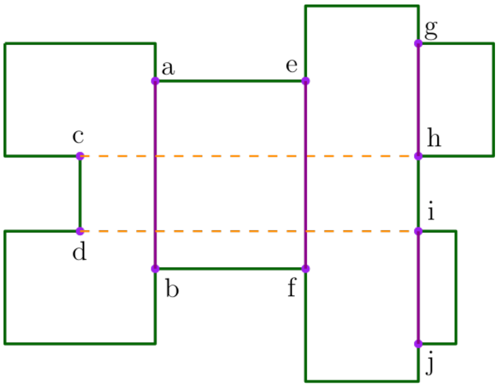


Chords *ab* and *ch* intersect. So do chords *ij* and *di*.

Polygons with chords

Solution

Draw the largest set of non-intersecting chords. After this step, no chords can remain. The remaining sub-polygons can be partitioned as usual.

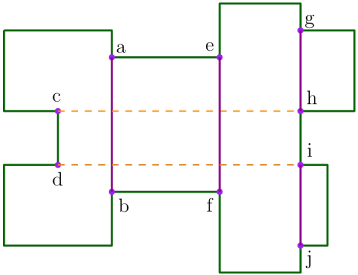


Polygons with chords

Theorem (Ferrari, 1984)

A rectilinear polygon R has a minimum partition of order $N - B + 1$, where

$N =$ Total number of concave vertices on the boundary of R.
$B =$ Maximum number of nonintersecting chords.



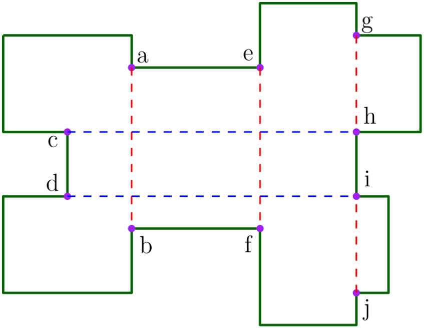
$N = 10$ and $B = 4$ therefore the optimal order is 7.

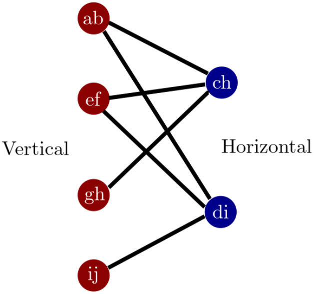
Finding the largest set of nonintersecting chords

Process

Construct a bipartite graph B, with vertices v_i for vertical chords, h_i for horizontal chords, and edges $\{v_i, h_j\}$ if v_i and h_j intersect.

Problem reduces to finding the maximum independent set of B.





Finding the largest set of nonintersecting chords

Maximum Independent Set

Recall that the MIS problem for a general graph is NP-hard. It is possible to find the MIS of a **bipartite** graph in polynomial time using Kőnig's theorem.

Kőnig's theorem

In any bipartite graph, the number of edges in a maximum matching (max. set of non-adjacent edges) equals the number of vertices in a minimum vertex cover.

Min Vertex Cover / Max Independent Set

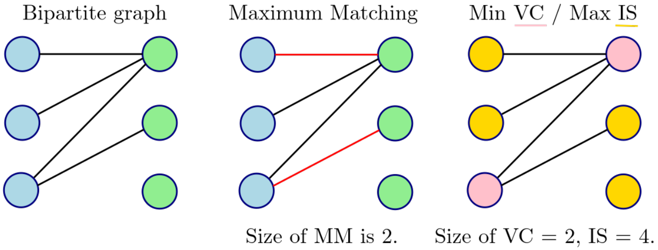
Both problems are complements of each other. So, the number of edges in a max matching is $\#V$ minus the size of the MIS.

Max. Independent Set on Bipartite Graphs

We can find the maximum matching by converting the bipartite graph into a flow network and running Ford-Fulkerson, in $O(n^3)$.

Hopcroft-Karp algorithm is faster with $O(n^{2.5})$, being n the number of vertices.

Knowing the size of the MIS, computing the actual set can be accomplished in $\omega(n^{2.5})$ using a simple algorithm.



Recap

Algorithm - $O(n^{2.5})$

1. Find chords of R.

2. Construct the bipartite graph $B = (V, H, E)$ as follows: each vertex v_i in V corresponds to a vertical chord, every h_i in H to a horizontal chord, and each edge $v_i h_j$ in E corresponds to an intersection between v_i and h_i.

3. Find maximum matching M of B. $O(n^{2.5})$

4. Find maximum independent set S of B based on M, by using Kőnig's Theorem. Let $b = |S|$.

5. Draw b chords corresponding to S, dividing R into $b + 1$ subpolygons, with each subpolygon being chord-less.

6. A minimal partition of each subpolygon can be found by using the sweep-line algorithm provided in slide 8.

Holes

The problem can be solved if it has holes. In that case, the smallest number of rectangles in a rectangular partition of a nonsimply rectilinear polygon R is:

$$N - B + 1 - D, \text{ being}$$

$N =$ Total number of concave vertices contained inside R.

$B =$ Maximum number of nonintersecting chords in R.

$D =$ Number of holes in R.

Improvements and References

For a polygon with holes, the optimal is $\Omega(n \log n)$.

It is possible to solve the problem without constructing the graph in $O(n)$ (Liou et al, 1990).

The 3D version using orthogonal parallelepipeds is NP-complete.

References

- Liou et al, *Minimum Rectangular Partition Problem for Simple Rectilinear Polygons*, IEEE Trans. Computer-Aided Design, 1990.
- Ferrari et al, *Minimal Rectangular Partitions of Digitized Blobs*, Department of Electrical Engineering University of California, 1984.

Thanks for your attention!