Partitioning Rectilinear Polygons

Govind Mittal – 2014B4A7530P

done under
Dr. Krishnendra Shekhawat
Department of Mathematics
BITS Pilani, Pilani Campus

for the fullfillment
of
Design Project – MATH F367

2^{nd} semester, 2016 - 2017

The Problem

Partitioning the given rectilinear polygon into its minimum non-overlapping rectangular cover.

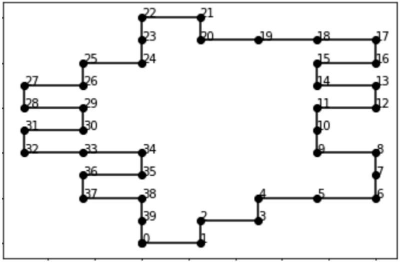


Figure: *Input:* Sample Rectilinear Polygon

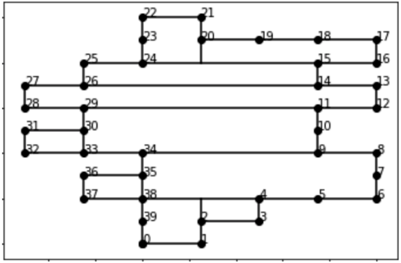


Figure: *Output:* Minimum Partitioned Rectilinear Polygon

Rectilinear Polygon

Rectilinear polygons are a simple connected single-cycle graph in $\mathbb{R} \times \mathbb{R}$, such that each of its edges is perpendicular or parallel to another one of its edge(s).

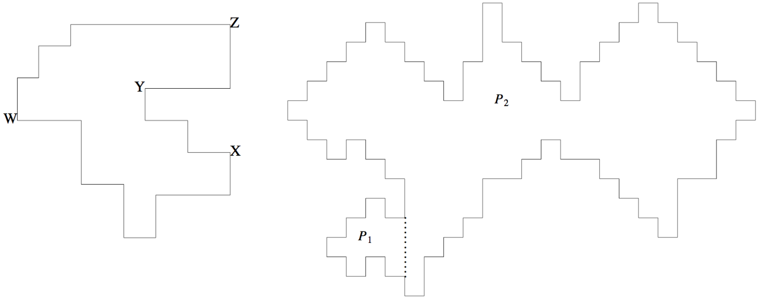
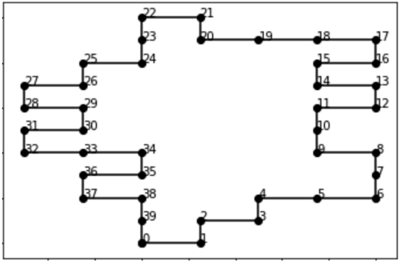


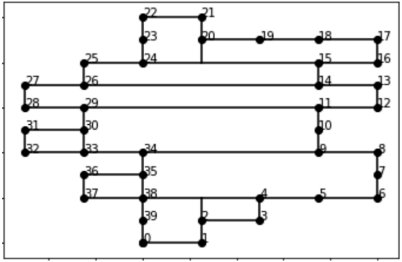
Figure: Examples of Rectilinear Polygon[1]

Aim

Partitioning any given rectilinear polygon into its minimum non-overlapping rectangular cover.



$\rightarrow$



What is minimum partition?

Partition of given rectillinear polygon into minimum number of **non-overlapping rectangles**, such that any two rectangles obtained after partitioning, if merged will not form a rectangle.

Literature Reviewed

To solve the problem I reviewed the following research publications

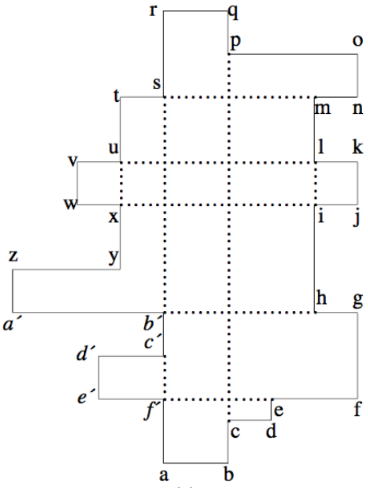
- *Wu, San-Yuan, and Sartaj Sahni.* "Fast algorithms to partition simple rectilinear polygons." VLSI Design 1.3 (1994): 193-215.
- *I Baybars, C M Eastman.* "Enumerating Architectural Arrangements by Generating Their Underlying Graphs." Environment and Planning B: Urban Analytics and City Science, Vol 7, Issue 3, pp. 289 - 310
- *S. Chaiken, D.J. Kleitman, M. Saks, and J. Shearer.* "Covering regions by rectangles." SIAM J. Alg. Disc. Meth., 2 (1981), pp. 394410.

So, I borrowed the idea for partitioning from the first paper.

In this work, I present the procedure for designing a prototype for partitioning of rectilinear polygons in a new way.

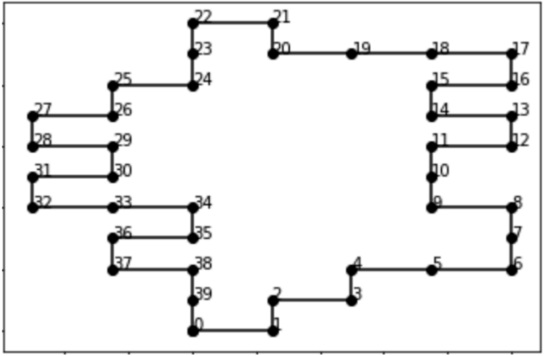
Definitions

- **Concave vertex** is a vertex, in which the two edges meeting at that point makes an angle of 270°, with the interior of polygon. In the adjacent figure, {c,h,i,m,p,s} are concave vertices
- **Chord:** A chord is a line joining any two co-horizontal concave vertices. In the adjacent figure, {cp, ix, lu, sb'} are chords.



STEP I: Input

We take input of a rectilinear polygon from the cursor keys, i.e., up($\uparrow$), left($\leftarrow$), and right ($\rightarrow$). As input is read, the pointer proceeds forward and draws a rectilinear polygon with its trail. The labelling of the vertices starts from v_0 to v_{n-1}, and $v_0 = v_n$, where n is the number of vertices in the polygon.

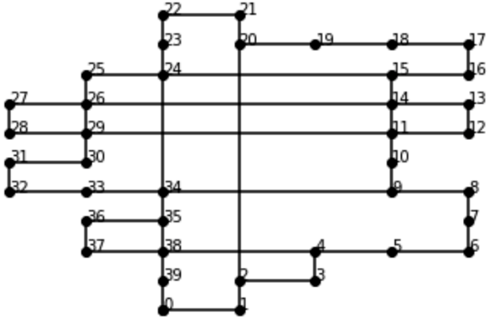


STEP II: Finding set of chords

Finding the set of horizontal chords and vertical chords.

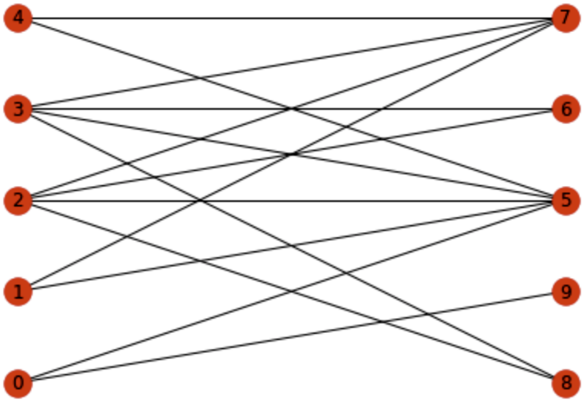
Horizontal_Chords $= \{(v_4, v_{38}), (v_9, v_{34}), (v_{11}, v_{29}), (v_{14}, v_{26}), (v_{15}, v_{24})\}$

Vertical_Chords $= \{(v_2, v_{20}), (v_{11}, v_{14}), (v_{24}, v_{34}), (v_{26}, v_{29}), (v_{35}, v_{38})\}$



STEP III: Compute Bipartite Graph

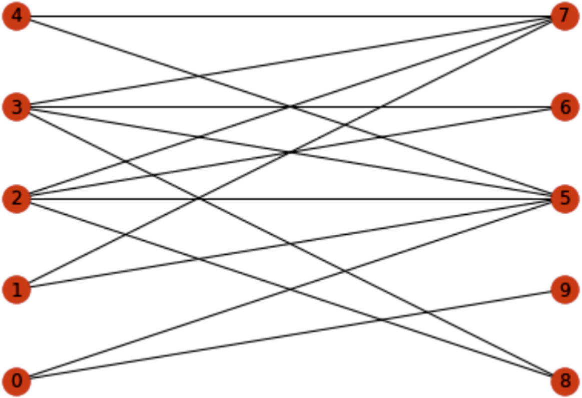
Compute the bipartite graph with the set of horizontal and vertical chords forming the partitions.



STEP IV: Finding maximum independent set

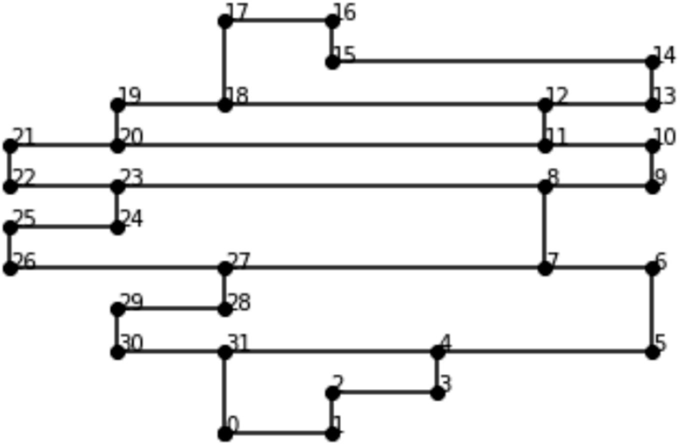
Find the set of vertices which form the maximum independent set of the bipartite graph

In this graph, the partition comes out to be $\{v_0, v_1, v_2, v_3, v_4\}$. Coincidentally, it is one of the partitions also!



STEP V: Plotting the max independent chords

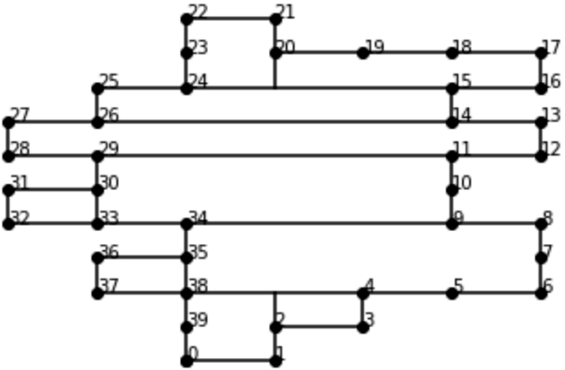
In this step, plot the corresponding indexed maximum independent set of chords. In the example, we have the complete set of horizontal chords as the set of maximum independent chords.



STEP VI: Completing the partition procedure

The final step of the partitioning is that from each of the concave vertices from which a chord was not drawn in *Step V*, draw a maximum length vertical line that is wholly within the smaller rectilinear polygon created in *Step V* that contains this vertex.

In this figure, the cardinality of partition $= 10$.



References and Resources I

- Sartaj Sahni, Wu, and San-Yuan
  Fast algorithms to partition simple rectilinear polygons.
  VLSI Design 1.3 (1994): 193-215.
- I Baybars, C M Eastman.
  Enumerating Architectural Arrangements by Generating Their Underlying Graphs.
  Environment and Planning B: Urban Analytics and City Science, Vol 7, Issue 3, pp. 289 - 310
- Python Networkx Library
- Author's Github Repository