# TAA: Practical Project 1

Dušan Hetlerović, Vincent Lopes

April 2019

## 1 Polygon generation

Our algorithm is an iterative one, adding one vertex per iteration until we have the number of points specified in input. For each iteration, here are the steps followed by our algorithm :

1 - choose a random free point P
2 - find all edges visible from P
3 - if no such edge, remove P from list and go back to step 1
4 - otherwise pick a random visible edge AB
5 - replace the edge AB with AP and PB
6 - mark all points inside APB as occupied
7 - if P is a border point, expand the grid (if possible, that is to say if the grid does not already occupy the max space allowed for it)

## 2 Complexity

We can now study the complexity of the algorithm described above. With $n$ being the number of vertices and $m$ the max grid dimension, we have the following steps:

- choosing a free point: $O(m^2)$, though in the vast majority of cases it's $\Theta(1)$
- checking edge visibility: $O(n^2)$ (for each of the $n$ edges checks visibility through other $n$ edges)
- choose a visible edge: $O(1)$
- insert point to sequence: $O(1)$
- marking points inside new triangle as occupied: $O(m^2)$ (although it only checks the boundary box of the triangle)
- grid expansion: $O(nm)$

Finally, n iterations give us a final complexity in $O(m^2n^3)$, but in practice the coefficient is rather small as both $n$ and $m$ expand only gradually over the course of the generation.

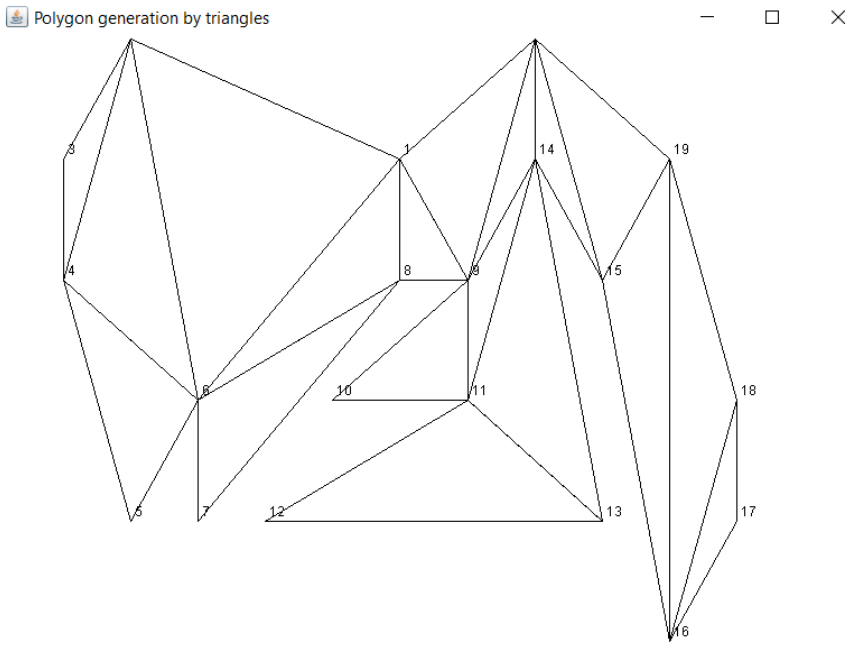


Figure 1: A generated polygon