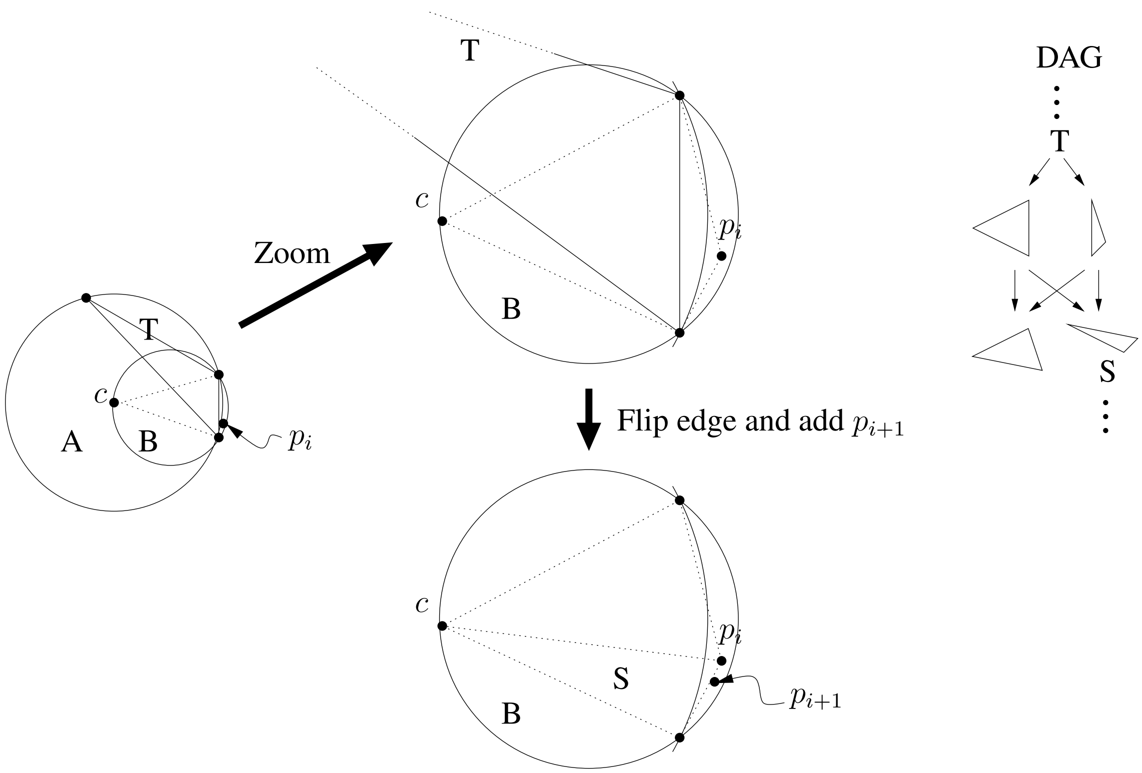
WORST-CASE TIME COMPLEXITY OF DELAUNAY REFINEMENT

The goal is to describe an example where Delaunay refinement takes O$(m^2 + n^2)$ time. This algorithm is based on the incremental Delaunay method and requires a history DAG. The worst-case performance of this DAG is O(n), and that occurs when the DAG has height O(n). This will happen when all the new triangles are created (at each increment) at about the same location. I will construct a data set with exponentially decreasing feature size to introduce triangles in the same vicinity in a manner that will result in a O(n) history DAG.

How to create the input set of n points

Instead of giving actual n points, I will show how each of the n points can be chosen (based on the other ones) so that performance is bad. Observe the behavior of the algorithm in the following bad case.



Triangle T is a bad triangle. A and B refer to the large and small circles, respectively. T's circumcenter c will be added during refinement, resulting in the (dashed) triangle whose circumcircle is B. However, there is a point p_i that lies within B and not A. Hence the new (dashed) triangle is not Delaunay, and an edge-flip will be required. The picture on the right shows a zoomed portion of the interesting part of the left picture before and after the edge flip. Notice that after the edge flip, the point p_i forms another bad triangle S. Another point, p_{i+1}, can be added close to S so that the situation shown in the left figure repeats for S as it did for T. Hence, n points can be determined in sequence such that they successively form bad triangles after introduction of Steiner points. This means that a new Steiner point is added for every additional point (like a) we add to the triangulation. So, $m = \Theta(n)$.

History DAG complexity

Notice that when the circumcenter c of T was added, pointers were added from T to the newly created triangles in the history DAG. After the subsequent edge flip, creating the bad triangle S, pointers were added from the previous triangles to this one. Hence, S will be a grandchild of T. Repeating the process on S results in a history DAG whose length is proportional to the number of points encountered along the way. Hence, the performance for the DAG per addition of a new Steiner point is $\mathrm{O}(m)$. This results in the quadratic time complexity $\mathrm{O}(n^2) = \mathrm{O}(m^2)$.