Klaara Huima
June 2025

Assignment 2

The project is run on my personal computer (2 years old, i7 processor with 14 cores).

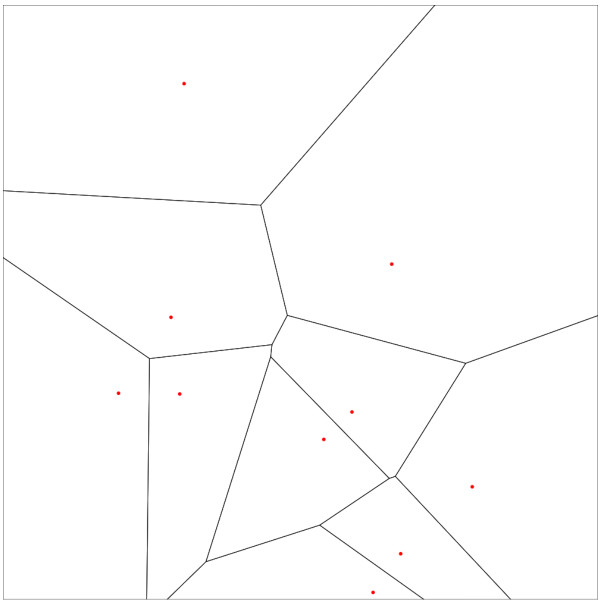
Naïve Voronoi Diagrams

To implement Voronoi Parallel Linear Enumeration with the Sutherland-Hodgman polygon clipping algorithm, I defined the following objects:

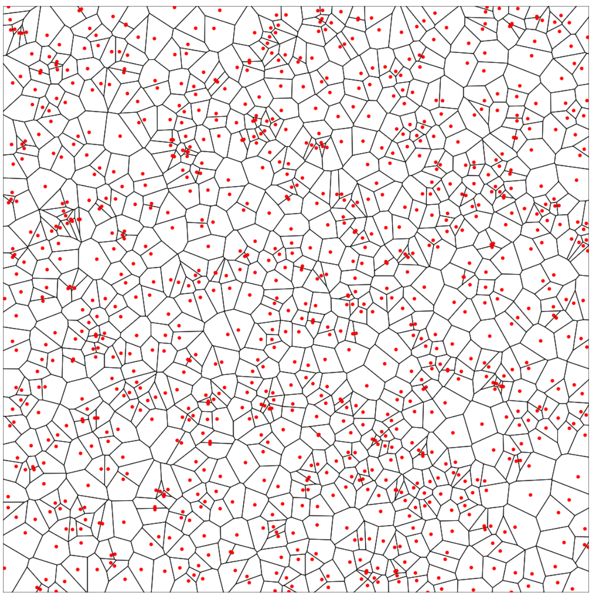
- *Polygon*, defined by a vector of 2-dimensional vertices
- *Voronoi* (diagram), defined by a vector of seed points and a vector of Polygons (every polygon is associated with a seed point)
- *Voronoi::clip_by_bisector*() executes the Sutherland-Hodgman clipping algorithm on one cell
- *Voronoi::compute*() computes the entire Voronoi diagram

We initialize N random points on a canvas with coordinates in $[0, 1] \times [0, 1]$ and run the algorithm, generating a set of convex polygons such that there is exactly one seed point in every polygon. In the photos below, the seed points are visualized in red.

N = 10 *(Elapsed time: <1 ms)*



N = 1000 *(Elapsed time: 1767 ms)*

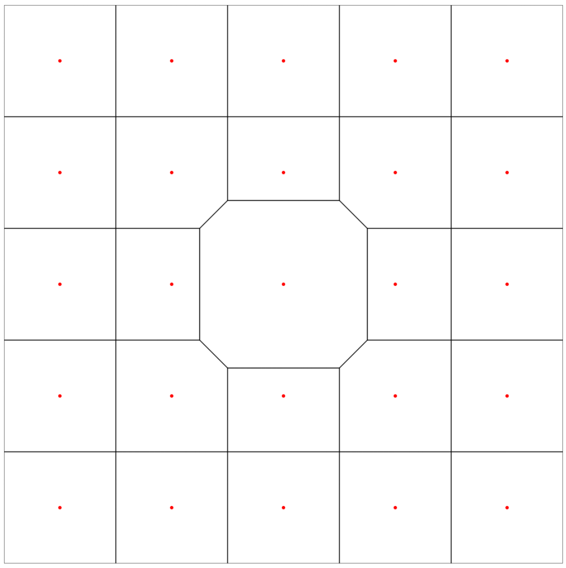


Power Diagrams

We then modify the Voronoi class to accommodate power diagrams. We add an associated weight to every polygon, and modify the functions compute() and clip_by_bisector() to take into account the cells' weights when building the diagram.

In the example below, I initialized a 5 x 5 grid of seed points. The center cell is given weight 1.02 and every other cell has weight 1.00. We notice that the increased weight results in a larger associated polygon.

N = 25, Non-uniform weight distribution *(Elapsed time: 242 ms)*

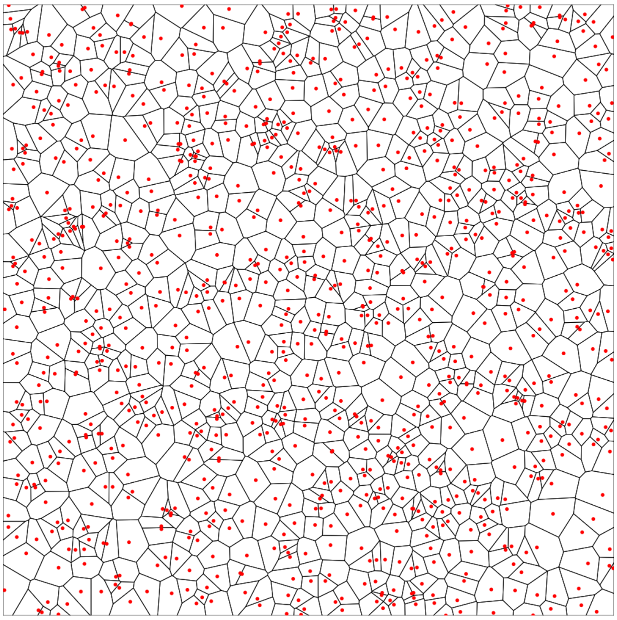


L-BFGS Optimization

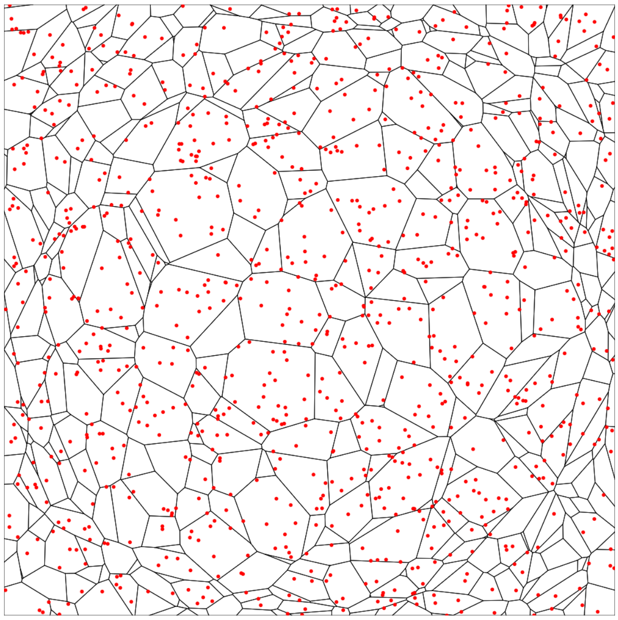
We then make use of the *libLBFGS* library to incorporate semi-discrete optimal transport for building the power-diagram. To compute the optimal power diagram associated with a given target area distribution, we use L-BFGS between a uniform density on the unit square and a discrete set of Dirac masses (the sites of the power diagram). I define an *OptimalTransport* object, which includes a Voronoi diagram and an *optimize*() function, executing the optimization with respect to target areas.

To replicate the example from the textbook, I generated 1000 points uniformly in the unit square and defined a target area for each point such that they follow a Gaussian shape centered in the square. That is, for a point at position x, the target area of the associated cell is proportional to $exp(-\ ||x\ -\ (0.5,\ 0.5)||^2/0.08)$.

N = 1000, Gaussian target area distribution with optimal transport *(Elapsed time: 25457 ms)*



i) Before optimal transport



ii) After optimal transport