Separators of Delaunay triangulations

Jonathan MacKenzie* Evan Thorpe†

January 26, 2019

1 Problem Description

Our chosen problem is a combination of two problems and their algorithms, namely *Delaunay triangulation* in $\mathbb{R}^2$ and the *Planar Separator theorem* of Lipton and Tarjan [3]. The former triangulates (connects into triangles) points in $\mathbb{R}^2$ such that no point is contained in the circumscribed circle of any triangle in the triangulation.[1] Given a set of points in $\mathbb{R}^2$, a Delaunay triangulation can be found in $O(n \log n)$ time. Lipton and Tarjan's separator theorem states that for any planar graph, one can always partition the graph into two disjoint connected subgraphs, each of size no more than $2n/3$, by removing at most $2\sqrt{2}\sqrt{n}$ vertices. These vertices to remove can be found in $O(n)$ time [3].

The question, as posed by Stefan Langerman in [4], is whether or not such an $O(\sqrt{n})$ separator can be found for a Delaunay triangulation in $O(n)$ time given only the set of points in $\mathbb{R}^2$.

Practical applications of such an algorithm are not immediately obvious, but Delaunay triangulation is heavily used in graphics because it avoids creating "sliver triangles" (triangles with one or two very acute angles) and the ability to split a graph into roughly equal-sized partitions in $O(n)$ time at the cost of $O(\sqrt{n})$ vertices can be very useful for recursive divide-and-conquer algorithms.

2 Background/Related Work

Delaunay triangulations are very well-known constructs in computational geometry. Two main approaches exist to find them, both of which are $O(n \log n)$ in time. The first is an incremental approach, as described in [1], the second is a divide-and-conquer approach, as described in [2].

It should be noted that both of these papers refer to Voronoi diagrams in their titles, and not Delaunay triangulations. A Voronoi diagram partitions the plane around points in the plane such that each section contains one point

*Department of Math and Computing Science, Saint Mary's University
†Department of Math and Computing Science, Saint Mary's University

[1] A delaunay triangulation can also be defined as a triangulation such that for any triangles sharing an edge, their opposing angles add to at least π

and is exactly the region consisting of all points closest to that point than any other. The Voronoi diagram of a set of points is dual to the Delaunay triangulation of those points, and in fact can be constructed by connecting the centres of the circumcircles of each triangle in the triangulation. Thus, the aforementioned papers give algorithms to find Voronoi diagrams by first finding Delaunay triangulations.

Lipton and Tarjan's separator theorem, as previously mentioned, provides an algorithm to separate a graph into disjoint connected subgraphs, each containing no more than 2/3 of the original vertices, at the cost of $O(\sqrt{n})$ vertices.

The problem of finding a $O(\sqrt{n})$ separator of the Delaunay triangulation of a set of points in linear time is currently an open problem, posed in [4].

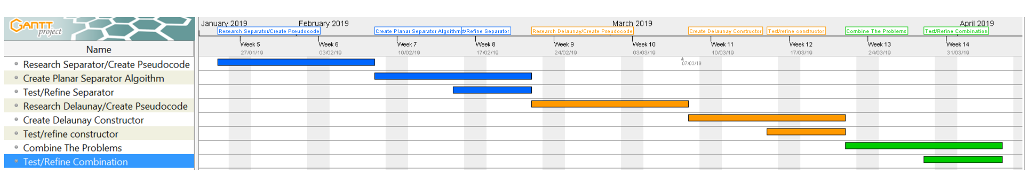
3 Methodology and Plan

Our first goal will be to write a program that implements Delaunay triangulation and the separator algorithm, most likely in C++. This will serve to familiarise us with both algorithms (we will probably end up using the divide-and-conquer algorithm from [2] as it is faster, and as our problem involves partitioning into two roughly-equal parts we may be able to take advantage of its recursive nature) and with whichever libraries we will end up using. This program will essentially be a concatenation of the two algorithms, and as Delaunay triangulation is $O(n\log n)$ and the separator algorithm is $O(n)$ it should run in $O(n\log n)$ time.

To reach this goal, we will need to understand both algorithms well enough to implement them. This will require research and planning, which has already begun: we've been reading articles on Delaunay triangulation and developing pseudocode for the separator algorithm as we will implement it.

Once we have this prototype, we will begin the merging of the two algorithms, with the hope of reaching an $O(n)$ algorithm. This will be done mostly in theory; we will analyse both algorithms, see what assumptions we can make about the input and output and which steps we can remove or shorten. The goal here will be to avoid computing the entire Delaunay triangulation while still having enough information to properly carry out the separation. Some optimisations may also be made directly in the code, with the theoretical implications extracted out afterwards.

If we are unable to find an $O(n)$ algorithm, or we prove it impossible, we hope to construct an algorithm that, while still $O(n\log n)$, is still faster in practice than the simple concatenation of the two algorithms. In this case, we plan to perform empirical testing on the runtime of our algorithm vs the simple concatenation.



References

[1] L. J. Guibas, D. E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Algorithmica*, 7, 06 1992.

[2] L. J. Guibas and J. Stolfi. Primitives for the manipulation of general subdivisions and the computation of Voronoi diagrams. *Proceedings of the fifteenth annual ACM symposium on Theory of computing*, 1983.

[3] R. J. Lipton and R. E. Tarjan. A separator theorem for planar graphs. *SIAM Journal on Applied Mathematics*, 36, 1979.

[4] J. ORourke. Open problems from CCCG 2017. *Canadian Conference on Computational Geometry*, 2018.