A Simple Way to Place a Point to Maximize Angles

Boris Aronov* aronov@poly.edu
Mark Yagnatinsky† myag@cis.poly.edu

Polytechnic Institute of NYU

Abstract

Given a set P of n points in the plane in general position, we seek to place a new point q in the interior of the convex hull of P, so that the Delaunay triangulation of $P \cup \{q\}$ has the largest possible minimum angle. The running time of our algorithm is $O(n^{2+\varepsilon})$, which is faster than the cubic time of the best previously known algorithm. It slows down to $O(n^{3+\varepsilon})$ if we also specify a set of non-crossing segments with endpoints in P and insist that the triangulation respect these segments, i.e., is the *constrained* Delaunay triangulation of the point and segments.

The previous algorithm could handle both the constrained and unconstrained version of the problem in (near-)cubic time. However, the current algorithm has another advantage over the previous one: it is simple to the point of cuteness.

1 Introduction

Inspired by the work in meshing on improving triangulations to keep the angles large, Aronov, Asano, and Funke [AAF10] asked the following question: Given a set P of n points in the plane, in general position, and a set C of non-crossing segments with endpoints in P, how do you optimally place a new point q in the interior of the convex hull of P, so that the constrained Delaunay triangulation of $P \cup \{q\}$ and C maximizes the measure of its smallest angle? The problem is attacked in [AAF10] by applying the envelope approach [SA95] to a natural set of bivariate functions, *after* partitioning the hull into $\Theta(n^2)$ regions.

In a follow-up work [AY13], we have refined the approach of [AAF10] by observing, roughly, that in every region of the above partition, one can solve the problem by invoking an LP-type problem solver and avoid computing bivariate envelopes. This sped up the algorithm to $O(n^3)$ in the unconstrained case (the constrained case is not explained in detail in [AAF10], but can be handled in near cubic time as well). The algorithm is randomized and the running time bound is expected, rather than worst case.

In this note we solve the unconstrained problem in time $O(n^{2+\varepsilon})$ and the constrained problem in time $O(n^{3+\varepsilon})$ deterministically. We achieve this by going back to the bivariate envelope tools used in [AAF10] and observing that the partitioning of the hull into $\Theta(n^2)$ regions is unnecessary. We do not require any partitioning at all in the unconstrained case and get away with $\Theta(n)$ regions in the constrained case, thereby achieving the claimed speedup.

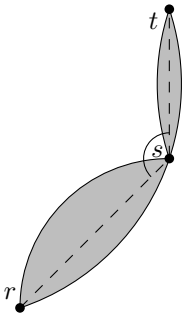
2 The unconstrained case

Let T be the Delaunay triangulation of P. If we place our new point q somewhere within T and re-triangulate, obtaining T_q, two things will happen: some edges of T will be eliminated and some new edges will appear. An old edge e goes away when we place q within its Delaunay lune L_e (that is, in the intersection of the two Delaunay disks through e). Expressing the condition for the appearance of new edges is a bit awkward, so we approach it indirectly via new triangles. A new triangle qrs will be present

*Research supported by NSF grants CCF-11-17336 and CCF-12-18791.

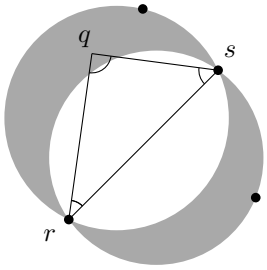
†Research supported by GAANN Grant P200A090157 from the US Department of Education and by NSF grant CCF-11-17336.

in T_q if and only if $e = rs$ is an edge of T, and q is in one, but not both, of e's Delaunay disks. (That is, if e is in their symmetric difference, S_e.) However, we are interested not in edges, but in angles. If $\triangle qrs$ is new, then it creates three new angles: $\angle qrs$, $\angle rsq$, and $\angle sqr$. An old angle rst disappears in one of two ways. Either one of its two bounding edges disappears, thus merging rst with a neighboring angle, or rst can be split by a new edge; we argue later that we need only handle the merging type of disappearance explicitly.



We define a (partial) function f_{rst} associated with each old angle rst. Its domain is the entire plane with the exception of $R = L_{rs} \cup L_{st}$, that is, with the exception of the region R (see figure to the right) such that placing q in R would cause $\angle rst$ to merge with a neighboring angle. The value of the function is simply the measure of the angle. We also define three (partial) functions associated with each edge rs of the old triangulation. We call them $f_{rs,r}$, $f_{rs,s}$, and $f_{rs,q}$. They all have the same domain S_e: the region of the plane where placing q would cause $\triangle qrs$ to appear (see figure below). The value of $f_{rs,r}(x, y)$ is the measure that $\angle qrs$ would have if q were placed at (x, y). Likewise, $f_{rs,s}$ measures $\angle qsr$, and $f_{rs,q}$ measures $\angle rqs$.

We are almost done. Each function can be constructed explicitly in constant time, given T. Let E be the lower envelope of these functions [SA95]. Notice that $E(x, y)$ is exactly the measure of the smallest angle in the Delaunay triangulation of $P \cup \{(x, y)\}$. As the functions are well behaved, the complexity of the envelope is $O(n^{2+\varepsilon})$, and we can compute it in that time [SA95, Theorem 7.16], and then apply standard tools to find its maximum, to identify the optimal placement for q, as claimed.



For correctness of the above algorithm, we need to argue that indeed the minimum of all functions defined at (x, y) is the measure of the smallest angle in the Delaunay triangulation of $P \cup \{(x, y)\}$. This would be trivial if the domain of each angle function corresponded precisely to the existence of the angle in the new triangulation. For the new angle functions, this is true: they are defined if and only if the corresponding new angle exists. For the old angle functions, the implication only holds in one direction: if an old angle exists, then its angle function is defined. If it doesn't exist, its function may or may not be defined. In particular, the function remains defined if the angle was split in two by a new edge, but both of its bounding segments remain in T_q. However, in that case, there are two new angles, both of whose measure is smaller than that of the old one, and hence the old angle is blocked from appearing on the lower envelope.

3 The constrained case

We now turn to the constrained case, where in addition to the point set P, we are given a set of constraint segments C that must be present in T_q, which would allow us, for instance, to triangulate a simple non-convex polygon, which was, in fact, the problem originally studied in [AAF10]; the algorithm from [AY13] can also be extended to the constrained case with only a slight slowdown.

We only sketch (due to lack of space) the required changes to the algorithm from section 2: besides replacing unconstrained Delaunay triangulations with constrained ones everywhere, the major change is that the functions f no longer have constant-complexity domains. To deal with this complication, we restrict the problem to each triangle Δ of T, compute the restriction of each function f to Δ, construct the envelope of the restrictions, and find its highest point, all in $O(n^{2+\varepsilon})$ time. Since this process needs to be repeated for every triangle, the total running time increases to $O(n^{3+\varepsilon})$, as promised.

References

[AAF10] B. Aronov, T. Asano, and S. Funke. Optimal triangulations of points and segments with Steiner points. *Int. J. Comput. Geom. Appl.*, 20(1) 89–104, 2010.

[AY13] B. Aronov and M. Yagnatinsky. How to place a point to maximize angles. *CCCG* 2013, pages 259–263.

[C89] L.P. Chew, Constrained Delaunay triangulations. *Algorithmica*, 4(1–4) 97–108, 1989.

[SA95] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. 1995.