Chapter 8

The Configuration Space Framework

In Section 7.1, we have discussed the incremental construction of the Delaunay triangulation of a finite point set. In this lecture, we want to analyze the runtime of this algorithm if the insertion order is chosen *uniformly at random* among all insertion orders. We will do the analysis not directly for the problem of constructing the Delaunay triangulation but in a somewhat more abstract framework, with the goal of reusing the analysis for other problems.

Throughout this lecture, we again assume general position: no three points on a line, no four on a circle.

8.1 The Delaunay triangulation — an abstract view

The incremental construction constructs and destroys triangles. In this section, we want to take a closer look at these triangles, and we want to understand exactly when a triangle is "there".

Lemma 8.1 *Given three points* $p, q, r \in R$*, the triangle* $\Delta(p, q, r)$ *with vertices* p, q, r *is a triangle of* $\mathcal{DT}(R)$ *if and only if the circumcircle of* $\Delta(p, q, r)$ *is empty of points from* R*.*

Proof. The "only if" direction follows from the definition of a Delaunay triangulation (Definition 6.8). The "if" direction is a consequence of general position and Lemma 6.17: if the circumcircle C of $\Delta(p, q, r)$ is empty of points from R, then all the three edges $\overline{pq}, \overline{qr}, \overline{pr}$ are easily seen to be in the Delaunay graph of R. C being empty also implies that the triangle $\Delta(p, q, r)$ is empty, and hence it forms a triangle of $\mathcal{DT}(R)$. □

Next we develop a somewhat more abstract view of $\mathcal{DT}(R)$.

Definition 8.2

(i) For all $p, q, r \in P$*, the triangle* $\Delta = \Delta(p, q, r)$ *is called a* configuration*. The points* p, q *and* r *are called the* defining elements *of* Δ*.*

(ii) *A configuration* Δ *is* in conflict *with a point* $s \in P$ *if* s *is strictly inside the circumcircle of* Δ*. In this case, the pair* (Δ, s) *is called a* conflict*.*

(iii) *A configuration* Δ *is called* active *w.r.t.* $R \subseteq P$ *if (a) the defining elements of* Δ *are in* R*, and (b) if* Δ *is not in conflict with any element of* R*.*

According to this definition and Lemma 8.1, $\mathcal{DT}(R)$ consists of exactly the configurations that are active w.r.t. R. Moreover, if we consider $\mathcal{DT}(R)$ and $\mathcal{DT}(R \cup \{s\})$ as sets of configurations, we can exactly say how these two sets differ.

There are the configurations in $\mathcal{DT}(R)$ that are not in conflict with s. These configurations are still in $\mathcal{DT}(R \cup \{s\})$. The configurations of $\mathcal{DT}(R)$ that *are* in conflict with s will be removed when going from R to $R \cup \{s\}$. Finally, $\mathcal{DT}(R \cup \{s\})$ contains some new configurations, all of which must have s in their defining set. According to Lemma 8.1, it cannot happen that we get a new configuration without s in its defining set, as such a configuration would have been present in $\mathcal{DT}(R)$ already.

8.2 Configuration Spaces

Here is the abstract framework that generalizes the previous configuration view of the Delaunay triangulation.

Definition 8.3 *Let* X *(the ground set) and* Π *(the set of configurations) be finite sets. Furthermore, let*

$$D : \Pi \to 2^X$$

be a function that assigns to every configuration Δ *a set of* defining elements $D(\Delta)$*. We assume that only a constant number of configurations have the same defining elements. Let*

$$K : \Pi \to 2^X$$

be a function that assigns to every configuration Δ *a set of elements* in conflict *with* Δ *(the "killer" elements). We stipulate that* $D(\Delta) \cap K(\Delta) = \emptyset$ *for all* $\Delta \in \Pi$*.*

Then the quadruple $\mathcal{S} = (X, \Pi, D, K)$ *is called a* configuration space*. The number*

$$d = d(\mathcal{S}) := \max_{\Delta \in \Pi} |D(\Delta)|$$

is called the dimension *of* $\mathcal{S}$*.*

Given $R \subseteq X$*, a configuration* Δ *is called* active *w.r.t.* R *if*

$$D(\Delta) \subseteq R \quad \textit{and} \quad K(\Delta) \cap R = \emptyset,$$

i.e. if all defining elements are in R *but no element of* R *is in conflict with* Δ*. The set of active configurations w.r.t.* R *is denoted by* $\mathcal{T}_{\mathcal{S}}(R)$*, where we drop the subscript if the configuration space is clear from the context.*

In case of the Delaunay triangulation, we set $X = P$ (the input point set). Π consists of all triangles $\Delta = \Delta(p, q, r)$ spanned by three points $p, q, r \in X \cup \{a, b, c\}$, where a, b, c are the three artificial far-away points. We set $D(\Delta) := \{p, q, r\} \cap X$. The set $K(\Delta)$ consists of all points strictly inside the circumcircle of Δ. The resulting configuration space has dimension 3, and the technical condition that only a constant number of configurations share the defining set is satisfied as well. In fact, every set of three points defines a *unique* configuration (triangle) in this case. A set of two points or one point defines three triangles (we have to add one or two artificial points which can be done in three ways). The empty set defines one triangle, the initial triangle consisting of just the three artificial points.

Furthermore, in the setting of the Delaunay triangulation, a configuration is active w.r.t. R if it is in $\mathcal{DT}(R \cup \{a, b, c\})$, i.e. we have $\mathcal{T}(R) = \mathcal{DT}(R \cup \{a, b, c\})$.

8.3 Expected structural change

Let us fix a configuration space $\mathcal{S} = (X, \Pi, D, K)$ for the remainder of this lecture. We can also interpret the incremental construction in $\mathcal{S}$. Given $R \subseteq X$ and $s \in X \setminus R$, we want to update $\mathcal{T}(R)$ to $\mathcal{T}(R \cup \{s\})$. What is the number of new configurations that arise during this step? For the case of Delaunay triangulations, this is the relevant question when we want to bound the number of Lawson flips during one update step, since this number is exactly the number of new configurations minus three.

Here is the general picture.

Definition 8.4 *For* $Q \subseteq X$ *and* $s \in Q$, $\deg(s, Q)$ *is defined as the number of configurations of* $\mathcal{T}(Q)$ *that have* s *in their defining set.*

With this, we can say that the number of new configurations in going from $\mathcal{T}(R)$ to $\mathcal{T}(R \cup \{s\})$ is precisely $\deg(s, R \cup \{s\})$, since the new configurations are by definition exactly the ones that have s in their defining set.

Now the random insertion order comes in for the first time: what is

$$E(\deg(s, R \cup \{s\})),$$

averaged over all insertion orders? In such a random insertion order, R is a random r-element subset of X (when we are about to insert the $(r+1)$-st element), and s is a random element of $X \setminus R$. Let $\mathcal{T}_r$ be the "random variable" for the set of active configurations after r insertion steps.

It seems hard to average over all R, but there is a trick: we make a movie of the randomized incremental construction, and then we watch the movie backwards. What we see is elements of X being deleted one after another, again in random order. This is due to the fact that the reverse of a random order is also random. At the point where the $(r+1)$-st element is being deleted, it is going to be a *random* element s of the currently present $(r+1)$-element subset Q. For fixed Q, the expected degree of s is simply the average degree of an element in Q which is

$$\frac{1}{r+1}\sum_{s\in Q}\deg(s,Q) \leqslant \frac{d}{r+1}|\mathcal{T}(Q)|,$$

since the sum counts every configuration of $\mathcal{T}(Q)$ at most d times. Since Q is a random $(r+1)$-element subset, we get

$$E(\deg(s, R\cup\{s\})) \leqslant \frac{d}{r+1}t_{r+1},$$

where t_{r+1} is defined as the expected number of active configurations w.r.t. a random $(r+1)$-element set.

Here is a more formal derivation that does not use the backwards movie view. It exploits the bijection

$$(R,s) \mapsto (\underbrace{R\cup\{s\}}_{Q}, s)$$

between pairs (R,s) with $|R| = r$ and $s \notin R$ and pairs (Q,s) with $|Q| = r+1$ and $s \in Q$. Let $n = |X|$.

$$\begin{aligned}
E(\deg(s, R\cup\{s\})) &= \frac{1}{\binom{n}{r}}\sum_{R\subseteq X, |R|=r}\frac{1}{n-r}\sum_{s\in X\setminus R}\deg(s, R\cup\{s\}) \\
&= \frac{1}{\binom{n}{r}}\sum_{Q\subseteq X, |Q|=r+1}\frac{1}{n-r}\sum_{s\in Q}\deg(s,Q) \\
&= \frac{1}{\binom{n}{r+1}}\sum_{Q\subseteq X, |Q|=r+1}\frac{\binom{n}{r+1}}{\binom{n}{r}}\frac{1}{n-r}\sum_{s\in Q}\deg(s,Q) \\
&= \frac{1}{\binom{n}{r+1}}\sum_{Q\subseteq X, |Q|=r+1}\frac{1}{r+1}\sum_{s\in Q}\deg(s,Q) \\
&\leqslant \frac{1}{\binom{n}{r+1}}\sum_{Q\subseteq X, |Q|=r+1}\frac{d}{r+1}|\mathcal{T}(Q)| \\
&= \frac{d}{r+1}t_{r+1}.
\end{aligned}$$

Thus, the expected number of new configurations in going from $\mathcal{T}_r$ to $\mathcal{T}_{r+1}$ is bounded by

$$\frac{d}{r+1}t_{r+1},$$

where t_{r+1} is the expected size of $\mathcal{T}_{r+1}$.

What do we get for Delaunay triangulations? We have $d = 3$ and $t_{r+1} \leqslant 2(r+4)-4$ (the maximum number of triangles in a triangulation of $r+4$ points. Hence,

$$E(\deg(s, R \cup \{s\})) \leqslant \frac{6r+12}{r+1} \approx 6.$$

This means that on average, ≈ 3 Lawson flips are done to update $\mathcal{DT}_r$ (the Delaunay triangulation after r insertion steps) to $\mathcal{DT}_{r+1}$. Over the whole algorithm, the expected update cost is thus $O(n)$.

8.4 Bounding location costs by conflict counting

Before we can even update $\mathcal{DT}_r$ to $\mathcal{DT}_{r+1}$ during the incremental construction of the Delaunay triangulation, we need to locate the new point s in $\mathcal{DT}_r$, meaning that we need to find the triangle that contains s. We have done this with the history graph: During the insertion of s we "visit" a sequence of triangles from the history graph, each of which contains s and was created at some previous iteration $k < r$.

However, some of these visited triangles are "ephemeral" triangles (recall the discussion at the end of Section 7.2), and they present a problem to the generic analysis we want to perform. Therefore, we will do a charging scheme, so that all triangles charged are valid Delaunay triangles.

The charging scheme is as follows: If the visited triangle Δ is a valid Delaunay triangle (from some previous iteration), then we simply charge the visit of Δ during the insertion of s to the triangle-point pair (Δ, s).

If, on the other hand, Δ is an "ephemeral" triangle, then Δ was destroyed, together with some neighbor Δ', by a Lawson flip into another pair Δ'', Δ'''. Note that this neighbor Δ' was a valid triangle. Thus, in this case we charge the visit of Δ during the insertion of s to the pair (Δ', s). Observe that s is contained in the circumcircle of Δ', so s is in conflict with Δ'.

This way, we have charged each visit to a triangle in the history graph to a triangle-point pair of the form (Δ, s), such that Δ is in conflict with s. Furthermore, it is easy to see that no such pair gets charged more than once.

We define the notion of a *conflict* in general:

Definition 8.5 *A* conflict *is a configuration-element pair (Δ, s) where $\Delta \in \mathcal{T}_r$ for some r and $s \in K(\Delta)$.*

Thus, the running time of the Delaunay algorithm is proportional to the number of conflicts. We now proceed to derive a bound on the expected number of conflicts in the generic configuration-space framework.

8.5 Expected number of conflicts

Since every configuration involved in a conflict has been created in some step r (we include step 0), the total number of conflicts is

$$\sum_{r=0}^{n} \sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} |K(\Delta)|,$$

where $\mathcal{T}_{-1} := \emptyset$. $\mathcal{T}_0$ consists of constantly many configurations only (namely those where the set of defining elements is the empty set), each of which is in conflict with at most all elements; moreover, no conflict is created in step n. Hence,

$$\sum_{r=0}^{n} \sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} |K(\Delta)| = O(n) + \sum_{r=1}^{n-1} \sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} |K(\Delta)|,$$

and we will bound the latter quantity. Let

$$K(r) := \sum_{\Delta \in \mathcal{T}_r \setminus \mathcal{T}_{r-1}} |K(\Delta)|, \quad r = 1, \ldots, n-1.$$

and $k(r) := E(K(r))$ the expected number of conflicts created in step r.

Bounding $k(r)$. We know that $\mathcal{T}_r$ arises from a random r-element set R. Fixing R, the backwards movie view tells us that $\mathcal{T}_{r-1}$ arises from $\mathcal{T}_r$ by deleting a random element s of R. Thus,

$$\begin{aligned} k(r) &= \frac{1}{\binom{n}{r}} \sum_{R \subseteq X, |R|=r} \frac{1}{r} \sum_{s \in R} \sum_{\Delta \in \mathcal{T}(R) \setminus \mathcal{T}(R \setminus \{s\})} |K(\Delta)| \\ &= \frac{1}{\binom{n}{r}} \sum_{R \subseteq X, |R|=r} \frac{1}{r} \sum_{s \in R} \sum_{\Delta \in \mathcal{T}(R), s \in D(\Delta)} |K(\Delta)| \\ &\leqslant \frac{1}{\binom{n}{r}} \sum_{R \subseteq X, |R|=r} \frac{d}{r} \sum_{\Delta \in \mathcal{T}(R)} |K(\Delta)|, \end{aligned}$$

since in the sum over $s \in R$, every configuration is counted at most d times. Since we can rewrite

$$\sum_{\Delta \in \mathcal{T}(R)} |K(\Delta)| = \sum_{y \in X \setminus R} |\{\Delta \in \mathcal{T}(R) : y \in K(\Delta)\}|,$$

we thus have

$$k(r) \leqslant \frac{1}{\binom{n}{r}} \sum_{R \subseteq X, |R|=r} \frac{d}{r} \sum_{y \in X \setminus R} |\{\Delta \in \mathcal{T}(R) : y \in K(\Delta)\}|.$$

To estimate this further, here is a simple but crucial

Lemma 8.6 *The configurations in $\mathcal{T}(R)$ that are not in conflict with y are the configurations in $\mathcal{T}(R \cup \{y\})$ that do not have y in their defining set; in formulas:*

$$|\mathcal{T}(R)| - |\{\Delta \in \mathcal{T}(R) : y \in K(\Delta)\}| = |\mathcal{T}(R \cup \{y\})| - \deg(y, R \cup \{y\}).$$

The proof is a direct consequence of the definitions: every configuration in $\mathcal{T}(R)$ not in conflict with y is by definition still present in $\mathcal{T}(R \cup \{y\})$ and still does not have y in its defining set. And a configuration in $\mathcal{T}(R \cup \{y\})$ with y not in its defining set is by definition already present in $\mathcal{T}(R)$ and already there not in conflict with y.

The lemma implies that

$$k(r) \leqslant k_1(r) - k_2(r) + k_3(r),$$

where

$$k_1(r) = \frac{1}{\binom{n}{r}} \sum_{R \subseteq X, |R|=r} \frac{d}{r} \sum_{y \in X \setminus R} |\mathcal{T}(R)|,$$

$$k_2(r) = \frac{1}{\binom{n}{r}} \sum_{R \subseteq X, |R|=r} \frac{d}{r} \sum_{y \in X \setminus R} |\mathcal{T}(R \cup \{y\})|,$$

$$k_3(r) = \frac{1}{\binom{n}{r}} \sum_{R \subseteq X, |R|=r} \frac{d}{r} \sum_{y \in X \setminus R} \deg(y, R \cup \{y\}).$$

Estimating $k_1(r)$. This is really simple.

$$\begin{aligned} k_1(r) &= \frac{1}{\binom{n}{r}} \sum_{R \subseteq X, |R|=r} \frac{d}{r} \sum_{y \in X \setminus R} |\mathcal{T}(R)| \\ &= \frac{1}{\binom{n}{r}} \sum_{R \subseteq X, |R|=r} \frac{d}{r}(n-r)|\mathcal{T}(R)| \\ &= \frac{d}{r}(n-r)t_r. \end{aligned}$$

Estimating $k_2(r)$. For this, we need to employ our earlier $(R, y) \mapsto (R \cup \{y\}, y)$ bijection again.

$$\begin{aligned}
k_2(r) &= \frac{1}{\binom{n}{r}} \sum_{R\subseteq X, |R|=r} \frac{d}{r} \sum_{y\in X\setminus R} |\mathcal{T}(R\cup\{y\})| \\
&= \frac{1}{\binom{n}{r}} \sum_{Q\subseteq X, |Q|=r+1} \frac{d}{r} \sum_{y\in Q} |\mathcal{T}(Q)| \\
&= \frac{1}{\binom{n}{r+1}} \sum_{Q\subseteq X, |Q|=r+1} \frac{\binom{n}{r+1}}{\binom{n}{r}} \frac{d}{r}(r+1)|\mathcal{T}(Q)| \\
&= \frac{1}{\binom{n}{r+1}} \sum_{Q\subseteq X, |Q|=r+1} \frac{d}{r}(n-r)|\mathcal{T}(Q)| \\
&= \frac{d}{r}(n-r)t_{r+1} \\
&= \frac{d}{r+1}(n-(r+1))t_{r+1} + \frac{dn}{r(r+1)}t_{r+1} \\
&= k_1(r+1) + \frac{dn}{r(r+1)}t_{r+1}.
\end{aligned}$$

Estimating $k_3(r)$. This is similar to $k_2(r)$ and in addition uses a fact that we have employed before: $\sum_{y\in Q} \deg(y, Q) \leqslant d|\mathcal{T}(Q)|$.

$$\begin{aligned}
k_3(r) &= \frac{1}{\binom{n}{r}} \sum_{R\subseteq X, |R|=r} \frac{d}{r} \sum_{y\in X\setminus R} \deg(y, R\cup\{y\}) \\
&= \frac{1}{\binom{n}{r}} \sum_{Q\subseteq X, |Q|=r+1} \frac{d}{r} \sum_{y\in Q} \deg(y, Q) \\
&\leqslant \frac{1}{\binom{n}{r}} \sum_{Q\subseteq X, |Q|=r+1} \frac{d^2}{r}|\mathcal{T}(Q)| \\
&= \frac{1}{\binom{n}{r+1}} \sum_{Q\subseteq X, |Q|=r+1} \frac{\binom{n}{r+1}}{\binom{n}{r}} \frac{d^2}{r}|\mathcal{T}(Q)| \\
&= \frac{1}{\binom{n}{r+1}} \sum_{Q\subseteq X, |Q|=r+1} \frac{n-r}{r+1} \cdot \frac{d^2}{r}|\mathcal{T}(Q)| \\
&= \frac{d^2}{r(r+1)}(n-r)t_{r+1} \\
&= \frac{d^2 n}{r(r+1)}t_{r+1} - \frac{d^2}{r+1}t_{r+1}.
\end{aligned}$$

Summing up. Let us recapitulate: the overall expected number of conflicts is $O(n)$ plus

$$\sum_{r=1}^{n-1} k(r) = \sum_{r=1}^{n-1} (k_1(r) - k_2(r) + k_3(r)).$$

Using our previous estimates, $k_1(2), \ldots, k_1(n-1)$ are canceled by the first terms of $k_2(1), \ldots, k_2(n-2)$. The second term of $k_2(r)$ can be combined with the first term of $k_3(r)$, so that we get

$$\begin{aligned}\sum_{r=1}^{n-1}(k_1(r) - k_2(r) + k_3(r)) &\leqslant k_1(1) - \underbrace{k_1(n)}_{=0} + n\sum_{r=1}^{n-1} \frac{d(d-1)}{r(r+1)} t_{r+1} - \sum_{r=1}^{n-1} \frac{d^2}{r+1} t_{r+1} \\ &\leqslant d(n-1)t_1 + d(d-1)n \sum_{r=1}^{n-1} \frac{t_{r+1}}{r(r+1)} \\ &= O\left(d^2 n \sum_{r=1}^{n} \frac{t_r}{r^2}\right).\end{aligned}$$

The Delaunay case. We have argued that the expected number of conflicts asymptotically bounds the expected total location cost over all insertion steps. The previous equation tells us that this cost is proportional to $O(n)$ plus

$$O\left(9n \sum_{r=1}^{n} \frac{2(r+4)-4}{r^2}\right) = O\left(n \sum_{r=1}^{n} \frac{1}{r}\right) = O(n \log n).$$

Here,

$$\sum_{r=1}^{n} \frac{1}{r} =: H_n$$

is the n-th Harmonic Number which is known to be approximately $\ln n$.

By going through the abstract framework of configuration spaces, we have thus analyzed the randomized incremental construction of the Delaunay triangulation of n points. According to Section 8.3, the expected update cost itself is only $O(n)$. The steps dominating the runtime are the location steps via the history graph. According to Section 8.5, all history graph searches (whose number is proportional to the number of conflicts) can be performed in expected time $O(n \log n)$, and this then also bounds the space requirements of the algorithm.

Exercise 8.7 *Design and analyze a sorting algorithm based on randomized incremental construction in configuration spaces. The input is a set S of numbers, and the output should be the sorted sequence (in increasing order).*

a) *Define an appropriate configuration space for the problem! In particular, the set of active configurations w.r.t. S should represent the desired sorted sequence.*

b) *Provide an efficient implementation of the incremental construction algorithm. "Efficient" means that the runtime of the algorithm is asymptotically dominated by the number of conflicts.*

c) *What is the expected number of conflicts (and thus the asymptotic runtime of your sorting algorithm) for a set S of n numbers?*

Questions

30. *What is a configuration space?* Give a precise definition! What is an active configuration?

31. *How do we get a configuration space from the problem of computing the Delaunay triangulation of a finite point set?*

32. *How many new active configurations do we get on average when inserting the* r-*th element?* Provide an answer for configuration spaces in general, and for the special case of the Delaunay triangulation.

33. *What is a conflict?* Provide an answer for configuration spaces in general, and for the special case of the Delaunay triangulation.

34. *Explain why counting the expected number of conflicts asymptotically bounds the cost for the history searches during the randomized incremental construction of the Delaunay triangulation!*