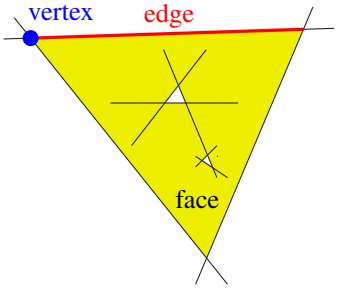
Chapter 21

Sampling and the Moments Technique

By Sariel Har-Peled, January 19, 2018[1]

> Sun and rain and bush had made the site look old, like the site of a dead civilization. The ruins, spreading over so many acres, seemed to speak of a final catastrophe. But the civilization wasn't dead. It was the civilization I existed in and in fact was still working towards. And that could make for an odd feeling: to be among the ruins was to have your time-sense unsettled. You felt like a ghost, not from the past, but from the future. You felt that your life and ambition had already been lived out for you and you were looking at the relics of that life. You were in a place where the future had come and gone.
>
> A bend in the river, V. S. Naipaul

21.1. Vertical decomposition



Given a set S of n segments in the plane, its ***arrangement***, denoted by $\mathcal{A}(\mathsf{S})$, is the decomposition of the plane into faces, edges, and vertices. The ***vertices*** of $\mathcal{A}(\mathsf{S})$ are the endpoints and the intersection points of the segments of S, the ***edges*** are the maximal connected portions of the segments not containing any vertex, and the ***faces*** are the connected components of the complement of the union of the segments of S. These definitions are depicted on the right.

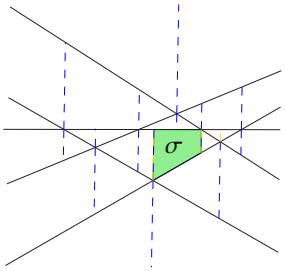
For numerical reasons (and also conceptually), a symbolic representation would be better than a numerical one. Thus, an intersection vertex would be represented by two pointers to the segments that their intersection is this vertex. Similarly, an edge would be represented as a pointer to the segment that contains it, and two pointers to the vertices forming its endpoints.

Naturally, we are assuming here that we have geometric primitives that can resolve any decision problem of interest that involve a few geometric entities. For example, for a given segment s and a point p, we would be interested in deciding if p lies vertically below s. From a theoretical point of view, all these primitives require a constant amount of computation, and are "easy". In the real world, numerical issues and degeneracies make implementing these primitives surprisingly challenging. We are going to ignore this major headache here, but the reader should be aware of it.

We will be interested in computing the arrangement $\mathcal{A}(\mathsf{S})$ and a representation of it that makes it easy to manipulate. In particular, we would like to be able to quickly resolve questions of the type (i) are two points in the same face?, (ii) can one traverse from one point to the other without crossing any segment?, etc. The naive representation of each face as polygons (potentially with holes) is not

conducive to carrying out such tasks, since a polygon might be arbitrarily complicated. Instead, we will prefer to break the arrangement into smaller canonical tiles.

To this end, a ***vertical trapezoid*** is a quadrangle with two vertical sides. The breaking of the faces into such trapezoids is the ***vertical decomposition*** of the arrangement $\mathcal{A}(\mathsf{S})$.



Formally, for a subset $\mathsf{R} \subseteq \mathsf{S}$, let $\mathcal{A}^{|}(\mathsf{R})$ denote the ***vertical decomposition*** of the plane formed by the arrangement $\mathcal{A}(\mathsf{R})$ of the segments of R. This is the partition of the plane into interior disjoint vertical trapezoids formed by erecting vertical walls through each vertex of $\mathcal{A}^{|}(\mathsf{R})$. Formally, a ***vertex*** of $\mathcal{A}^{|}(\mathsf{R})$ is either an endpoint of a segment of R or an intersection point of two of its segments. From each such vertex we shoot up (similarly, down) a vertical ray till it hits a segment of R or it continues all the way to infinity. See the figure on the right.

Note that a vertical trapezoid is defined by at most four segments: two segments defining its ceiling and floor and two segments defining the two intersection points that induce the two vertical walls on its boundary. Of course, a vertical trapezoid might be degenerate and thus be defined by fewer segments (i.e., an unbounded vertical trapezoid or a triangle with a vertical segment as one of its sides).

Vertical decomposition breaks the faces of the arrangement that might be arbitrarily complicated into entities (i.e., vertical trapezoids) of constant complexity. This makes handling arrangements (decomposed into vertical trapezoid) much easier computationally.

In the following, we assume that the n segments of S have k pairwise intersection points overall, and we want to compute the arrangement $\mathcal{A} = \mathcal{A}(\mathsf{S})$; namely, compute the edges, vertices, and faces of $\mathcal{A}(\mathsf{S})$. One possible way is the following: Compute a random permutation of the segments of S: $\mathsf{S} = \langle \mathsf{s}_1, \ldots, \mathsf{s}_n \rangle$. Let $\mathsf{S}_i = \langle \mathsf{s}_1, \ldots, \mathsf{s}_i \rangle$ be the prefix of length i of S. Compute $\mathcal{A}^{|}(\mathsf{S}_i)$ from $\mathcal{A}^{|}(\mathsf{S}_{i-1})$, for $i = 1, \ldots, n$. Clearly, $\mathcal{A}^{|}(\mathsf{S}) = \mathcal{A}^{|}(\mathsf{S}_n)$, and we can extract $\mathcal{A}(\mathsf{S})$ from it. Namely, in the ith iteration, we insert the segment s_i into the arrangement $\mathcal{A}^{|}(\mathsf{S}_{i-1})$.

This technique of building the arrangement by inserting the segments one by one is called ***randomized incremental construction***.

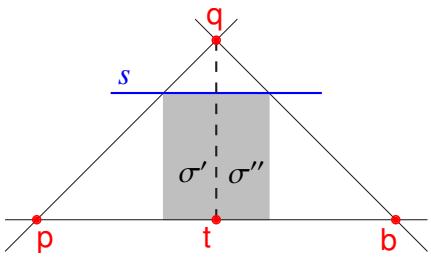
Who need these pesky arrangements anyway? The reader might wonder who needs arrangements? As a concrete examples, consider a situation where you are give several maps of a city containing different layers of information (i.e., streets map, sewer map, electric lines map, train lines map, etc). We would like to compute the overlay map formed by putting all these maps on top of each other. For example, we might be interested in figuring out if there are any buildings lying on a planned train line, etc.

More generally, think about a set of general constraints in $\mathbb{R}^d$. Each constraint is bounded by a surface, or a patch of a surface. The decomposition of $\mathbb{R}^d$ formed by the arrangement of these surfaces gives us a description of the parametric space in a way that is algorithmically useful. For example, finding if there is a point inside all the constraints, when all the constraints are induced by linear inequalities, is linear programming. Namely, arrangements are a useful way to think about any parametric space partitioned by various constraints.

21.1.1. Randomized incremental construction (RIC)

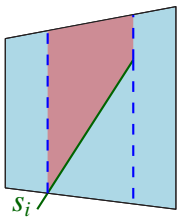
Imagine that we had computed the arrangement $\mathcal{B}_{i-1} = \mathcal{A}^{|}(\mathsf{S}_{i-1})$. In the ith iteration we compute $\mathcal{B}_i$ by inserting s_i into the arrangement $\mathcal{B}_{i-1}$. This involves splitting some trapezoids (and merging some others).

As a concrete example, consider the figure on the right. Here we insert s in the arrangement. To this end we split the "vertical trapezoids" $\triangle\mathsf{pqt}$ and $\triangle\mathsf{bqt}$, each into three trapezoids. The two trapezoids σ' and σ'' now need to be merged together to form the new trapezoid which appears in the vertical decomposition of the new arrangement. (Note that the figure does not show all the trapezoids in the vertical decomposition.)



To facilitate this, we need to compute the trapezoids of $\mathcal{B}_{i-1}$ that intersect s_i. This is done by maintaining a ***conflict graph***. Each trapezoid $\sigma \in \mathcal{A}^{\vert}(\mathsf{S}_{i-1})$ maintains a ***conflict list*** $\mathrm{cl}(\sigma)$ of *all* the segments of S that intersect its interior. In particular, the conflict list of σ cannot contain any segment of S_{i-1}, and as such it contains only the segments of $\mathsf{S} \setminus \mathsf{S}_{i-1}$ that intersect its interior. We also maintain a similar structure for each segment, listing all the trapezoids of $\mathcal{A}^{\vert}(\mathsf{S}_{i-1})$ that it currently intersects (in its interior). We maintain those lists with cross pointers, so that given an entry (σ, s) in the conflict list of σ, we can find the entry (s, σ) in the conflict list of s in constant time.

Thus, given s_i, we know what trapezoids need to be split (i.e., all the trapezoids in $\mathrm{cl}(\mathsf{s}_i)$). Splitting a trapezoid σ by a segment s_i is the operation of computing a set of (at most) four trapezoids that cover σ and have s_i on their boundary. We compute those new trapezoids, and next we need to compute the conflict lists of the new trapezoids. This can be easily done by taking the conflict list of a trapezoid $\sigma \in \mathrm{cl}(\mathsf{s}_i)$ and distributing its segments among the $O(1)$ new trapezoids that cover σ. Using careful implementation, this requires a linear time in the size of the conflict list of σ.



Note that only trapezoids that intersect s_i in their interior get split. Also, we need to update the conflict lists for the segments (that were not inserted yet).

We next sketch the low-level details involved in maintaining these conflict lists. For a segment s that intersects the interior of a trapezoid σ, we maintain the pair (s, σ). For every trapezoid σ, in the current vertical decomposition, we maintain a doubly linked list of all such pairs that contain σ. Similarly, for each segment s we maintain the doubly linked list of all such pairs that contain s. Finally, each such pair contains two pointers to the location in the two respective lists where the pair is being stored.

It is now straightforward to verify that using this data-structure we can implement the required operations in linear time in the size of the relevant conflict lists.

In the above description, we ignored the need to merge adjacent trapezoids if they have identical floor and ceiling – this can be done by a somewhat straightforward and tedious implementation of the vertical decomposition data-structure, by providing pointers between adjacent vertical trapezoids and maintaining the conflict list sorted (or by using hashing) so that merge operations can be done quickly. In any case, this can be done in linear time in the input/output size involved, as can be verified.

21.1.1.1. Analysis

Claim 21.1.1. *The (amortized) running time of constructing $\mathcal{B}_i$ from $\mathcal{B}_{i-1}$ is proportional to the size of the conflict lists of the vertical trapezoids in $\mathcal{B}_i \setminus \mathcal{B}_{i-1}$ (and the number of such new trapezoids).*

Proof: Observe that we can charge all the work involved in the ith iteration to either the conflict lists of the newly created trapezoids or the deleted conflict lists. Clearly, the running time of the algorithm in the ith iteration is linear in the total size of these conflict lists. Observe that every conflict gets charged twice – when it is being created and when it is being deleted. As such, the (amortized) running time in the ith iteration is proportional to the total length of the newly created conflict lists. ■

Thus, to bound the running time of the algorithm, it is enough to bound the expected size of the destroyed conflict lists in ith iteration (and sum this bound on the n iterations carried out by the algorithm). Or alternatively, bound the expected size of the conflict lists created in the ith iteration.

Lemma 21.1.2. *Let S be a set of n segments (in general position[2]) with k intersection points. Let S_i be the first i segments in a random permutation of S. The expected size of $\mathcal{B}_i = \mathcal{A}^{\vert}(\mathsf{S}_i)$, denoted by $\tau(i)$ (i.e., the number of trapezoids in $\mathcal{B}_i$), is $O\big(i + k(i/n)^2\big)$.*

Proof: Consider[3] an intersection point $\mathsf{p} = s \cap s'$, where $s, s' \in \mathsf{S}$. The probability that p is present in $\mathcal{A}^{\vert}(\mathsf{S}_i)$ is equivalent to the probability that both s and s' are in S_i. This probability is

$$\alpha = \frac{\binom{n-2}{i-2}}{\binom{n}{i}} = \frac{(n-2)!}{(i-2)!\,(n-i)!} \cdot \frac{i!\,(n-i)!}{n!} = \frac{i(i-1)}{n(n-1)}.$$

For each intersection point p in $\mathcal{A}(\mathsf{S})$ define an indicator variable X_{p}, which is 1 if the two segments defining p are in the random sample S_i and 0 otherwise. We have that $\mathbf{E}\big[X_{\mathsf{p}}\big] = \alpha$, and as such, by linearity of expectation, the expected number of intersection points in the arrangement $\mathcal{A}(\mathsf{S}_i)$ is

$$\mathbf{E}\left[\sum_{\mathsf{p}\in V} X_{\mathsf{p}}\right] = \sum_{\mathsf{p}\in V} \mathbf{E}\big[X_{\mathsf{p}}\big] = \sum_{\mathsf{p}\in V} \alpha = k\alpha,$$

where V is the set of k intersection points of $\mathcal{A}(\mathsf{S})$. Also, every endpoint of a segment of S_i contributed its two endpoints to the arrangement $\mathcal{A}(\mathsf{S}_i)$. Thus, we have that the expected number of vertices in $\mathcal{A}(\mathsf{S}_i)$ is

$$2i + \frac{i(i-1)}{n(n-1)}k.$$

Now, the number of trapezoids in $\mathcal{A}^{\vert}(\mathsf{S}_i)$ is proportional to the number of vertices of $\mathcal{A}(\mathsf{S}_i)$, which implies the claim. ■

21.1.2. Backward analysis

In the following, we would like to consider the total amount of work involved in the ith iteration of the algorithm. The way to analyze these iterations is (conceptually) to run the algorithm for the first i iterations and then run "backward" the last iteration.

So, imagine that the overall size of the conflict lists of the trapezoids of $\mathcal{B}_i$ is W_i and the total size of the conflict lists created only in the ith iteration is C_i.

We are interested in bounding the expected size of C_i, since this is (essentially) the amount of work done by the algorithm in this iteration. Observe that the structure of $\mathcal{B}_i$ is defined independently of

[2] In this case, no two intersection points of input segments are the same, no two intersection points (or vertices) have the same x-coordinate, no two segments lie on the same line, etc. Making the geometric algorithm work correctly for all degenerate inputs is a huge task that can usually be handled by tedious and careful implementation. Thus, we will always assume general position of the input. In other words, in theory all geometric inputs are inherently good, while in practice they are all evil (as anybody who tried to implement geometric algorithms can testify). The reader is encouraged not to use this to draw any conclusions on the human condition.

[3] The proof is provided in excruciating detail to get the reader used to this kind of argumentation. I would apologize for this pain, but it is a minor trifle, not to be mentioned, when compared to the other offenses in this book.

the permutation S_i and depends only on the (unordered) set $\mathsf{S}_i = \{\mathsf{s}_1, \ldots, \mathsf{s}_i\}$. So, fix S_i. What is the probability that s_i is a specific segment s of S_i? Clearly, this is $1/i$ since this is the probability of s being the last element in a permutation of the i elements of S_i (i.e., we consider a random permutation of S_i).

Now, consider a trapezoid $\sigma \in \mathcal{B}_i$. If σ was created in the ith iteration, then s_i must be one of the (at most four) segments that define it. Indeed, if s_i is not one of the segments that define σ, then σ existed in the vertical decomposition before s_i was inserted. Since $\mathcal{B}_i$ is independent of the internal ordering of S_i, it follows that $\mathbf{Pr}[\sigma \in (\mathcal{B}_i \setminus \mathcal{B}_{i-1})] \leq 4/i$. In particular, the overall size of the conflict lists in the end of the ith iteration is

$$W_i = \sum_{\sigma \in \mathcal{B}_i} |\operatorname{cl}(\sigma)|.$$

As such, the expected overall size of the conflict lists created in the ith iteration is

$$\mathbf{E}\left[C_i \,\middle|\, \mathcal{B}_i\right] \leq \sum_{\sigma \in \mathcal{B}_i} \frac{4}{i} |\operatorname{cl}(\sigma)| \leq \frac{4}{i} W_i.$$

By Lemma 21.1.2, the expected size of $\mathcal{B}_i$ is $O\left(i + ki^2/n^2\right)$. Let us guess (for the time being) that on average the size of the conflict list of a trapezoid of $\mathcal{B}_i$ is about $O(n/i)$. In particular, assume that we know that

$$\mathbf{E}\left[W_i\right] = O\left(\left(i + \frac{i^2}{n^2}k\right)\frac{n}{i}\right) = O\left(n + k\frac{i}{n}\right),$$

by Lemma 21.1.2, implying

$$\mathbf{E}\left[C_i\right] = \mathbf{E}\left[\mathbf{E}\left[C_i \,\middle|\, \mathcal{B}_i\right]\right] \leq \mathbf{E}\left[\frac{4}{i}W_i\right] = \frac{4}{i}\mathbf{E}\left[W_i\right] = O\left(\frac{4}{i}\left(n + \frac{ki}{n}\right)\right) = O\left(\frac{n}{i} + \frac{k}{n}\right), \tag{21.1}$$

using Lemma 21.7.2$_{\text{p16}}$. In particular, the expected (amortized) amount of work in the ith iteration is proportional to $\mathbf{E}\left[C_i\right]$. Thus, the overall expected running time of the algorithm is

$$\mathbf{E}\left[\sum_{i=1}^{n} C_i\right] = \sum_{i=1}^{n} O\left(\frac{n}{i} + \frac{k}{n}\right) = O\left(n \log n + k\right).$$

Theorem 21.1.3. *Given a set S of n segments in the plane with k intersections, one can compute the vertical decomposition of $\mathcal{A}(\mathsf{S})$ in expected $O(n \log n + k)$ time.*

Intuition and discussion. What remains to be seen is how we came up with the guess that the average size of a conflict list of a trapezoid of $\mathcal{B}_i$ is about $O(n/i)$. Note that using ε-nets implies that the bound $O((n/i)\log i)$ holds with constant probability (see Theorem 21.7.1$_{\text{p16}}$) for all trapezoids in this arrangement. As such, this result is only slightly surprising. To prove this, we present in the next section a "strengthening" of ε-nets to geometric settings.

To get some intuition on how we came up with this guess, consider a set P of n points on the line and a random sample R of i points from P. Let $\widehat{\mathcal{I}}$ be the partition of the real line into (maximal) open intervals by the endpoints of R, such that these intervals do not contain points of R in their interior.

Consider an interval (i.e., a one-dimensional trapezoid) of $\widehat{\mathcal{I}}$. It is intuitively clear that this interval (in expectation) would contain $O(n/i)$ points. Indeed, fix a point x on the real line, and imagine that we pick each point with probability i/n to be in the random sample. The random variable which is the number of points of P we have to scan starting from x and going to the right of x till we "hit" a point that is in the random sample behaves like a geometric variable with probability i/n, and as such its

expected value is n/i. The same argument works if we scan P to the left of x. We conclude that the number of points of P in the interval of $\widehat{\mathcal{I}}$ that contains x but does not contain any point of R is $O(n/i)$ in expectation.

Of course, the vertical decomposition case is more involved, as each vertical trapezoid is defined by four input segments. Furthermore, the number of possible vertical trapezoids is larger. Instead of proving the required result for this special case, we will prove a more general result which can be applied in a lot of other settings.

21.2. General settings

21.2.1. Notation

Let S be a set of objects. For a subset $\mathsf{R} \subseteq \mathsf{S}$, we define a collection of 'regions' called $\mathcal{F}(\mathsf{R})$. For the case of vertical decomposition of segments (i.e., Theorem 21.1.3), the objects are segments, the regions are trapezoids, and $\mathcal{F}(\mathsf{R})$ is the set of vertical trapezoids in $\mathcal{A}^{|}(\mathsf{R})$. Let

$$\mathcal{T} = \mathcal{T}(\mathsf{S}) = \bigcup_{\mathsf{R} \subseteq \mathsf{S}} \mathcal{F}(\mathsf{R})$$

denote the set of ***all possible regions*** defined by subsets of S.

In the vertical trapezoids case, the set $\mathcal{T}$ is the set of all vertical trapezoids that can be defined by any subset of the given input segments.

We associate two subsets $D(\sigma), K(\sigma) \subseteq \mathsf{S}$ with each region $\sigma \in \mathcal{T}$.

The ***defining set*** $D(\sigma)$ of σ is the subset of S defining the region σ (the precise requirements from this set are specified in the axioms below). We assume that for every $\sigma \in \mathcal{T}$, $|D(\sigma)| \leq d$ for a (small) constant d. The constant d is sometime referred to as the ***combinatorial dimension***. In the case of Theorem 21.1.3, each trapezoid σ is defined by at most four segments (or lines) of S that define the region covered by the trapezoid σ, and this set of segments is $D(\sigma)$. See Figure 21.1.

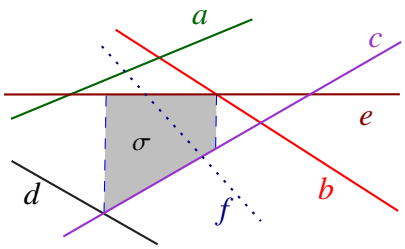


Figure 21.1: $D(\sigma) = \{b, c, d, e\}$ and $K(\sigma) = \{f\}$.

The ***stopping set*** $K(\sigma)$ of σ is the set of objects of S such that including any object of $K(\sigma)$ in R prevents σ from appearing in $\mathcal{F}(\mathsf{R})$. In many applications $K(\sigma)$ is just the set of objects intersecting the cell σ; this is also the case in Theorem 21.1.3, where $K(\sigma)$ is the set of segments of S intersecting the interior of the trapezoid σ (see Figure 21.1). Thus, the stopping set of a region σ, in many cases, is just the conflict list of this region, when it is being created by an RIC algorithm. The ***weight*** of σ is $\omega(\sigma) = |K(\sigma)|$.

Axioms. Let $\mathsf{S}, \mathcal{F}(\mathsf{R}), D(\sigma)$, and $K(\sigma)$ be such that for any subset $\mathsf{R} \subseteq \mathsf{S}$, the set $\mathcal{F}(\mathsf{R})$ satisfies the following axioms:

(i) For any $\sigma \in \mathcal{F}(\mathsf{R})$, we have $D(\sigma) \subseteq \mathsf{R}$ and $\mathsf{R} \cap K(\sigma) = \emptyset$.
(ii) If $D(\sigma) \subseteq \mathsf{R}$ and $K(\sigma) \cap \mathsf{R} = \emptyset$, then $\sigma \in \mathcal{F}(\mathsf{R})$.

21.2.1.1. Examples of the general framework

(A) **Vertical decomposition.** Discussed above.

(B) **Points on a line.** Let S be a set of n points on the real line. For a set $\mathsf{R} \subseteq \mathsf{S}$, let $\mathcal{F}(\mathsf{R})$ be the set of atomic intervals of the real lines formed by R; that is, the partition of the real line into maximal connected sets (i.e., intervals and rays) that do not contain a point of R in their interior.
Clearly, in this case, an interval $\mathcal{I} \in \mathcal{F}(\mathsf{R})$ the defining set of $\mathcal{I}$ (i.e., $D(\mathcal{I})$) is the set containing the (one or two) endpoints of $\mathcal{I}$ in R. The stopping set of an $\mathcal{I}$ is the set $K(\mathcal{I})$, which is the set of all points of S contained in $\mathcal{I}$.

(C) **Vertices of the convex-hull in 2d.** Consider a set S of n points in the plane. A vertex on the convex hull is defined by the point defining the vertex, and the two edges before and after it on the convex hull. To this end, a ***certified vertex*** of the convex hull (say this vertex is q) is a triplet $(\mathsf{p}, \mathsf{q}, \mathsf{r})$, such that p, q and r are consecutive vertices of $\mathcal{CH}(\mathsf{S})$ (say, in clockwise order). Observe, that computing the convex-hull of S is equivalent to computing the set of certified vertices of S.
For a set $\mathsf{R} \subseteq \mathsf{S}$, let $\mathcal{F}(\mathsf{R})$ denote the set of certified vertices of R (i.e., this is equivalent to the set of vertices of the convex-hull of R. For a certified vertex $\sigma \in \mathcal{F}(\mathsf{R})$, its defining set is the set of three vertices $\mathsf{p}, \mathsf{q}, \mathsf{r}$ that (surprise, surprise) define it. Its stopping set, is the set of all points in S, that either on the "wrong" side of the line spanning pq, or on the "wrong" side of the line spanning qr. Equivalently, $K(\sigma)$ is the set of all points $\mathsf{t} \in \mathsf{S} \setminus \mathsf{R}$, such that the convex-hull of $\mathsf{p}, \mathsf{q}, \mathsf{r}$, and t does not form a convex quadrilateral.

(D) **Edges of the convex-hull in 3d.**
Let S be a set of points in three dimensions. An edge e of the convex-hull of a set $\mathsf{R} \subseteq \mathit{ObjSet}$ of points in $\mathbb{R}^3$ is defined by two vertices of S, and it can be certified as being on the convex hull $\mathcal{CH}(\mathsf{R})$, by the two faces f, f' adjacent to e. If all the points of R are on the "right" side of both these two faces then e is an edge of the convex hull of R. Computing all the certified edges of S is equivalent to computing the convex-hull of S.
In the following, assume that each face of any convex-hull of a subset of points of S is a triangle. As such, a face of the convex-hull would be defined by three points. Formally, the ***butterfly*** of an edge e of $\mathcal{CH}(\mathsf{R})$ is $(\mathsf{e}, \mathsf{p}, \mathsf{q})$, where $\mathit{pnt}, \mathsf{q} \in \mathsf{R}$, and such that all the points of R are on the same side as q of the plane spanned by e and p (we have symmetric condition requiring that all the points of S are on the same as p of the plane spanned by e and q).
For a set $\mathsf{R} \subseteq \mathsf{P}$, let $\mathcal{F}(\mathsf{R})$ be its set of butterflies. Clearly, computing all the butterflies of S (i.e., $\mathcal{F}(\mathsf{S})$) is equivalent to computing the convex-hull of S.
For a butterfly $\sigma = (\mathsf{e}, \mathsf{p}, \mathsf{q}) \in \mathcal{F}(\mathsf{R})$ its defining set (i.e., $D(\sigma)$) is a set of four points (i.e., the two points defining its edge e, and the to additional vertices defining the two faces Face and f' adjacent to it). Its stopping set $K(\sigma)$, is the set of all the points of $\mathsf{S} \setminus \mathsf{R}$ that of different sides of the plane spanned by e and p (resp. e and q) than q (resp. p) [here, the stopping set is the union of these two sets].

(E) **Delaunay triangles in 2d.**
For a set of S of n points in the plane. Consider a subset $\mathsf{R} \subseteq \mathsf{S}$. A ***Delaunay circle*** of R is a disc D that has three points $\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3$ of R on its boundary, and no points of R in its interior. Naturally, these three points define a ***Delaunay triangle*** $\triangle = \triangle \mathsf{p}_1 \mathsf{p}_2 \mathsf{p}_3$. The defining set is $D(\triangle) = \{\mathsf{p}_1, \mathsf{p}_2, \mathsf{p}_3\}$, and the stopping set $K(\triangle)$ is the set of all points in S that are contained in the interior of the disk D.

21.2.2. Analysis

In the following, S is a set of n objects complying with axioms (i) and (ii).

The challenge. What makes the analysis not easy is that there are dependencies between the defining

set of a region and its stopping set (i.e., conflict list). In particular, we have the following difficulties

(A) The defining set might be of different sizes depending on the region σ being considered.

(B) Even if all the regions have a defining set of the same size d (say, 4 as in the case of vertical trapezoids), it is not true that every d objects define a valid region. For example, for the case of segments, the four segments might be vertically separated from each other (i.e., think about them as being four disjoint intervals on the real line), and they do not define a vertical trapezoid together. Thus, our analysis is going to be a bit loopy loop – we are going to assume we know how many regions exists (in expectation) for a random sample of certain size, and use this to derive the desired bounds.

21.2.2.1. On the probability of a region to be created

Inherently, to analyze a randomized algorithm using this framework, we will be interested in the probability that a certain region would be created. Thus, let

$$\rho_{r,n}(d,k)$$

denote the probability that a region $\sigma \in \mathcal{T}$ appears in $\mathcal{F}(\mathsf{R})$, where its defining set is of size d, its stopping set is of size k, R is a random sample of size r from a set S, and $n = |\mathsf{S}|$. Specifically, σ is a *feasible* region that might be created by an algorithm computing $\mathcal{F}(\mathsf{R})$.

The sampling model. For describing algorithms it is usually easier to work with samples created by picking a subset of a certain size (without repetition) from the original set of objects. Usually, in the algorithmic applications this would be done by randomly permuting the objects and interpreting a prefix of this permutation as a random sample. Insisting on analyzing this framework in the "right" sampling model creates some non-trivial technical pain.

Lemma 21.2.1. *We have that* $\rho_{r,n}(d,k) \approx \left(1-\frac{r}{n}\right)^k \left(\frac{r}{n}\right)^d$. *Formally,*

$$\frac{1}{2^{2d}} \left(1-4\cdot\frac{r}{n}\right)^k \left(\frac{r}{n}\right)^d \;\le\; \rho_{r,n}(d,k) \;\le\; 2^{2d} \left(1-\frac{1}{2}\cdot\frac{r}{n}\right)^k \left(\frac{r}{n}\right)^d. \tag{21.2}$$

Proof: Let σ be the region under consideration that is defined by d objects and having k stoppers (i.e., $k = K(\sigma)$). We are interested in the probability of σ being created when taking a sample of size r (without repetition) from a set S of n objects. Clearly, this probability is $\rho_{r,n}(d,k) = \binom{n-d-k}{r-d} \,/\, \binom{n}{r}$, as we have to pick the d defining objects into the random sample and avoid picking any of the k stoppers. A tedious but careful calculation, delegated to Section 21.4, implies Eq. (21.2).

Instead, here is an elegant argument for why this estimate is correct in a slightly different sampling model. We pick every element of S into the sample R with probability r/n, and this is done independently for each object. In expectation, the random sample is of size r, and clearly the probability that σ is created is the probability that we pick its d defining objects (that is, $(r/n)^d$) multiplied by the probability that we did not pick any of its k stoppers (that is, $(1-r/n)^k$). ∎

Remark 21.2.2. The bounds of Eq. (21.2) hold only when r, d, and k are in certain (reasonable) ranges. For the sake of simplicity of exposition we ignore this minor issue. With care, all our arguments work when one pays careful attention to this minor technicality.

21.2.2.2. On exponential decay

For any natural number r and a number $t > 0$, consider R to be a random sample of size r from S without repetition. We will refer to a region $\sigma \in \mathcal{F}(\mathsf{R})$ as being ***t-heavy*** if $\omega(\sigma) \geq t \cdot \frac{n}{r}$. Let $\mathcal{F}_{\geq t}(\mathsf{R})$ denote all the t-heavy regions of $\mathcal{F}(\mathsf{R})$.[④]

Intuitively, and somewhat incorrectly, we expect the average weight of a region of $\mathcal{F}(\mathsf{R})$ to be roughly n/r. We thus expect the size of this set to drop fast as t increases. Indeed, Lemma 21.2.1 tells us that a trapezoid of weight $t\,(n/r)$ has probability

$$\begin{aligned}\rho_{r,n}\Big(d, t\cdot\frac{n}{r}\Big) &\approx \Big(1-\frac{r}{n}\Big)^{t\,(n/r)}\Big(\frac{r}{n}\Big)^d \approx \exp(-t)\cdot\Big(\frac{r}{n}\Big)^d \approx \exp\big(-t+1\big)\cdot\Big(1-\frac{r}{n}\Big)^{n/r}\Big(\frac{r}{n}\Big)^d \\ &\approx \exp(-t+1)\cdot\rho_{r,n}(d, n/r)\end{aligned}$$

to be created, since $(1-r/n)^{n/r} \approx 1/e$. Namely, a t-heavy region has exponentially lower probability to be created than a region of weight n/r. We next formalize this argument.

Lemma 21.2.3. *Let $r \leq n$ and let t be parameters, such that $1 \leq t \leq r/d$. Furthermore, let R be a sample of size r, and let R' be a sample of size $r' = \lfloor r/t \rfloor$, both from S. Let $\sigma \in \mathcal{T}$ be a region with weight $\omega(\sigma) \geq t\,(n/r)$. Then, $\mathbf{Pr}\Big[\sigma \in \mathcal{F}(\mathsf{R})\Big] = O\Big(\exp\Big(-\frac{t}{2}\Big)t^d\,\mathbf{Pr}\Big[\sigma \in \mathcal{F}(\mathsf{R}')\Big]\Big)$.*

Proof: For the sake of simplicity of exposition, assume that $k = \omega(\sigma) = t\,(n/r)$. By Lemma 21.2.1 (i.e., Eq. (21.2)) we have

$$\begin{aligned}\frac{\mathbf{Pr}[\sigma \in \mathcal{F}(\mathsf{R})]}{\mathbf{Pr}[\sigma \in \mathcal{F}(\mathsf{R}')]} = \frac{\rho_{r,n}(d,k)}{\rho_{r',n}(d,k)} &\leq \frac{2^{2d}\big(1-\frac{1}{2}\cdot\frac{r}{n}\big)^k\big(\frac{r}{n}\big)^d}{\frac{1}{2^{2d}}\big(1-4\frac{r'}{n}\big)^k\big(\frac{r'}{n}\big)^d} \\ &\leq 2^{4d}\exp\Big(-\frac{kr}{2n}\Big)\Big(1+8\frac{r'}{n}\Big)^k\Big(\frac{r}{r'}\Big)^d \leq 2^{4d}\exp\Big(8\frac{kr'}{n}-\frac{kr}{2n}\Big)\Big(\frac{r}{r'}\Big)^d \\ &= 2^{4d}\exp\Big(8\frac{tn\,\lfloor r/t\rfloor}{nr}-\frac{tnr}{2nr}\Big)\Big(\frac{r}{\lfloor r/t\rfloor}\Big)^d = O\Big(\exp(-t/2)t^d\Big),\end{aligned}$$

since $1/(1-x) \leq 1+2x$ for $x \leq 1/2$ and $1+y \leq \exp(y)$, for all y. (The constant in the above $O(\cdot)$ depends exponentially on d.) ■

Let

$$\mathbf{E}f(r) = \mathbf{E}[|\mathcal{F}(\mathsf{R})|] \quad \text{and} \quad \mathbf{E}f_{\geq t}(r) = \mathbf{E}[|\mathcal{F}_{\geq t}(\mathsf{R})|],$$

where the expectation is over random subsets $\mathsf{R} \subseteq \mathsf{S}$ of size r. Note that $\mathbf{E}f(r) = \mathbf{E}f_{\geq 0}(r)$ is the expected number of regions created by a random sample of size r. In words, $\mathbf{E}f_{\geq t}(r)$ is the expected number of regions in a structure created by a sample of r random objects, such that these regions have weight which is t times larger than the "expected" weight (i.e., n/r). In the following, we assume that $\mathbf{E}f(r)$ is a monotone increasing function.

Lemma 21.2.4 (The exponential decay lemma). *Given a set S of n objects and parameters $r \leq n$ and $1 \leq t \leq r/d$, where $d = \max_{\sigma\in\mathcal{T}(\mathsf{S})}|D(\sigma)|$, if axioms (i) and (ii) above hold for any subset of S, then*

$$\mathbf{E}f_{\geq t}(r) = O\Big(t^d\exp(-t/2)\,\mathbf{E}f(r)\Big). \tag{21.3}$$

[④]These are the regions that are at least t times overweight. Speak about an obesity problem.

Proof: Let R be a random sample of size r from S and let R' be a random sample of size $r' = \lfloor r/t \rfloor$ from S. Let $H = \bigcup_{X \subseteq \mathsf{S}, |X|=r} \mathcal{F}_{\geq t}(X)$ denote the set of all t-heavy regions that might be created by a sample of size r. In the following, the expectation is taken over the content of the random samples R and R'.

For a region σ, let X_σ be the indicator variable that is 1 if and only if $\sigma \in \mathcal{F}(\mathsf{R})$. By linearity of expectation and since $\mathbf{E}[X_\sigma] = \mathbf{Pr}[\sigma \in \mathcal{F}(\mathsf{R})]$, we have

$$\begin{aligned}
\mathbf{E}f_{\geq t}(r) &= \mathbf{E}\Big[|\mathcal{F}_{\geq t}(\mathsf{R})|\Big] = \mathbf{E}\left[\sum_{\sigma \in H} X_\sigma\right] = \sum_{\sigma \in H} \mathbf{E}[X_\sigma] = \sum_{\sigma \in H} \mathbf{Pr}[\sigma \in \mathcal{F}(\mathsf{R})] \\
&= O\left(t^d \exp(-t/2) \sum_{\sigma \in H} \mathbf{Pr}[\sigma \in \mathcal{F}(\mathsf{R}')]\right) = O\left(t^d \exp(-t/2) \sum_{\sigma \in \mathcal{T}} \mathbf{Pr}[\sigma \in \mathcal{F}(\mathsf{R}')]\right) \\
&= O\Big(t^d \exp(-t/2)\, \mathbf{E}f(r')\Big) = O\Big(t^d \exp(-t/2)\, \mathbf{E}f(r)\Big),
\end{aligned}$$

by Lemma 21.2.3 and since $\mathbf{E}f(r)$ is a monotone increasing function. ■

21.2.2.3. Bounding the moments

Consider a different randomized algorithm that in a first round samples r objects, $\mathsf{R} \subseteq \mathsf{S}$ (say, segments), computes the arrangement induced by these r objects (i.e., $\mathcal{A}^{|}(\mathsf{R})$), and then inside each region σ it computes the arrangement of the $\omega(\sigma)$ objects intersecting the interior of this region, using an algorithm that takes $O((\omega(\sigma))^c)$ time, where $c > 0$ is some fixed constant. The overall expected running time of this algorithm is

$$\mathbf{E}\left[\sum_{\sigma \in \mathcal{F}(\mathsf{R})} \Big(\omega(\sigma)\Big)^c\right].$$

We are now able to bound this quantity.

Theorem 21.2.5 (Bounded moments theorem). *Let $\mathsf{R} \subseteq \mathsf{S}$ be a random subset of size r. Let $\mathbf{E}f(r) = \mathbf{E}[|\mathcal{F}(\mathsf{R})|]$ and let $c \geq 1$ be an arbitrary constant. Then,*

$$\mathbf{E}\left[\sum_{\sigma \in \mathcal{F}(\mathsf{R})} \Big(\omega(\sigma)\Big)^c\right] = O\Big(\mathbf{E}f(r) \left(\frac{n}{r}\right)^c\Big).$$

Proof: Let $\mathsf{R} \subseteq \mathsf{S}$ be a random sample of size r. Observe that all the regions with weight in the range $\left[(t-1)\frac{n}{r}, t \cdot \frac{n}{r}\right)$ are in the set $\mathcal{F}_{\geq t-1}(\mathsf{R}) \setminus \mathcal{F}_{\geq t}(\mathsf{R})$. As such, we have by Lemma 21.2.4 that

$$\begin{aligned}
\mathbf{E}\left[\sum_{\sigma \in \mathcal{F}(\mathsf{R})} \omega(\sigma)^c\right] &\leq \mathbf{E}\left[\sum_{t \geq 1} \Big(t\frac{n}{r}\Big)^c \big(|\mathcal{F}_{\geq t-1}(\mathsf{R})| - |\mathcal{F}_{\geq t}(\mathsf{R})|\big)\right] \leq \mathbf{E}\left[\sum_{t \geq 1} \Big(t\frac{n}{r}\Big)^c |\mathcal{F}_{\geq t-1}(\mathsf{R})|\right] \\
&\leq \left(\frac{n}{r}\right)^c \sum_{t \geq 0} (t+1)^c \cdot \mathbf{E}[|\mathcal{F}_{\geq t}(\mathsf{R})|] \\
&= \left(\frac{n}{r}\right)^c \sum_{t \geq 0} (t+1)^c\, \mathbf{E}f_{\geq t}(r) = \left(\frac{n}{r}\right)^c \sum_{t \geq 0} O\Big((t+1)^{c+d} \exp(-t/2)\, \mathbf{E}f(r)\Big) \\
&= O\left(\mathbf{E}f(r) \left(\frac{n}{r}\right)^c \sum_{t \geq 0} (t+1)^{c+d} \exp(-t/2)\right) = O\Big(\mathbf{E}f(r) \left(\frac{n}{r}\right)^c\Big),
\end{aligned}$$

since c and d are both constants. ■

21.3. Applications

21.3.1. Analyzing the RIC algorithm for vertical decomposition

We remind the reader that the input of the algorithm of Section 21.1.2 is a set S of n segments with k intersections, and it uses randomized incremental construction to compute the vertical decomposition of the arrangement $\mathcal{A}(\mathsf{S})$.

Lemma 21.1.2 shows that the number of vertical trapezoids in the randomized incremental construction is in expectation $\mathbf{E}f(i) = O\big(i + k\,(i/n)^2\big)$. Thus, by Theorem 21.2.5 (used with $c = 1$), we have that the total expected size of the conflict lists of the vertical decomposition computed in the ith step is

$$\mathbf{E}[W_i] = \mathbf{E}\left[\sum_{\sigma\in\mathcal{B}_i} \omega(\sigma)\right] = O\left(\mathbf{E}f(i)\frac{n}{i}\right) = O\left(n + k\frac{i}{n}\right).$$

This is the missing piece in the analysis of Section 21.1.2. Indeed, the amortized work in the ith step of the algorithm is $O(W_i/i)$ (see Eq. $(21.1)_{\text{p5}}$), and as such, the expected running time of this algorithm is

$$\mathbf{E}\left[O\left(\sum_{i=1}^{n}\frac{W_i}{i}\right)\right] = O\left(\sum_{i=1}^{n}\frac{1}{i}\left(n + k\frac{i}{n}\right)\right) = O(n\log n + k).$$

This implies Theorem 21.1.3.

21.3.2. Cuttings

Let S be a set of n lines in the plane, and let r be an arbitrary parameter. A ***$(1/r)$-cutting*** of S is a partition of the plane into constant complexity regions such that each region intersects at most n/r lines of S. It is natural to try to minimize the number of regions in the cutting, as cuttings are a natural tool for performing "divide and conquer".

Consider the range space having S as its ground set and vertical trapezoids as its ranges (i.e., given a vertical trapezoid σ, its corresponding range is the set of all lines of S that intersect the interior of σ). This range space has a VC dimension which is a constant as can be easily verified. Let $X \subseteq \mathsf{S}$ be an ε-net for this range space, for $\varepsilon = 1/r$. By Theorem $21.7.1_{\text{p16}}$ (ε-net theorem), there exists such an ε-net X of this range space, of size $O((1/\varepsilon)\log(1/\varepsilon)) = O(r\log r)$. In fact, Theorem $21.7.1_{\text{p16}}$ states that an appropriate random sample is an ε-net with non-zero probability, which implies, by the probabilistic method, that such a net (of this size) exists.

Lemma 21.3.1. *There exists a $(1/r)$-cutting of a set of lines S in the plane of size $O\big((r\log r)^2\big)$.*

Proof: Consider the vertical decomposition $\mathcal{A}^{|}(X)$, where X is as above. We claim that this collection of trapezoids is the desired cutting.

The bound on the size is immediate, as the complexity of $\mathcal{A}^{|}(X)$ is $O\Big(|X|^2\Big)$ and $|X| = O(r\log r)$.

As for correctness, consider a vertical trapezoid σ in the arrangement $\mathcal{A}^{|}(X)$. It does not intersect any of the lines of X in its interior, since it is a trapezoid in the vertical decomposition $\mathcal{A}^{|}(X)$. Now, if σ intersected more than n/r lines of S in its interior, where $n = |\mathsf{S}|$, then it must be that the interior of σ intersects one of the lines of X, since X is an ε-net for S, a contradiction.

It follows that σ intersects at most $\varepsilon n = n/r$ lines of S in its interior. ■

Claim 21.3.2. *Any $(1/r)$-cutting in the plane of n lines contains at least $\Omega(r^2)$ regions.*

Proof: An arrangement of n lines (in general position) has $M = \binom{n}{2}$ intersections. However, the number of intersections of the lines intersecting a single region in the cutting is at most $m = \binom{n/r}{2}$. This implies that any cutting must be of size at least $M/m = \Omega(n^2/(n/r)^2) = \Omega(r^2)$. ■

We can get cuttings of size matching the above lower bound using the moments technique.

Theorem 21.3.3. *Let S be a set of n lines in the plane, and let r be a parameter. One can compute a $(1/r)$-cutting of S of size $O(r^2)$.*

Proof: Let $\mathsf{R} \subseteq \mathsf{S}$ be a random sample of size r, and consider its vertical decomposition $\mathcal{A}^{\vert}(\mathsf{R})$. If a vertical trapezoid $\sigma \in \mathcal{A}^{\vert}(\mathsf{R})$ intersects at most n/r lines of S, then we can add it to the output cutting. The other possibility is that a σ intersects $t(n/r)$ lines of S, for some $t > 1$, and let $\mathrm{cl}(\sigma) \subset \mathsf{S}$ be the conflict list of σ (i.e., the list of lines of S that intersect the interior of σ). Clearly, a $(1/t)$-cutting for the set $\mathrm{cl}(\sigma)$ forms a vertical decomposition (clipped inside σ) such that each trapezoid in this cutting intersects at most n/r lines of S. Thus, we compute such a cutting inside each such "heavy" trapezoid using the algorithm (implicit in the proof) of Lemma 21.3.1, and these subtrapezoids to the resulting cutting. Clearly, the size of the resulting cutting inside σ is $O(t^2 \log^2 t) = O(t^4)$. The resulting two-level partition is clearly the required cutting. By Theorem 21.2.5, the expected size of the cutting is

$$O\left(\mathbf{E}f(r) + \mathbf{E}\left[\sum_{\sigma \in \mathcal{F}(\mathsf{R})} \left(2\frac{\omega(\sigma)}{n/r}\right)^4\right]\right) = O\left(\mathbf{E}f(r) + \left(\frac{r}{n}\right)^4 \mathbf{E}\left[\sum_{\sigma \in \mathcal{F}(\mathsf{R})} (\omega(\sigma))^4\right]\right)$$
$$= O\left(\mathbf{E}f(r) + \left(\frac{r}{n}\right)^4 \cdot \mathbf{E}f(r)\left(\frac{n}{r}\right)^4\right) = O(\mathbf{E}f(r)) = O(r^2),$$

since $\mathbf{E}f(r)$ is proportional to the complexity of $\mathcal{A}(\mathsf{R})$ which is $O(r^2)$. ■

21.4. Bounds on the probability of a region to be created

Here we prove Lemma 21.2.1$_{\mathrm{p8}}$ in the "right" sampling model. The casual reader is encouraged to skip this section, as it contains mostly tedious (and not very insightful) calculations.

Let S be a given set of n objects. Let $\rho_{r,n}(d,k)$ be the probability that a region $\sigma \in \mathcal{T}$ whose defining set is of size d and whose stopping set is of size k appears in $\mathcal{F}(\mathsf{R})$, where R is a random sample from S of size r (without repetition).

Lemma 21.4.1. *We have* $\rho_{r,n}(d,k) = \dfrac{\binom{n-d-k}{r-d}}{\binom{n}{r}} = \dfrac{\binom{n-d-k}{r-d}}{\binom{n}{r-d}} \cdot \dfrac{\binom{r}{d}}{\binom{n-(r-d)}{d}} = \dfrac{\binom{n-d-k}{r-d}}{\binom{n-d}{r-d}} \cdot \dfrac{\binom{r}{d}}{\binom{n}{d}}$.

Proof: So, consider a region σ with d defining objects in $D(\sigma)$ and k detractors in $K(\sigma)$. We have to pick the d defining objects of $D(\sigma)$ to be in the random sample R of size r but avoid picking any of the k objects of $K(\sigma)$ to be in R.

The second part follows since $\binom{n}{r} = \binom{n}{r-d}\binom{n-(r-d)}{d} \Big/ \binom{r}{d}$. Indeed, for the right-hand side first pick a sample of size $r-d$ and then a sample of size d from the remaining objects. Merging the two

random samples, we get a random sample of size r. However, since we do not care if an object is in the first sample or second sample, we observe that every such random sample is being counted $\binom{r}{d}$ times.

The third part is easier, as it follows from $\binom{n}{r-d}\binom{n-(r-d)}{d} = \binom{n}{d}\binom{n-d}{r-d}$. The two sides count the different ways to pick two subsets from a set of size n, the first one of size d and the second one of size $r-d$. ■

Lemma 21.4.2. *For $M \geq m \geq t \geq 0$, we have* $\left(\frac{m-t}{M-t}\right)^t \leq \frac{\binom{m}{t}}{\binom{M}{t}} \leq \left(\frac{m}{M}\right)^t$.

Proof: We have that $\alpha = \frac{\binom{m}{t}}{\binom{M}{t}} = \frac{m!}{(m-t)!t!}\frac{(M-t)!t!}{M!} = \frac{m}{M}\cdot\frac{m-1}{M-1}\cdots\frac{m-t+1}{M-t+1}$. Now, since $M \geq m$, we have that $\frac{m-i}{M-i} \leq \frac{m}{M}$, for all $i \geq 0$. As such, the maximum (resp. minimum) fraction on the right-hand size is m/M (resp. $\frac{m-t+1}{M-t+1}$). As such, we have $\left(\frac{m-t}{M-t}\right)^t \leq \left(\frac{m-t+1}{M-t+1}\right)^t \leq \alpha \leq (m/M)^t$. ■

Lemma 21.4.3. *Let $0 \leq X, Y \leq N$. We have that* $\left(1-\frac{X}{N}\right)^Y \leq \left(1-\frac{Y}{2N}\right)^X$.

Proof: Since $1-\alpha \leq \exp(-\alpha) \leq (1-\alpha/2)$, for $0 \leq \alpha \leq 1$, it follows that

$$\left(1-\frac{X}{N}\right)^Y \leq \exp\left(-\frac{XY}{N}\right) = \left(\exp\left(-\frac{Y}{n}\right)\right)^X \leq \left(1-\frac{Y}{2n}\right)^X.$$

■

Lemma 21.4.4. *For $2d \leq r \leq n/8$ and $k \leq n/2$, we have that*

$$\frac{1}{2^{2d}}\left(1-4\cdot\frac{r}{n}\right)^k\left(\frac{r}{n}\right)^d \leq \rho_{r,n}(d,k) \leq 2^{2d}\left(1-\frac{1}{2}\cdot\frac{r}{n}\right)^k\left(\frac{r}{n}\right)^d.$$

Proof: By Lemma 21.4.1, Lemma 21.4.2, and Lemma 21.4.3 we have

$$\begin{aligned}\rho_{r,n}(d,k) &= \frac{\binom{n-d-k}{r-d}}{\binom{n-d}{r-d}}\cdot\frac{\binom{r}{d}}{\binom{n}{d}} \leq \left(\frac{n-d-k}{n-d}\right)^{r-d}\left(\frac{r}{n}\right)^d \leq \left(1-\frac{k}{n}\right)^{r-d}\left(\frac{r}{n}\right)^d \leq 2^d\left(1-\frac{k}{n}\right)^r\left(\frac{r}{n}\right)^d \\ &\leq 2^d\left(1-\frac{r}{2n}\right)^k\left(\frac{r}{n}\right)^d,\end{aligned}$$

since $k \leq n/2$. As for the other direction, by similar argumentation, we have

$$\begin{aligned}\rho_{r,n}(d,k) &= \frac{\binom{n-d-k}{r-d}}{\binom{n}{r-d}}\cdot\frac{\binom{r}{d}}{\binom{n-(r-d)}{d}} \geq \left(\frac{n-d-k-(r-d)}{n-(r-d)}\right)^{r-d}\left(\frac{r-d}{n-(r-d)-d}\right)^d \\ &= \left(1-\frac{d+k}{n-(r-d)}\right)^{r-d}\left(\frac{r-d}{n-r}\right)^d \geq \left(1-\frac{d+k}{n/2}\right)^r\left(\frac{r/2}{n}\right)^d \\ &\geq \frac{1}{2^d}\left(1-\frac{4r}{n}\right)^{d+k}\left(\frac{r}{n}\right)^d \geq \frac{1}{2^{2d}}\left(1-\frac{4r}{n}\right)^k\left(\frac{r}{n}\right)^d,\end{aligned}$$

by Lemma 21.4.3 (setting $N = n/4$, $X = r$, and $Y = d+k$) and since $r \geq 2d$ and $4r/n \leq 1/2$. ■

21.5. Bibliographical notes

The technique described in this chapter is generally attributed to the work by Clarkson and Shor [CS89], which is historically inaccurate as the technique was developed by Clarkson [Cla88]. Instead of mildly confusing the matter by referring to it as the Clarkson technique, we decided to make sure to really confuse the reader and refer to it as the ***moments technique***. The Clarkson technique [Cla88] is in fact more general and implies a connection between the number of "heavy" regions and "light" regions. The general framework can be traced back to the earlier paper [Cla87]. This implies several beautiful results, some of which we cover later in the book.

For the full details of the algorithm of Section 21.1, the interested reader is refereed to the books [dBCKO08, BY98]. Interestingly, in some cases the merging stage can be skipped; see [Har00a].

Agarwal *et al.* [AMS98] presented a slightly stronger variant than the original version of Clarkson [Cla88] that allows a region to disappear even if none of the members of its stopping set are in the random sample. This stronger setting is used in computing the vertical decomposition of a single face in an arrangement (instead of the whole arrangement). Here an insertion of a faraway segment of the random sample might cut off a portion of the face of interest. In particular, in the settings of Agarwal *et al.* Axiom (ii) is replaced by the following:

(ii) If $\sigma \in \mathcal{F}(\mathsf{R})$ and R' is a subset of R with $D(\sigma) \subseteq \mathsf{R}'$, then $\sigma \in \mathcal{F}(\mathsf{R}')$.

Interestingly, Clarkson [Cla88] did not prove Theorem 21.2.5 using the exponential decay lemma but gave a direct proof. In fact, his proof implicitly contains the exponential decay lemma. We chose the current exposition since it is more modular and provides a better intuition of what is really going on and is hopefully slightly simpler. In particular, Lemma 21.2.1 is inspired by the work of Sharir [Sha03].

The exponential decay lemma (Lemma 21.2.4) was proved by Chazelle and Friedman [CF90]. The work of Agarwal *et al.* [AMS98] is a further extension of this result. Another analysis was provided by Clarkson *et al.* [CMS93].

Another way to reach similar results is using the technique of Mulmuley [Mul94], which relies on a direct analysis on 'stoppers' and 'triggers'. This technique is somewhat less convenient to use but is applicable to some settings where the moments technique does not apply directly. Also, his concept of the omega function might explain why randomized incremental algorithms perform better in practice than their worst case analysis [Mul89].

Backwards analysis in geometric settings was first used by Chew [Che86] and was formalized by Seidel [Sei93]. It is similar to the "leave one out" argument used in statistics for cross validation. The basic idea was probably known to the Greeks (or Russians or French) at some point in time.

(Naturally, our summary of the development is cursory at best and not necessarily accurate, and all possible disclaimers apply. A good summary is provided in the introduction of [Sei93].)

Sampling model. As a rule of thumb all the different sampling approaches are similar and yield similar results. For example, we used such an alternative sampling approach in the "proof" of Lemma 21.2.1. It is a good idea to use whichever sampling scheme is the easiest to analyze in figuring out what's going on. Of course, a formal proof requires analyzing the algorithm in the sampling model its uses.

Lazy randomized incremental construction. If one wants to compute a single face that contains a marking point in an arrangement of curves, then the problem in using randomized incremental construction is that as you add curves, the region of interest shrinks, and regions that were maintained should be ignored. One option is to perform flooding in the vertical decomposition to figure out what trapezoids are still reachable from the marking point and maintaining only these trapezoids in the conflict graph. Doing it in each iteration is way too expensive, but luckily one can use a lazy strategy that performs this

cleanup only a logarithmic number of times (i.e., you perform a cleanup in an iteration if the iteration number is, say, a power of 2). This strategy complicates the analysis a bit; see [dBDS95] for more details on this ***lazy randomized incremental construction*** technique. An alternative technique was suggested by the author for the (more restricted) case of planar arrangements; see [Har00b]. The idea is to compute only what the algorithm really needs to compute the output, by computing the vertical decomposition in an exploratory online fashion. The details are unfortunately overwhelming although the algorithm seems to perform quite well in practice.

Cuttings. The concept of cuttings was introduced by Clarkson. The first optimal size cuttings were constructed by Chazelle and Friedman [CF90], who proved the exponential decay lemma to this end. Our elegant proof follows the presentation by de Berg and Schwarzkopf [dBS95]. The problem with this approach is that the constant involved in the cutting size is awful[5]. Matoušek [Mat98] showed that there $(1/r)$-cuttings with $8r^2 + 6r + 4$ trapezoids, by using level approximation. A different approach was taken by the author [Har00a], who showed how to get cuttings which seem to be quite small (i.e., constant-wise) in practice. The basic idea is to do randomized incremental construction but at each iteration greedily add all the trapezoids with conflict list small enough to the cutting being output. One can prove that this algorithm also generates $O(r^2)$ cuttings, but the details are not trivial as the framework described in this chapter is not applicable for analyzing this algorithm.

Cuttings also can be computed in higher dimensions for hyperplanes. In the plane, cuttings can also be computed for well-behaved curves; see [SA95].

Another fascinating concept is ***shallow cuttings***. These are cuttings covering only portions of the arrangement that are in the "bottom" of the arrangement. Matoušek came up with the concept [Mat92]. See [AES99, CCH09] for extensions and applications of shallow cuttings.

Even more on randomized algorithms in geometry. We have only scratched the surface of this fascinating topic, which is one of the cornerstones of "modern" computational geometry. The interested reader should have a look at the books by Mulmuley [Mul94], Sharir and Agarwal [SA95], Matoušek [Mat02], and Boissonnat and Yvinec [BY98].

21.6. Exercises

Exercise 21.6.1 (Convex hulls incrementally). Let P be a set of n points in the plane.

(A) Describe a randomized incremental algorithm for computing the convex hull $\mathcal{CH}(\mathsf{P})$. Bound the expected running time of your algorithm.

(B) Assume that for any subset of P, its convex hull has complexity t (i.e., the convex hull of the subset has t edges). What is the expected running time of your algorithm in this case? If your algorithm is not faster for this case (for example, think about the case where $t = O(\log n)$), describe a variant of your algorithm which is faster for this case.

Exercise 21.6.2 (Compressed quadtree made incremental). Given a set P of n points in $\mathbb{R}^d$, describe a randomized incremental algorithm for building a compressed quadtree for P that works in expected $O(dn \log n)$ time. Prove the bound on the running time of your algorithm.

[5] This is why all computations related to cuttings should be done on a waiter's bill pad. As Douglas Adams put it: "On a waiter's bill pad, reality and unreality collide on such a fundamental level that each becomes the other and anything is possible, within certain parameters."

21.7. From previous lectures

Theorem 21.7.1 (ε-net theorem, [HW87]). *Let $(X, \mathcal{R})$ be a range space of* VC *dimension δ, let x be a finite subset of X, and suppose that $0 < \varepsilon \leq 1$ and $\varphi < 1$. Let N be a set obtained by m random independent draws from x, where*

$$m \geq \max\left(\frac{4}{\varepsilon} \lg \frac{4}{\varphi}, \frac{8\delta}{\varepsilon} \lg \frac{16}{\varepsilon} \right). \tag{21.4}$$

Then N is an ε-net for x with probability at least $1 - \varphi$.

Lemma 21.7.2. *For any two random variables X and Y, we have* $\mathbf{E}\Big[\mathbf{E}\big[X \,\big|\, Y\big]\Big] = \mathbf{E}\big[X\big]$.

Bibliography

[AES99] P. K. Agarwal, A. Efrat, and M. Sharir. Vertical decomposition of shallow levels in 3-dimensional arrangements and its applications. *SIAM J. Comput.*, 29:912–953, 1999.

[AMS98] P. K. Agarwal, J. Matoušek, and O. Schwarzkopf. Computing many faces in arrangements of lines and segments. *SIAM J. Comput.*, 27(2):491–505, 1998.

[BY98] J.-D. Boissonnat and M. Yvinec. *Algorithmic Geometry*. Cambridge University Press, 1998.

[CCH09] C. Chekuri, K. L. Clarkson., and S. Har-Peled. On the set multi-cover problem in geometric settings. In *Proc. 25th Annu. Sympos. Comput. Geom.* (SoCG), pages 341–350, 2009.

[CF90] B. Chazelle and J. Friedman. A deterministic view of random sampling and its use in geometry. *Combinatorica*, 10(3):229–249, 1990.

[Che86] L. P. Chew. Building Voronoi diagrams for convex polygons in linear expected time. Technical Report PCS-TR90-147, Dept. Math. Comput. Sci., Dartmouth College, Hanover, NH, 1986.

[Cla87] K. L. Clarkson. New applications of random sampling in computational geometry. *Discrete Comput. Geom.*, 2:195–222, 1987.

[Cla88] K. L. Clarkson. Applications of random sampling in computational geometry, II. In *Proc. 4th Annu. Sympos. Comput. Geom.* (SoCG), pages 1–11, New York, NY, USA, 1988. ACM.

[CMS93] K. L. Clarkson, K. Mehlhorn, and R. Seidel. Four results on randomized incremental constructions. *Comput. Geom. Theory Appl.*, 3(4):185–212, 1993.

[CS89] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Discrete Comput. Geom.*, 4:387–421, 1989.

[dBCKO08] M. de Berg, O. Cheong, M. van Kreveld, and M. H. Overmars. *Computational Geometry: Algorithms and Applications*. Springer-Verlag, Santa Clara, CA, USA, 3rd edition, 2008.

[dBDS95] M. de Berg, K. Dobrindt, and O. Schwarzkopf. On lazy randomized incremental construction. *Discrete Comput. Geom.*, 14:261–286, 1995.

[dBS95] M. de Berg and O. Schwarzkopf. Cuttings and applications. *Internat. J. Comput. Geom. Appl.*, 5:343–355, 1995.

[Har00a] S. Har-Peled. Constructing planar cuttings in theory and practice. *SIAM J. Comput.*, 29(6):2016–2039, 2000.

[Har00b] S. Har-Peled. Taking a walk in a planar arrangement. *SIAM J. Comput.*, 30(4):1341–1367, 2000.

[HW87] D. Haussler and E. Welzl. ε-nets and simplex range queries. *Discrete Comput. Geom.*, 2:127–151, 1987.

[Mat92] J. Matoušek. Reporting points in halfspaces. *Comput. Geom. Theory Appl.*, 2(3):169–186, 1992.

[Mat98] J. Matoušek. On constants for cuttings in the plane. *Discrete Comput. Geom.*, 20:427–448, 1998.

[Mat02] J. Matoušek. *Lectures on Discrete Geometry*, volume 212 of *Grad. Text in Math.* Springer, 2002.

[Mul89] K. Mulmuley. An efficient algorithm for hidden surface removal. *Comput. Graph.*, 23(3):379–388, 1989 1989.

[Mul94] K. Mulmuley. *Computational Geometry: An Introduction Through Randomized Algorithms*. Prentice Hall, Englewood Cliffs, NJ, 1994.

[SA95] M. Sharir and P. K. Agarwal. *Davenport-Schinzel Sequences and Their Geometric Applications*. Cambridge University Press, New York, 1995.

[Sei93] R. Seidel. Backwards analysis of randomized geometric algorithms. In J. Pach, editor, *New Trends in Discrete and Computational Geometry*, volume 10 of *Algorithms and Combinatorics*, pages 37–68. Springer-Verlag, 1993.

[Sha03] M. Sharir. The Clarkson-Shor technique revisited and extended. *Comb., Prob. & Comput.*, 12(2):191–201, 2003.