Four Results on Randomized Incremental Constructions*

Kenneth L. Clarkson† Kurt Mehlhorn‡ Raimund Seidel§

November 5, 1991
revised February 13, 1992

Abstract

We prove four results on randomized incremental constructions (RICs):

- an analysis of the expected behavior under insertion and deletions,
- a fully dynamic data structure for convex hull maintenance in arbitrary dimensions,
- a tail estimate for the space complexity of RICs,
- a lower bound on the complexity of a game related to RICs.

1 Introduction

Randomized incremental construction (RIC) is a powerful paradigm for geometric algorithms [CS89, Mul88, BDS$^+$]. It leads to simple and efficient algorithms for a wide range of geometric problems: line segment intersection [CS89, Mul88], convex hulls [CS89, Sei90], Voronoi diagrams [CS89, MMO91, GKS90, Dev], triangulation of simple polygons [Sei91], and many others. In this paper we make four contributions to the study of RICs.

- We give a simple analysis of the expected behavior of RICs; cf. § 2. We deal with insertions and deletions and derive bounds for the expected number of regions constructed and the expected number of conflicts encountered in the construction. In the case of deletions our bounds are new, but compare [DMT91, Mul91a, Mul91b, Mul91c, Sch91] for related results, in the case of insertions the results were known, but our proofs are simpler.

*A preliminary version of this papers was presented at the 9th Symposium on Theoretical Aspects of Computer Science (STACS 92)

†AT&T Bell Laboratories, Murray Hill, NJ 07974

‡Max Planck Institut für Informatik and Universität des Saarlandes; supported in part by ESPRIT II Basic Research Actions Program of the EC under contract no. 3075(project ALCOM)

§Computer Science Division, University of California at Berkeley, Berkeley, CA 94720

- We apply the general results on RIC to the problem of maintaining convex hulls in d-dimensional space; cf. § 3. We show that random insertions and deletions take expected time $O(\log n)$ for $d \leq 3$ and time $O(n^{\lfloor d/2 \rfloor - 1})$ otherwise. If the points are in convex position, which is, e.g., the case when Voronoi diagrams are transformed into convex hulls of one higher dimension, the deletion time becomes $\log \log n$ for $d \leq 3$. Schwarzkopf [Sch91] has obtained the same bounds for all $d \geq 6$, Mulmuley [Mul91c] has obtained the same bound for all d but with a more complex construction, and Devillers et al [DMT91] have previoulsy obtained the result for 2-dimensional Voronoi-diagrams.

- We derive a tail estimate for the number of regions constructed in RICs; cf. § 4.

- We study the complexity of a game related to the $O(n \log^* n)$ RICs of [Sei91] and [Dev] and show that the complexity of the game is $\Theta(n \log^* n)$; cf. § 5.

2 Randomized Incremental Constructions: General Theorems

Let S be a set with $|S| = n$ elements, which we will sometimes call *objects*. Let $\mathcal{F}(S)$ be a multiset whose elements are nonempty subsets of S, and let b be the size of the largest element of $\mathcal{F}(S)$. We will call the elements of $\mathcal{F}(S)$ *regions* or *ranges*. If all the regions have size b, we will say that $\mathcal{F}(S)$ is *uniform*. For a region $F \in \mathcal{F}(S)$ and an object x, if $x \in F$ we say that F *relies on* x or x *supports* F. For $R \subseteq S$, define $\mathcal{F}(R) = \{F \in \mathcal{F}(S) \mid F \subset R\}$. (That is, the multiplicity of F in $\mathcal{F}(R)$ is the same as in $\mathcal{F}(S)$.) We also assume a *conflict relation* $C \subseteq S \times \mathcal{F}(S)$ between objects and regions. We postulate that for all $x \in S$ and $F \in \mathcal{F}(S)$, if $(x, F) \in C$ then F does not rely on x.

For a subset $R \subseteq S$, $\mathcal{F}_0(R)$ will denote the set of $F \in \mathcal{F}(R)$ having no $x \in R$ with $(x, F) \in C$; that is, $\mathcal{F}_0(R)$ is the set of regions over R which do not conflict with any object in R.

Clarkson and Shor [CS89] analyzed the incremental computation of $\mathcal{F}_0(S)$. In the general step, $\mathcal{F}_0(R)$ for some subset $R \subseteq S$ is already available, a random element $x \in S \setminus R$ is chosen, and $\mathcal{F}_0(R \cup \{x\})$ is constructed from $\mathcal{F}_0(R)$.

Let $(x_1, \ldots, x_j)$ be a sequence of pairwise distinct elements of S, and R_j the set $\{x_1, \ldots, x_j\}$. Let $R_0 = \{\}$, the empty set. The *history* $H = H(x_1, \ldots, x_r)$ for insertion sequence $(x_1, \ldots, x_r)$ is defined as $H = \bigcup_{1 \leq i \leq r} \mathcal{F}_0(R_i)$. Let Π_S be the set of permutations of S. For $\pi = (x_1, \ldots, x_n) \in \Pi_S$, $H_r(\pi)$ or simply H_r denotes the history $H(x_1, \ldots, x_r)$.

First, some simple facts about random permutations, whose proofs we leave to the reader:

Lemma 1 *If $\pi = (x_1, \ldots, x_n)$ is a random permutation of S, then R_j is a random subset of S of size j, $(x_1, \ldots, x_j)$ is a random permutation of R_j, x_j is a random element of R_j, and if δ is a (fixed) permutation, then $\pi\delta$ is a random permutation.*

We are now ready for an average case analysis of randomized incremental constructions. All expected values are computed with respect to a random ordering $(x_1, \cdots, x_n) \in \Pi_S$ of the objects in S.

For subset $R \subseteq S$, $r = |R|$, and distinct objects $x, y \in R$, let

$$\begin{aligned}
deg(x, R) &= |\{F \in \mathcal{F}_0(R); x \text{ supports } F\}| \\
pdeg(x, y, R) &= |\{F \in \mathcal{F}_0(R); x \text{ and } y \text{ support } F\}| \\
c(R) &= \frac{1}{r} \sum_{x \in R} deg(x, R) \\
p(R) &= \frac{1}{r(r-1)} \sum_{(x,y) \in R^2} pdeg(x, y, R) \ .
\end{aligned}$$

We call $deg(x, R)$ the *degree* of x in R, $pdeg(x, y, R)$ the *degree* of the ordered pair (x, y) in R, $c(R)$ the *average degree* of a random object in R and $p(R)$ the *average pair degree* of a random pair of objects in R. Of course, $p(R)$ is only defined for $r \geq 2$.

For integer r, $1 \leq r \leq n$, let

$$c_r = E[c(R)] = \sum_{R \subseteq S, |R| = r} c(R) / \binom{n}{r}$$

and

$$p_r = E[p(R)] = \sum_{R \subseteq S, |R| = r} p(R) / \binom{n}{r}$$

be the expected average degree and pair degree for random $R_r \subset S$, and let

$$f_r = \sum_{R \subseteq S, |R| = r} |\mathcal{F}_0(R)| / \binom{n}{r}$$

be the expected number of conflict-free regions of $\mathcal{F}(R)$, with respect to random R_r. Note that $c_1 = f_1$. It will be convenient to adopt the convention that $c_j = p_j = f_j = 0$ for $j < 1$ or $j > n$, and (almost always) convenient to adopt the convention that $p_1 = f_1$.

Lemma 2 *The expectations c_r, p_r, and f_r satisfy $c_r \leq b f_r / r$, and for $r > 1$, $p_r \leq b(b-1) f_r / r(r-1)$, with equality if $\mathcal{F}(S)$ is uniform.*

Proof: For every region $F \in \mathcal{F}(S)$ there are at most b objects and at most $b(b-1)$ ordered pairs of objects which support F, and exactly as many if $\mathcal{F}(S)$ is uniform. ∎

Theorem 3 *Let C_r be the expected size of history H_r. Then $C_r = \sum_{j \leq r} c_j$.*

Proof: H_0 is empty and hence $C_0 = 0$. For $r \geq 1$ the number of elements of H_r which are not already elements of H_{r-1} is equal to $deg(x_r, R_r)$. Since R_r is a random subset of S of size r and x_r is a random object in R, we have

$$E[deg(x_r, R_r)] = E[c(R)] = c_r.$$

∎

In §4 we will strengthen Theorem 3 and prove a tail estimate for $|H_n|$.

Theorem 4 *The expected number of regions in H_{r-1} which are in conflict with x_r is $-c_r + \sum_{j \leq r} p_j$.*

Proof: Let X be the number of regions $F \in H_{r-1}$ with $(x_r, F) \in C$. Let $H = H_{r-1} = H(x_1, \ldots, x_{r-1})$ and $H' = H(x_r, x_1, \ldots, x_{r-1})$, i.e., in H' we "pretend" that x_r was put in first. We have

$$|H| + |H' \backslash H| = |H'| + |H \backslash H'|,$$

which holds for any two finite sets. Now $X = |H \backslash H'|$ since $H \backslash H'$ is the set of regions in H which conflict with x_r. On the other hand, $H' \backslash H$ comprises regions supported by x_r; to count these regions, we count the number that appear when x_j is inserted. That is, letting $R'_j = R_j \cup \{x_r\}$, for each region $F \in H' \backslash H$ there is exactly one $j \geq 1$ such that $F \in \mathcal{F}_0(R'_j)$ and x_j supports F. Such a region is also supported by x_r, and so for given j the number of regions we count is $pdeg(x_r, x_j, R'_j)$. Putting these observations together,

$$X = |H| - |H'| + |\mathcal{F}_0(\{x_r\})| + \sum_{1 \leq j \leq r-1} pdeg(x_r, x_j, R'_j),$$

and so

$$EX = E|H| - E|H'| + E|\mathcal{F}_0(\{x_r\})| + \sum_{1 \leq j \leq r-1} E[pdeg(x_r, x_j, R'_j)]$$

We have $E|H| = C_{r-1}$ by Theorem 3, and $E|H'| = C_r$ by Theorem 3 and Lemma 1. Also $E|\mathcal{F}_0(\{x_r\})| = f_1 = p_1$ by convention, and $E[pdeg(x_r, x_j, R'_j)] = p_{j+1}$, since $R'_j = R_j \cup \{x_r\}$ is a random subset of S of size $j+1$ and x_r and x_j are random elements of this subset. ∎

The following estimates are also useful.

Lemma 5 *For $j \leq r$ the following holds:*

(a) *The expected number of regions in $\mathcal{F}_0(R_{j-1})$ in conflict with x_r is $f_{j-1} - f_j + c_j$.*

(b) *The expected number of regions in $\mathcal{F}_0(R_{j-1})$ supported by x_{j-1} and in conflict with x_r is at most $b(f_{j-1} - f_j + c_j)/(j-1)$, with equality if $\mathcal{F}(S)$ is uniform.*

Proof:

(a) We have

$$\begin{aligned} \mathcal{F}_0(R_{j-1} \cup \{x_r\}) &= \mathcal{F}_0(R_{j-1}) \backslash \{F \in \mathcal{F}_0(R_{j-1}); (x_r, F) \in C\} \\ &\cup \{F \in \mathcal{F}_0(R_{j-1} \cup \{x_r\}); x_r \text{ supports } F\} \end{aligned}$$

and hence the desired quantity is

$$E|\mathcal{F}_0(R_{j-1})| - E|\mathcal{F}_0(R_{j-1} \cup \{x_r\})| + E|\{F \in \mathcal{F}_0(R_{j-1} \cup \{x_r\}); x_r \text{ supports } F\}|$$

$$= f_{j-1} - f_j + c_j$$

(b) x_{j-1} is a random element of R_{j-1}. Hence a region considered in part (a) is supported by x_{j-1} with probability at most $b/(j-1)$.

∎

Summation of the bound in Lemma 5b for j from 1 to $r-1$ gives an alternative bound on the expected number of regions in H_{r-1} which conflict with x_r.

The *conflict history* $G = G_n = G(\pi)$ for insertion sequence $\pi = (x_1, \cdots, x_n)$ is the relation $C \cap (S \times H_n)$. We may also describe this relation as a bipartite graph, with an edge between object $x \in S$ and region $F \in H_n$ when x and F conflict. The conflict history corresponds to the union (over time) of the conflict graphs in [CS89]. We use $|G|$ to denote the size of the conflict history, i.e., the number of pairs in it.

Theorem 6 *The expected size of the conflict history is*

$$E|G| = -C_n + \sum_j (n - j + 1) p_j$$

Proof: Theorem 4 counts the expected number of edges incident to node $x_r \in S$. The claim follows by summation over r. ∎

We next turn to random deletions. For $\pi = (x_1, \ldots, x_n) \in \Pi_S$ and $r \in [1\,..\,n]$, let

$$\pi \backslash r = (x_1, \ldots, x_{r-1}, x_{r+1}, \ldots, x_n),$$

and $\pi \backslash r = \pi$ for $r \notin [1\,..\,n]$. We bound the expected size of the difference between $H(\pi)$ and $H(\pi \backslash r)$ and between $G(\pi)$ and $G(\pi \backslash r)$ for random $\pi \in \Pi_S$ and random $r \in [1\,..\,n]$.

Theorem 7

$$\frac{1}{n!n} \sum_{\pi \in \Pi_S} \sum_r |H(\pi) \oplus H(\pi \backslash r)| \leq 2b \frac{C_n}{n} - c_n,$$

with equality if $\mathcal{F}(S)$ is uniform.

Proof: For finite sets A and B,

$$|B \oplus A| = |A| - |B| + 2|B \backslash A|,$$

and so for $H = H(\pi)$ and $H(\pi \backslash r)$,

$$|H \oplus H(\pi \backslash r)| = |H(\pi \backslash r)| - |H| + 2|H \backslash H(\pi \backslash r)|.$$

The set $H \backslash H(\pi \backslash r)$ comprises the regions in H supported by x_r. By Theorem 3, $E|H| = C_n$, and any $F \in H$ is supported by no more than b objects, with equality if $\mathcal{F}(S)$ is uniform. Therefore on average the random $x_r \in S$ supports no more than bC_n/n regions of H. By Theorem 3 and Lemma 1, we have $E|H(\pi \backslash r)| = C_{n-1}$, and the theorem follows since $C_{n-1} - C_n = -c_n$ by definition. ∎

Theorem 8

$$\begin{aligned} E|G(\pi \backslash i) \backslash G(\pi)| &= \frac{1}{n!n} \sum_{\pi \in \Pi_S} \sum_i |G(\pi \backslash i) \backslash G(\pi)| \\ &\leq c_n - (b+1)C_n/n + \sum_j bp_j - \sum_j (b+1)(j-1)p_j/n, \end{aligned}$$

with equality if $\mathcal{F}(S)$ is uniform.

Proof: Letting $G = G(\pi)$, we have

$$|G(\pi \backslash i) \backslash G| = |G(\pi \backslash i)| - |G| + |G \backslash G(\pi \backslash i)|,$$

and by linearity of expectation,

$$E|G(\pi \backslash i) \backslash G| = E|G(\pi \backslash i)| - E|G| + E|G \backslash G(\pi \backslash i)|.$$

Theorem 6 gives an expression for $E|G|$, and together with Lemma 1 it gives a similar expression for $E|G(\pi\backslash i)|$, yielding

$$E|G(\pi\backslash i)\backslash G| = E|G\backslash G(\pi\backslash i)| + c_n - \sum_j p_j.$$

(Alternatively, note that $E|G| - E|G(\pi\backslash i)|$ is the expected number of regions of H_n conflicting with x_i, and use Theorem 4.) We need to find $E|G\backslash G(\pi\backslash i)|$. A pair (x, F) is in $G\backslash G(\pi\backslash i)$ if it is in G and either $x_i = x$ or $x_i \in F$. At most $b+1$ choices of x_i allow this, for any $(x, F) \in G$, and so $E|G\backslash G(\pi\backslash i)| \leq (b+1)E|G|/n$, with equality if $\mathcal{F}(S)$ is uniform. The result follows using Theorem 6 and easy manipulations. ∎

In the convex hull algorithm of §3, the conflicts of $G(\pi\backslash i)\backslash G(\pi)$ are not quite all those examined when deleting x_i. The following bound will also be useful.

Lemma 9 *Let I be the set of conflicts of the form (x_j, F) with $j > i$ and $F \in \mathcal{F}_0(R_{i-1}) \setminus \mathcal{F}_0(R_i)$. Then for random $\pi \in \Pi_S$ and random $i \in [1\,..\,n]$, $E|I| = (E|G| - E|H| + f_n)/n$.*

Proof: Let I_i denote the set I for x_i. Then $E|I| = \sum_i E|I_i|/n$, and since the I_i are disjoint, $E|I| = E|\cup_i I_i|/n$. For any conflict $(x_j, F) \in G$, either $F \in \mathcal{F}_0(R_{j-1})$, or there is exactly one $i < j$ such that $F \in \mathcal{F}_0(R_{i-1})\backslash\mathcal{F}_0(R_i)$. In the latter case, $(x_j, F) \in I_i$. To count the conflicts (x_j, F) with $F \in \mathcal{F}_0(R_{j-1})$, note that each $F \in H\backslash\mathcal{F}_0(S)$ appears this way exactly once. Thus $E|G| = E|\cup_i I_i| + E|H| - |\mathcal{F}_0(S)|$, from which the Lemma follows. ∎

3 Dynamic Convex Hulls

We apply the results of §2 to the problem of maintaining the convex hull in d-dimensional space under insertions and deletions of points. Let $X \subset \mathbb{R}^d$ be a set of points, which we assume to be in nondegenerate position: no $d+1$ lie in a common hyperplane. For $R \subseteq X$, let $\operatorname{conv} R$ denote the convex hull of R. We let $x_1, x_2, \ldots, x_n$ denote the points in X in the order of their insertion, and let R_i denote $\{x_1, \ldots, x_i\}$.

3.1 The Insertion Algorithm

To maintain the convex hull of R under insertions, we maintain a triangulation T of the hull: a simplicial complex whose union is $\operatorname{conv} R$. (A simplicial complex is a collection of simplices such that the intersection of any two is a face of each.) The vertices of the simplices of T are points of R. The triangulation is updated as follows when a point x is added to R: if $x \in \operatorname{conv} R$, and so is in some simplex S of T, leave T as it was. If $x \notin \operatorname{conv} R$, then for every facet F of the hull of R visible to x, add to T the simplex $S(F,x) = \operatorname{conv}(F \cup \{x\})$. Call F the *base* facet and x the *peak* vertex of the simplex. A facet is *visible* to x or x*-visible* just when $S(F,x)$ meets the hull only at F. We may also say, for x-visible F, that x is visible to F, and they *see* each other. Use T_r to denote the triangulation after the insertion of $x_1, x_2, \ldots, x_r$.

This process is called triangulation by “placing” [Ede87]. It should be clear that the stated conditions on the triangulation are preserved. (When $r \leq d+1$, we simply maintain a single $(r-1)$–dimensional simplex.) It will be convenient to extend the triangulation so that facets of the current hull are also base facets of simplices; this gives a uniform representation. The peak vertex of these simplices is a “dummy” that in effect is visible to all current facets; we use $\overline{O}$ to denote this dummy vertex and we use O to denote a point inside the first full-dimensional simplex created, when $r = d+1$. (Here we use the assumption of nondegenerate position.) Call the first full-dimensional simplex the *origin simplex.* (In the terminology of ”two-sided space” [Sto87] O and $\overline{O}$ could be called the origin and anti-origin respectively: while the origin sees no facets of the current hull of R, the anti-origin sees all of them.) We use T to also denote the extended triangulation. To carry the uniformity even further, we designate the vertex x_{d+1} the peak of the origin simplex and call its opposite facet the base of the origin simplex. In this way, there are $d+2$ simplices in the (extended) triangulation when the first full-dimensional simplex is created: the origin simplex and $d+1$ simplices with peak $\overline{O}$. One facet of the origin simplex (better: its two sides) is base facet of two simplices and all other facets of the origin simplex are base facet of one simplex.

Two simplices of T are *neighbors* if they share a facet. The neighbor relation defines the neighborhood graph on the set of simplices. Call a neighbor of some simplex S and a vertex x of S *opposite to each other*, if the common facet does not contain x. In an implementation, we propose to store the directed version of the neighborhood graph augmented by information which supports the following operations in constant time: identification of the neighbor of a simplex sharing the base facet, identification of the peak vertex of a simplex, and identification of the vertex opposite to a facet. We also store for each simplex the equation of the hyperplane supporting the base facet of the simplex. The equation is normalized such that the peak lies in the positive half-space.

We discuss next two *search* methods for finding the x-visible current facets of $\operatorname{conv} R$.

Here is one method: locate x in T by walking along the segment $\overline{Ox}$ beginning at O. If this walk enters a simplex whose peak vertex is the anti-origin, then an x-visible current facet has been found. Otherwise, a simplex of T containing x has been found, showing that $x \in \operatorname{conv} R$. In the former case, find all x-visible hull facets by a search of the simplices incident to the anti-origin. These simplices form a connected set in the neighborhood graph. We call this search method the *segment-walking* method.

Another search method is the following: starting at the origin simplex and the simplex sharing its base facet explore simplices according to the rule: if a simplex has an x-visible base facet, search its neighbors (not including the neighbor that shares the base facet). Here we say that a base facet F is x-visible if that was true (in the previous sense) at the time that F was a current hull facet. This search procedure reaches all x-visible current hull facets, i.e., all simplices $S(F, \overline{O})$ with x-visible base facet F, since the base facets of all simplices traversed in the segment-walking search method are x-visible. We call this search scheme the *all-visibilities* method.

We finally turn to the update procedure. At this point, we have found the current hull facets seeing x, in the form of the simplices whose base facets see x and with the anti-origin as their peak vertex. Let $\mathcal{V}$ be the set of such simplices. Now we update T by altering these simplices, and creating some others. The alteration is simply to replace the anti-origin with x in every simplex in $\mathcal{V}$.

The new simplices correspond to new hull facets. Such facets are the hull of x and a horizon ridge f; a *horizon ridge* is a $(d-2)$–dimensional face of $\operatorname{conv} R$ with the property that exactly one of the two incident hull facets sees x. Each horizon ridge f gives rise to a new simplex A_f with base facet $\operatorname{conv}(f \cup \{x\})$ and peak $\overline{O}$. For each horizon ridge of $\operatorname{conv} R$ there is a non-base facet G of a simplex in $\mathcal{V}$ such that x does not see the base facet of the other simplex incident to the facet G. Thus the set of horizon ridges is easily determined.

It remains to update the neighbor relationship. Let $A_f = S(\operatorname{conv}(f \cup \{x\}), \overline{O})$ be a new simplex corresponding to horizon ridge f. In the old triangulation (before adding x) there were two simplices V and N incident to the facet $\operatorname{conv}(f \cup \{\overline{O}\})$; $V \in \mathcal{V}$ and $N \notin \mathcal{V}$. In the updated triangulation V has peak x. The neighbor of A_f opposite to x is N and the neighbor opposite to $\overline{O}$ is (the updated version) of V. Now consider any vertex $q \in f$ and let $\mathcal{S} = \mathcal{S}_{f,q}$ be the set of simplices with peak x and including $\operatorname{vertex}(f) \setminus \{q\} \cup \{x\}$ in their vertex set; for a face f we use $\operatorname{vertex}(f)$ to denote the set of vertices contained in f. We will show that the neighbor of A_f opposite to q can be determined by a simple walk through $\mathcal{S}$. This walk amounts to a rotation about the $(d-2)$–face $\operatorname{conv}(\operatorname{vertex}(f) \setminus \{q\} \cup \{x\})$. Note first that $V \in \mathcal{S}$. Consider next any simplex $S = S(F, x) \in \mathcal{S}$. Th

3.2 Analysis of Insertions

The cost of adding a point to set R is the time needed to locate the point x in the triangulation T, plus the time needed to update the triangulation.

We need some additional notation. Let t_0 be the number of simplices visited by the walk along segment $\overline{Ox}$, let t_1 be the set of simplices with x-visible base facet, let t_2 be the set of simplices visited by the all-visibilites method, let t_3 be the number of simplices with peak x, and let t_4 be the number of new hull facets. Then $t_0 \leq t_1$, since the base facets of all simplices traversed by the segment-walking method see x, and $t_2 \leq (d+1) \cdot t_1$ since a simplex has $d+1$ neighbors.

In the segment-walking method the time spent on the walk is $O(d^2) \cdot t_0$, since given the entry point of segment $\overline{Ox}$ into a simplex S the exit point can be found in time $O(d^2)$; it takes time $O(d)$ per facet to compute the point of intersection, i.e., $O(d^2)$ altogether, and $O(d)$ time to select the first intersection following the entry point. The segment-walk determines the simplex containing x. All visible hull facets can then be determined in time $O(d^2) \cdot t_3$, since visiblity can be checked in time $O(d)$ per base facet and since a visible facet has at most d invisible neighbors. We define the search time of the segment-walking method to be $O(d^2) \cdot t_0 = O(d^2) \cdot t_1$ and include the $O(d^2) \cdot t_3$ term in the update time.

The search time for the all-visibilities method is $O(d) \cdot t_2 = O(d^2) \cdot t_1$, since $O(d)$ per simplex is needed for the visibility check and since the degree of the neighborhood graph is $d+1$.

Let's turn to the update time next. We need to alter t_3 simplices; this takes time $O(1) \cdot t_3$. For each new simplex we have to compute the equation of the hyperplane supporting the base facet. This takes time $O(d^3) \cdot t_4$, since solving the linear systems for the normal vectors requires $O(d^3)$ time per simplex (A factor of d can be removed using complicated rank-one updating techniques, if desired.). Finally, we need to update the neighbor relation. Let $\mathcal{S} = \mathcal{S}_{f,q}$ be defined as in the previous section. The walk through $\mathcal{S}$ takes time $O(d \cdot |\mathcal{S}|)$, since the neighbors in $\mathcal{S}$ of a simplex in $\mathcal{S}$ can be determined in time $O(d)$. Next observe, that a simplex $S = S(F, x) \in \mathcal{V}$ can belong to at most $d(d-1)$ different sets $\mathcal{S}_{f,q}$, since $f \setminus \{q\}$ can be obtained from F by deleting two vertices ($\binom{d}{2}$ choices) and since there are only two choices for q once $f \setminus \{q\}$ is fixed (Note that there are only two horizon ridges containing $f \setminus \{q\}$.). Thus the time to update the neighbor relation is $O(d^3) \cdot t_3$ and total update time is $O(d^3) \cdot (t_3 + t_4)$.

We next establish the connection to § 2. Our regions are half spaces. More formally, we have $b = d$ and $\mathcal{F}(X)$ contains two copies of each subset $\{x_1, \ldots, x_d\} \subseteq X$ of cardinality d. These two copies are identified with the two open half-spaces defined by the hyperplane through points $x_1, x_2, \ldots, x_d$. A point x is said to conflict with a half-space if it is contained in the half-space. In this way, for $|R| \geq d+1$ the regions in $\mathcal{F}_0(R)$ correspond precisely to the facets of the convex hull of R (recall that we assume our points to be in general position) and a facet F of $\operatorname{conv} R$ is visible from $x \notin R$ if x conflicts with the half-space supporting the facet. Also $|\mathcal{F}_0(R)| = 2$ if $|R| = d$, $\mathcal{F}_0(R) = \emptyset$ for $|R| < d$, and $\mathcal{F}(X)$ is uniform. Using the notation of §2, we therefore have $f_r = 0$ for $r < d$ and $f_d = 2$; for $r > d$, f_r is the expected number of facets of $\operatorname{conv} R$ for random subset $R \subseteq X$ with $|R| = r$.

Theorem 10 (a) *The expected number of simplices of T_r is $C_r = \sum_{j \leq r} df_j/j$.*

(b) *The expected search time for x_r, using either search method, is $O(d^2)$ times*

$$-c_r + \sum_{2\leq j\leq r} p_j = -\frac{d}{r} f_r + \sum_{2\leq j\leq r} \frac{d(d-1)}{j(j-1)} f_j.$$

(c) *The expected time to construct the convex hull of n points using either search method is*

$$O(d^3)\sum_j \frac{d}{j} f_j + O(d^3)\sum_j \frac{d(d-1)}{j(j-1)}(n-j+1)f_j = O(d^5)\sum_j \frac{nf_j}{j(j-1)}.$$

Proof:

(a) Each simplex has a base facet, and so the bound follows from Theorem 3 and Lemma 2.

(b) ¿From the above discussion, we need to find t_1, the expected number of facets that are x_r–visible. The expected number of visible facets is $-c_r + \sum_{j\leq r} p_j$, by Theorem 4.

(c) The work per simplex of T_n is $O(d^3)$, as discussed above. The bound follows, using (a) and summing the bound of (b) over r.

∎

Since $f_r = O(r^{\lfloor d/2 \rfloor})$ in the worst case, the running time is $O(n \log n)$ for $d \leq 3$, and $O(n^{\lfloor d/2 \rfloor})$ for $d \geq 4$. We note also that for many natural probability distributions, the expected complexity of the hull of random points satisfies $f_r = O(r)$ for fixed d. For such point sets, our algorithm requires $O(n \log n)$ expected time.

3.3 The Deletion Algorithm

The global plan is quite simple. When a point is deleted from R, we change the triangulation T so that in effect x was never added. This is in the spirit of § 2. The effect of the deletion of x on the triangulation T is easy to describe. All simplices having x as a vertex disappear (If x is not a vertex of T then T does not change). The new simplices of T resulting from the deletion of x all have base facets visible to x, with peak vertices inserted after x. These are the simplices that would have been included had x not been inserted into R. Let $R(x)$ be the set of points of R that are contained simplices with vertex x, and also inserted after x. We will, in effect, reinsert the points of $R(x)$ in the order in which they were inserted into R, constructing only those simplices that have bases visible to x. On a superficial level, this describes the deletion process. The details follow.

Let $\pi = (x_1, \ldots, x_n)$ be the insertion order and assume that $x = x_i$ is deleted. We first characterize the triangulation $T(\pi \setminus i)$.

Lemma 11 *Assume that $x = x_i$ is a vertex of T. The triangulation $T(\pi \setminus i)$ can be obtained from the triangulation $T(\pi)$ as follows:*

1. *Set $T = T(\pi)$.*

2. *Remove all simplices having x_i as a vertex.*

3. $k \leftarrow i + 1$;
   while $k \leq n$
   do (* *invariant A holds here* *)
       for *all facets F of* $\mathrm{conv}(R_{k-1} \setminus \{x_i\})$ *visible to x_i and x_k*
       do *add $S(F, x_k)$ to T* **od**;
       $k \leftarrow k + 1$
   od
   (* *invariant A holds here with $k = n + 1$* *)

Proof: It suffices to show that the following statement A is an invariant of the **while**–loop.

$$A: \quad T = T(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k-1}) \cup \{S; S = S(F, x_j) \text{ is a simplex in } T(\pi \setminus i) \text{ with peak } x_j,\ j \geq k, \text{ and } x_i\text{–invisible base facet } F\}$$

Assume $k = i + 1$ first. All simplices in $T(x_1, \ldots, x_{i-1})$ clearly belong to T. Consider next a simplex $S = S(F, x_j)$ in $T(\pi \setminus i)$ with peak x_j, $j \geq k = i + 1$, and base facet F invisible to x_i. Then F is a facet of $\mathrm{conv}(R_{j-1} \setminus \{x_i\})$ and, since F is invisible to x_i, also a facet of $\mathrm{conv}\, R_{j-1}$. Thus S belongs to $T(\pi)$ and is not removed in step 2. Therefore, $S \in T$.

Conversely, let $S = S(F, x_j)$ be any simplex of T. If $j < i$ then clearly $S \in T(x_1, \ldots, x_{i-1})$. If $j > i$ then F is a facet of $\mathrm{conv}\, R_{j-1}$ and x_i is not a vertex of F (since S is not removed). Thus F is a facet of $\mathrm{conv}(R_{j-1} \setminus \{x_i\})$ and hence $S \in T(\pi \setminus i)$. Also F is not visible from x_i. This completes the case $k = i + 1$.

For $k > i + 1$, we only have to observe that $T(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k-1})$ can be obtained from $T(x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{k-2})$ by adding all simplices $S(F, x_{k-1})$ where F is a facet of $\mathrm{conv}(R_{k-2} \setminus \{x_i\})$ visible to x_{k-1}. If F is also visible to x_i then the loop body adds S to T, if F is not visible to x_i then S is already in T by the induction hypothesis. ∎

Lemma 11 characterizes the new simplices added to T. We show next that only points in $R(x)$ can contribute new simplices (Lemma 12) and that the set of facets of $\mathrm{conv}(R_{k-1} \setminus \{x_i\})$ visible from x_i can be easlily maintained (Lemma 13).

Lemma 12 *Let $i < k$ and assume that $x = x_i$ is a vertex of T. Then $y = x_k \in R(x_i)$ if some facet F of $\operatorname{conv}(R_{k-1} \setminus \{x\})$ is visible to x and y.*

Proof: The hyperplane supporting F separates $\operatorname{conv}(R_{k-1} \setminus \{x\})$ from y and x and hence y is not the convex combination of points in $R_{k-1} \setminus \{x\}$. If $y \in \operatorname{conv} R_{k-1}$ then y is the convex combination of points in R_{k-1} and therefore the simplex of T containing y must have x as a vertex. Thus $y \in R(x)$. If $y \notin \operatorname{conv} R_{k-1}$ then some facet G of $\operatorname{conv} R_{k-1}$ that contains x must be visible from y (e.g. one that intersects the line segment joining y with some point of F, which, being visible from x, is not a facet of $\operatorname{conv} R_{k-1}$). But now $S(G;x)$ is a simplex of T, and hence $y \in R(x)$. ∎

Lemma 13 *For $y \in R(x)$ let $B(y) = \{\operatorname{conv}(f \cup \{y\}); S(f \cup \{x_i\}, y)$ is a simplex of $T\}$. Step 3 in Lemma 11 may be replaced by:*

$B \leftarrow$ *set of facets of* $\operatorname{conv} R_{i-1}$ *visible to* x;
for *all* $y \in R(x)$ *in ascending insertion order*
do (* *let* $y = x_k$; *then* B *is the set of facets of* $\operatorname{conv}(R_{k-1} \setminus \{x_i\})$ *visible from* x_i *);
  $VB \leftarrow \{F; F \in B; F$ *is* y*-visible* $\}$
  $NB \leftarrow \{F; F$ *is a non-base facet of* $S(G, x_k)$ *with* $G \in VB$, F x_i*-visible and* $f = F \cap G$
    a horizon ridge of $\operatorname{conv}(R_{k-1} \setminus \{x_i\})\}$
  $B \leftarrow (B \setminus VB) \cup NB \cup B(y)$
od

Proof: We only need to verify the invariant. By Lemma 12 B changes only when a point $y = x_k \in R(x)$ is reinserted. Let B be the set of x_i-visible facets of $\operatorname{conv}(R_{k-1} \setminus \{x_i\})$ and let B' be the set of x_i-visible facets of $\operatorname{conv}(R_k \setminus \{x_i\})$. Then $B' = (B' \cap B) \cup (B' \setminus B)$. Observe next that $B' \cap B = B \setminus VB$ and that each facet $F \in B' \setminus B$ is incident to x_k. Thus $f = \operatorname{conv}(f \cup \{x_k\})$ for some horizon ridge f of $\operatorname{conv}(R_{k-1} \setminus \{x_i\})$. Let G be the base facet of the simplex having F as a facet. Then $f = F \cap G$. Also, if G is x_i-visible then $F \in NB$ and if G is not x_i-visible then $F \in B(y)$. ∎

We are now ready for the algorithmic details.

To handle deletions, we must augment our data structure slightly. We assume that each point can access some simplex containing it in constant time and that every simplex knows the set of points contained in it.

Check first, whether x is a vertex of the simplex pointed to by x. If not, x is removed and we are done. If so, construct the set $R(x)$ by inspection of all simplices incident to x. This takes time proportional to d times $|R(x)|$ plus the number of simplices with peak x. (Note that a simplex which has x in its base facet contributes its peak to $R(x)$ and that a simplex has at most $d+1$ neighboring simplices).

Next, sort the points in $R(x)$ by the time of insertion. This takes time $O(\min\{n, |R(x)| \log\log n\})$, where the former bound is obtained by bucket sort and the latter bound comes from the use of bounded ordered dictionaries ([vKZ77, MN90]). Also, collect for each point $y \in \{p\} \cup R(x)$ the set of simplices $\mathcal{S}(y)$ with peak y and also having x as a vertex. Next remove all simplices incident to x from T (cf. step 2 in Lemma 11). The set B (cf. Lemma 13) is initialized to the set of base facets of simplices in $\mathcal{S}(x)$. The neighborhood graph on the set B is given by the incidence information of the simplices in $\mathcal{S}(x)$. The set $B(y)$, for $y \in R(x)$, and its neighborhood graph is easily determined from the set $\mathcal{S}(y)$.

The points in $R(x)$ are now processed in the insertion order. Consider $y = x_k \in R(x)$. We first determine all facets in B visible from y. To do so we distinquish cases.

Assume first that y is a vertex. For each simplex $S \in \mathcal{S}(y)$ let $f(S)$ be the ridge with all the vertices of S but x and y. $f(S)$ is a ridge of $\operatorname{conv} R_{k-1}$ not incident to x and hence a ridge of $\operatorname{conv}(R_{k-1} \setminus \{x\})$. Also, $f(S)$ is visible from x and y and hence a ridge of B. Thus if we maintain the correspondence (via a dictionary for $(d-2)$–tuples representing ridges) between the ridges of removed simplices and ridges of B, we can find the set $\{f(S); S \in \mathcal{S}(y)\}$ in B in time proportional to d times the number of simplices in $\mathcal{S}(y)$. Starting in the facets of B incident to these ridges a simple graph search of B's neighborhood graph determines all y-visible facets of B in time proportional to d times their number. This is the content of

Lemma 14 *Let y be a vertex. Then all y-visible facets of B can be reached from a ridge in $\{f(S); S \in \mathcal{S}(y)\}$ in the neighborhood graph of B.*

Proof: Let $\mathcal{G}$ be the facet graph of $\operatorname{conv} R_{k-1}$ and let $\mathcal{G}_x$ and $\mathcal{G}_y$ be the parts of $\mathcal{G}$ formed by the facets and ridges of $\operatorname{conv} R_{k-1}$ that are visible from x and y, respectively. Note that $\mathcal{G}_x$ as well as $\mathcal{G}_y$ is connected (in the topological sense and in the graph theoretic sense). Moreover note that $\mathcal{G}_x$ is nothing but B. The set $\{f(S); S \in \mathcal{S}(y)\}$ comprises exactly all ridges in $\mathcal{G}_y$ for which exactly one of the containing facets is in $\mathcal{G}_x$. Connectedness of $\mathcal{G}_y$ now ensures that all facets and ridges that are in $\mathcal{G}_x$ and in $\mathcal{G}_y$ can be reached from some ridge in $\{f(S); S \in \mathcal{S}(y)\}$. ∎

For non–vertices y one has to work harder. We first show how to identify a single facet in B visible from y and then argue that a graph seach determines all y-visible facets in B. Assume first that y is contained in a simplex $S \in \mathcal{S}(x)$. Let $S = (F, x)$, and let O be the intersection of F with the line through x and y. Locate y by a walk along $\overline{Oy}$ starting at O. Assume next that y is contained in a simplex $S \in \mathcal{S}(z)$ for some $z \in R(x)$. The ridge $f(S)$ of S with all vertices but x and y is a ridge of B when point z is reinserted and hence the facet spanned by $f(S)$ and z is added to T when point z is reinserted. Let O be the intersection of that facet with the line through x and y. Locate y by a walk along $\overline{Oy}$ starting at O.

Lemma 15 *The walk along $\overline{Oy}$ traverses only newly constructed simplices whose base facet is y–visible.*

Proof: The line segment $\overline{Oy}$ is contained in the simplex S. This implies that $\overline{Oy}$ traverses only new simplices. Let S' with base facet G be a simplex traversed. Then G is x–visible. Since S' is intersected by $\overline{Oy}$ and G is visible from every point in S', G must be visible from either O or y. But O-visibility and x-visibility of G and the fact that $y \in \overline{Ox}$ implies y-visibility of G. Thus G is y-visible. ∎

At this point we have found one y-visible facet in B.

Lemma 16 *Let $y = x_k$ be a non-vertex. Then all y-visible facets of* $\text{conv}(R_{k-1} \setminus \{x_i\})$ *are also x_i-visible.*

Proof: Assume that there are y-visible facets of $\text{conv}(R_{k-1} \setminus \{x_i\})$ and let F be one of them. Then $x_i \notin \text{conv}\, R_{k-1}$ and there is a facet G of $\text{conv}(R_{k-1} \setminus \{x_i\})$ such that $y \in S(G, x_i)$. Then the hyperplane supported by F separates G and y. Thus x_i sees F. ∎

The set of y-facets of $\text{conv}(R_{k-1} \setminus \{x_i\})$ is neighbor-connected and is identical to the set of y-visible facets in B. Thus a graph search on B finds all y-visible facets in B.

If the points in S are in convex position, as when computing Voronoi diagrams in one less dimension [Ede87, section 1.8], all points y are vertices and hence the search time is well covered by the time to construct new simplices.

In either case we have now identified the set of y-vertices in B. We now construct new simplices, add $B(y)$ to B and update the neighborhood graph on B.

Having reinserted the points in $R(x)$ the cavity created by the removal of x is filled. A traversal of the new simplices and the boundary of the cavity allows to match (using a dictionary for facets) the new simplices with the old simplices sharing a common facet. This completes the update step.

In summary, the cost of the removal of x is bounded by the sum of the following quantities:

(1) $\min(n, |R(x)| \log\log n)$,

(2) $O(d^3)$ times the number of removed and newly constructed simplices,

(3) $O(d^2)$ times the sum over all points $y \in R(x)$ of the number of y–visible facets ever contained in B.

If the points are in convex position, then the sum of the first two quantities suffices.

3.4 Analysis of Deletions

We analyze the cost of a deletion under the assumption that the points were inserted in random order and that a random point is deleted.

Lemma 17 *The expected number of removed simplices is bounded by*

$$\sum_{i \leq n} d(d+1) f_i/(i \cdot n)$$

and the expected number of new simplices is no larger.

Proof: The expected number of simplices in $T(\pi)$ is C_n and the expected number of simplices in $T(\pi \backslash i)$ is C_{n-1} according to Theorem 10 and Lemma 1. Also each simplex of $T(\pi)$ has $d+1$ vertices and therefore the expected number of removed simplices is $(d+1)(C_n - f_n)/n$. The expected number of new simplices is thus $(C_{n-1} - f_{n-1}) - (C_n - f_n - (d+1)(C_n - f_n)/n)$ which is no larger than the number of removed simplices. The bound now follows from Theorem 3 and Lemma 2. ∎

Lemma 18 *The expected size of $R(x)$ is bounded by*

$$(d+1)\left(2 + d \sum_{i \leq n} f_i/(i \cdot n)\right).$$

Proof: Let $R_1(x)$ be the set of points $y \in R(x)$ which are vertices of T and let $R_2(x) = R(x) \setminus R_1(x)$. To bound $|R_1(x)|$, observe that $|R_1(x)|$ is at most d plus the number of destroyed simplices. Thus

$$E\left[|R_1(x)|\right] \leq d + \sum_{i \leq n} d(d+1) f_i/(i \cdot n).$$

To bound $|R_2(x)|$, observe that each non–vertex y is incident to exactly one simplex (recall that our points are in general position) and that x is the vertex of such a simplex with probability $(d+1)/n$. Thus

$$E\left[|R_2(x)|\right] \leq n(d+1)/n = (d+1).$$

∎

We next bound the sum over $y \in R(x)$ of the number of y–visible facets ever contained in B. Such a facet is either a y– and x–visible facet of $\operatorname{conv} R_{i-1}$ (recall that $x = x_i$) or a newly constructed base facet visible to y and x.

Lemma 19 (a) *The expected number of facets of* $\operatorname{conv} R_{i-1}$ *visible to* x_i *and* x_j *summed over* $j > i$ *and for random* i *is* $(E|G| - E|H| + f_n)/n$.

(b) *The expected number of new base facets visible to* x_j *summed over* $j > i$ *and for random* i *is* $E|G(\pi\backslash i) \setminus G(\pi)|$.

Proof: Part (a) follows from Lemma 9 and part (b) is obvious. ∎

Theorem 20 *The expected time to delete a random point from the convex hull of* n *points (constructed by random insertions) is*

$$O\left(\min\left\{n, \left(d + d^2 \sum_{i\le n} f_i/(i\cdot n)\right) \log\log n\right\} + d^5 \sum_{i\le n} f_i/(i\cdot n) + d^5 \sum_{2\le i\le n} f_i/(i(i-1))\right).$$

If the points are in convex position, then time

$$O\left(\min\left\{n, \left(d + d^2 \sum_{i\le n} f_i/(i\cdot n)\right) \log\log n\right\} + d^5 \sum_{i\le n} f_i/(i\cdot n)\right)$$

suffices.

Proof: This follows immediately from the summary at the end of § 3.3, Lemmas 17 to 19, and Theorems 3 and 8. ∎

We have $f_i = O(i^{\lfloor d/2\rfloor})$. A deletion from a convex hull in $\mathbb{R}^3$ therefore takes time $O(\log n)$ and a deletion from a Voronoi diagram in $\mathbb{R}^2$ takes time $O(\log\log n)$. For $d \ge 4$, a deletion from a convex hull in $\mathbb{R}^d$ and a Voronoi diagram in $\mathbb{R}^{d-1}$ takes time $O(n^{\lfloor d/2\rfloor -1})$. We note also that for many natural probability distributions, the expected complexity of the hull of random points satisfies $f_r = O(r)$ for fixed d. For such point sets, a random deletion requires $O(\log n)$ expected time.

4 A Tail Estimate for the Size of the History

In this section, we derive a tail estimate for the size of the history. We first prove a general lemma and then apply one of its consequences to obtain a tail estimate for the size of the history in randomized incremental constructions.

In the notation of §2, we want to study the random variable $X = \sum_j deg(y_j, R_j)$ for random permutations $\pi = (y_1, \ldots, y_n)$ of S, inducing the subsets $R_j = \{y_1, \ldots, y_j\}$. Let $p(x) = p_S(x)$ be the generating function of this random variable. By the following standard observation, we can use bounds on $p(x)$ to show that X is large only with low probability.

Fact 21 *If Z is a non–negative integer random variable with generating function $p(x)$, then for any $k \geq 0$*

$$\Pr[Z \geq k] \leq p(a)/a^k \qquad \text{for any } a \geq 1.$$

Suppose for some function $M(r)$ we have $b \cdot |\mathcal{F}_0(R)| \leq M(r)$ when $|R| = r$. Then we have the following bound on $p(x)$.

Claim 22 *For all $x > 1$ we have*

$$p(x) \leq p_n(x) := \prod_{1 \leq i \leq n} \left(1 + \frac{1}{i}(x^{M(i)} - 1)\right) .$$

Proof: We use induction on n, the size of S, looking at corresponding generating functions for subsets of S. The claim holds vacuously for $n = 0$.

For the random permutation π of S, we know that y_n is a random element of S, and so

$$p(x) = p_S(x) = \frac{1}{n} \sum_{y \in S} x^{deg(y,S)} p_{S \setminus \{y\}}(x).$$

Applying the inductive assumption to every $(n-1)$-element subset of S, we get

$$p(x) = \frac{p_{n-1}(x)}{n} \sum_{y \in S} x^{deg(y,S)}.$$

Since

$$\sum_{y \in S} deg(y, S) \leq b|\mathcal{F}_0(S)| \leq M(n),$$

the power sum is maximized for $x > 1$ when $deg(y, S) = M(n)$ for some $y \in S$ and the degrees of the other members of S are zero. Thus

$$p(x) \leq \frac{p_{n-1}(x)}{n}\left(x^{M(n)} + (n-1)\right) = \left(1 + \frac{1}{n}(x^{M(n)} - 1)\right)p_{n-1}(x) = p_n(x) .$$

∎

Theorem 23 *For any integer $M \geq 0$ and any real $x \geq 1$*

$$\Pr[X \geq M] \leq \frac{\prod_{1 \leq i \leq n}\left(1 + \frac{1}{i}(x^{M(i)} - 1)\right)}{x^M} \leq \frac{e^{\sum_{1 \leq i \leq n} \frac{1}{i}(x^{M(i)} - 1)}}{x^M} .$$

Proof: This follows from Fact 21 and Claim 22, using the inequality $1 + x \leq e^x$.

∎

Corollary 24 *If $M(i)/i$ is non–decreasing, then for all $c > 1$*

$$\Pr[X \geq cM(n)] \leq (1/e) \cdot (e/c)^c .$$

Proof: If $M(i)/i$ is non–decreasing, then for all $x \geq 1$ we have $\frac{1}{i}(x^{M(i)} - 1) \leq \frac{1}{n}(x^{M(n)} - 1)$ for each $i \leq n$. (The polynomial $\frac{1}{n}(x^{M(n)} - 1) - \frac{1}{i}(x^{M(i)} - 1)$ has a root at $x = 1$ and nonnegative derivative for $x \geq 1$.) Therefore

$$\Pr[X \geq cM(n)] \leq \frac{e^{\sum_{1\leq i\leq n} \frac{1}{i}(x^{M(i)}-1)}}{x^{cM(n)}} \leq \frac{e^{\sum_{1\leq i\leq n} \frac{1}{n}(x^{M(n)}-1)}}{x^{cM(n)}} = \frac{e^{x^{M(n)}-1}}{x^{cM(n)}} .$$

Now choose x such that $x^{M(n)} = c$. ∎

For many RICs, e.g., the construction of convex hulls (in any dimension) ([CS89] and this paper), Delauney triangulations ([GKS90]), abstract Voronoi diagrams ([MMO91]), trapezoidal diagrams for non–intersecting line segments ([CS89, Sei91]), spherical intersections ([CS89]) and the construction of a single face of an arrangement ([CEG$^+$91]), there is a function $M(r)$ such that $M(r)/r$ is non-decreasing, $b|\mathcal{F}_0(R)| \leq M(r)$ when $|R| = r$, and $M(r) \leq dC_r$ for some small constant d. In these situations, Corollary 24 bounds the probability that the size of the history exceeds its expected value by a constant factor.

The following Corollary of Theorem 23 may also be useful.

Corollary 25 *If $M(i) = m_0$ for all i, then $\Pr[X \geq cm_0H_n] \leq e^{-H_n(1+c\log(c/e))}$ for $c > 1$, where H_n is the n–th harmonic number.*

Proof: From Theorem 23 we get

$$\Pr[X \geq cm_0H_n] \leq \frac{e^{\sum_{1\leq i\leq n} \frac{1}{i}(x^{m_0}-1)}}{x^{cm_0H_n}} = \frac{e^{H_n(x^{m_0}-1)}}{x^{cm_0H_n}} .$$

Now choose x such that $x^{m_0} = c$ to obtain the desired result. ∎

5 A game related to some randomized incremental constructions

Seidel [Sei91] gave a randomized $O(n \log^* n)$ algorithm for the triangulation of simple polygons. Devillers [Dev] recently extended the approach to other problems, e. g., the construction of the Voronoi–diagram for the edges of a simple polygon. The idea behind the $O(n \log^* n)$ is as follows: When an object $x \in S - R$ is added to R in standard RIC, the object x traces through the history of the construction. This takes time $O(\log r)$ for the r–th object to be inserted (apply Theorem 4 with $f_j = O(j)$). On the other hand, in the two examples mentioned above, all conflicts between objects in $S-R$ and regions in $\mathcal{F}_0(R_i)$ can be computed in expected linear time. Seidel and Devillers therefore interrupt the standard algorithm at suitable breakpoints , say after the i–th insertion, and compute all conflicts between $S - R_i$ and $\mathcal{F}_0(R_i)$. The crucial observation is now that if object $x_k \in S - R_i$ knows its conflicts with the regions in $\mathcal{F}_0(R_i)$ then its conflicts with the regions in $\mathcal{F}_0(R_{k-1})$ can be computed in additional $O(\log(k/i))$ expected time; sum the bound in Lemma 5 for j between i and k to see that only $O(\log(k/i))$ additional conflicts exist on average. A suitable choice of breakpoints yields an $O(n \log^* n)$ algorithm. Can this approach yield linear time algorithms? The following game is supposed to shed some light on this question.

The game is played on a sequence of n balls. Initially, all balls have label 1 and color white. There are two players A and B who take turns. The game stops when all balls are black. In its r–th turn player A selects a white ball, turns it black and labels it r. The cost of this move is $\log(r/r_{old})$, where r_{old} is the previous label of the ball. In its turn, B performs one or more of the following moves: She selects an interval of balls and relabels all balls in the interval with the highest label occurring in the interval. The cost of the move is the length of the interval. A tries to maximize cost, B tries to minimize it.

The intended relationship to RIC is as follows: A ball is black if it belongs to R. The label of a ball is i if the ball knows its conflicts with the regions in $\mathcal{F}_0(R_i)$. A move of player A moves a ball from time r_{old} in the history to time r and a move of player B moves an interval of points to the latest time in history occurring in the interval. In the algorithms mentioned above, the interval is always the entire sequence of balls.

Let $L = \log^* n = \max\{i; \log^{(i)} n > 1\}$, $D_i = \log^{(i)} n$ for $1 \le i \le L$, $D_{L+1} = 1$, and $D_0 = n + 1$. Let $B_i = \lfloor n/D_i \rfloor$ for $0 \le i \le L + 1$.

Lemma 26 *Player B can keep the cost in $O(n \log^* n)$.*

Proof: B plays the following simple strategy. In its B_i–th turn, $1 \le i \le L$, B relabels the complete sequence of balls. The total cost of B's moves is $nL = n \log^* n$. The total cost of A's moves is

$$\le \sum_{0 \le i \le L} (B_{i+1} - B_i) \log(B_{i+1}/\max\{B_i, 1\}) = O(n \log^* n)$$

∎

Lemma 27 *Player A can force the cost into $\Omega(n \log^* n)$.*

Proof: We first describe the strategy of player A. A's game is divided into phases; the i–th phase, $1 \leq i \leq L+1$, consists of moves $B_{i-1}+1$ to B_i. In the i–th phase, A labels all multiples of D_i which are not multiples of D_{i-1}. We assume here that the balls are numbered 1 through n.

We show that the total cost of A's and B's moves in the i–th phase is $\Omega(n)$. Call a multiple of D_i interesting if A labels it by one of the moves $B_i/2+1$ to B_i. If for more than $1/2$ of the interesting balls the cost of A's move is $\log((B_i/2)/\max(1, B_{i-1}))$, then the total cost of A's moves in the i–th phase is $\Omega(B_i/2 \cdot \log(D_{i-1}/D_i)) = \Omega(B_i \cdot (D_i - D_{i+1})) = \Omega(n)$. Otherwise, more than half of the interesting points must have been relabeled in the i–th phase by a move of B, since all interesting points have label at most B_{i-1} at the beginning of the i–th phase. Since an interesting point has distance D_i from any point touched by A in the i–th phase, the total cost of B's moves must be at least $\Omega(B_i/2 \cdot D_i) = \Omega(n)$. In either case we have shown that the cost of a phase is $\Omega(n)$. Since there are $\log^* n$ phases, the lower bound follows. ∎

In Lemma 27, player A chooses balls so as to make the life for player B as difficult as possible. In RIC's objects are chosen randomly. Let us say that player A plays *randomly* if he always chooses a random white ball.

Lemma 28 *If A plays randomly, then the expected cost of the game is* $\Omega(n \log^* n)$.

Proof: Define the division into phases as in Lemma 27. At the end of the i–th phase there are B_i black balls. These balls from a random subset of $[1\,..\,n]$. In order to lower bound the expected cost of the i–th phase we change the rules of the game in B's favor.
At the end of the i–th phase, player B selects $B_i/2$ black balls and declares that A's moves in the i–th phase involving these balls are free of charge.
We now distinguish two cases. For the remaining $B_i/2$ balls which are black at the end of the i–th phase, either at least $B_i/4$ were relabeled by B before A selects the ball, or this is not the case. In the former case, the cost of B's moves is clearly lower bounded by the sum of the B smallest distances between black balls. The expected value of this sum is $\Omega(n)$. In the latter case, the cost of A's moves is $\Omega(n)$. ∎

References

[BDS+] J.D. Boissonnat, O. Devillers, R. Schott, M. Teillaud, and M. Yvinec. Applications of random sampling to on-line algorithms in computational geometry. *Discrete and Computational Geometry.* To be published. Available as Technical Report INRIA 1285. Abstract published in IMACS 91 in Dublin.

[CEG+91] B. Chazelle, H. Edelsbrunner, L.J. Guibas, M. Sharir, and J. Snoeyink. Computing a face in an arrangement of line segments. *2nd Ann. ACM-SIAM Symp. on Discrete Algorithms*, pages 441 – 448, 1991.

[CS89] K. L. Clarkson and P. W. Shor. Applications of random sampling in computational geometry, II. *Journal of Discrete and Computational Geometry*, pages 387–421, 1989.

[Dev] O. Devillers. Randomization yields simple $o(n \log^{\star} n)$ algorithms for difficult $\omega(n)$ problems. *International Journal on Computational Geometry and Aplications.* To be published. Full paper available as Technical Report INRIA 1412. Abstract published in the Third Canadian Conference on Computational Geometry 1991 in Vancouver.

[DMT91] O. Devillers, S. Meiser, and M. Teillaud. Fully dynamic Delaunay triangulation in logarithmic expected time per operation. In *WADS 91*, volume LNCS 519. Springer Verlag, 1991. Full version available as Technical Report INRIA 1349.

[Ede87] H. Edelsbrunner. *Algorithms in Combinatorial Geometry.* Springer Berlin-Heidelberg, 1987.

[GKS90] L.J. Guibas, D.E. Knuth, and M. Sharir. Randomized incremental construction of Delaunay and Voronoi diagrams. *Proc. of ICALP*, pages 414 – 431, 1990. also to appear in Algorithmica.

[MMO91] K. Mehlhorn, St. Meiser, and C. Ò'Dunlaing. On the construction of abstract voronoi diagrams. *Discrete Comput. Geom.*, 6:211 – 224, 1991.

[MN90] K. Mehlhorn and St. Näher. Bounded ordered dictionaries in $O(\log\log\ n)$ time and $O(n)$ space. *Information Processing Letters*, 35:183 – 189, 1990.

[Mul88] K. Mulmuley. A fast planar partition algorithm, I. *Proc. of the 29th FOCS*, pages 580–589, 1988.

[Mul91a] K. Mulmuley. Randomized multidimensional search trees: dynamic sampling. *Proc. ACM Symposium on Computational Geometry*, 1991.

[Mul91b] K. Mulmuley. Randomized multidimensional search trees: further results in dynamic sampling. *32nd IEEE FOCS*, pages 216 – 227, 1991.

[Mul91c] K. Mulmuley. Randomized multidimensional search trees: Lazy balancing and dynamic shuffling. *32nd IEEE FOCS*, pages 180 – 194, 1991.

[Sch91] O. Schwarzkopf. Dynamic maintenance of geometric structures made easy. *32nd IEEE FOCS*, pages 197 – 206, 1991.

[Sei90] R. Seidel. Linear programming and convex hulls made easy. *Proc. 6th Ann. ACM Symp. Computational Geometry*, pages 211 – 215, 1990.

[Sei91] R. Seidel. A simple and fast incremental randomized algorithm for computing trapezoidal decompositions and for triangulating polygons. *Computational Geometry: Theory and Applications*, 1:51 – 64, 1991.

[Sto87] A. J. Stolfi. Oriented projective geometry (extended abstract). *Proceedings of the 3rd Annual ACM Symp. on Computational Geometry*, pages 76–85, 1987.

[vKZ77] P. van Emde Boas, R. Kaas, and E. Zijlstra. Design and implementation of an efficient priority queue. *Math. Systems Theory*, 10:99 – 127, 1977.