Discrete Comput Geom 10:377–409 (1993)

Discrete & Computational
Geometry
© 1993 Springer-Verlag New York Inc.

An Optimal Convex Hull Algorithm in Any Fixed Dimension*

Bernard Chazelle

Department of Computer Science, Princeton University,
Princeton, NJ 08544, USA
chazelle@cs.princeton.edu

Abstract. We present a deterministic algorithm for computing the convex hull of n points in E^d in optimal $O(n \log n + n^{\lfloor d/2 \rfloor})$ time. Optimal solutions were previously known only in even dimension and in dimension 3. A by-product of our result is an algorithm for computing the Voronoi diagram of n points in d-space in optimal $O(n \log n + n^{\lceil d/2 \rceil})$ time.

1. Introduction

This paper provides a simple algorithm for computing the convex hull of n points in d-space deterministically in optimal $O(n^{\lfloor d/2 \rfloor})$ time, for $d > 3$. This result settles an open question of long standing. Optimal deterministic solutions were previously known only in even dimension and in dimension 3 [11], [17], [19]. By duality, the algorithm can be used to construct the full lattice structure of the feasible set of n linear constraints in optimal $O(n \log n + n^{\lfloor d/2 \rfloor})$ time. As a by-product, we obtain an algorithm for computing the Voronoi diagram of n points in optimal $O(n \log n + n^{\lceil d/2 \rceil})$ time, which is also a new result.

The convex hull problem has had an intriguing history. The cases $d = 2, 3$ have long been solved, but the problem has remained elusive in higher dimensions. More than a decade ago, Seidel [19] gave an optimal, $O(n \log n + n^{\lfloor d/2 \rfloor})$-time,

* This research was supported in part by the National Science Foundation under Grant CCR-9002352 and The Geometry Center, University of Minnesota, an STC funded by NSF, DOE, and Minnesota Technology, Inc. A preliminary version of this paper has appeared in "An optimal convex hull algorithm and new results on cuttings," *Proceedings of the 32nd Annual IEEE Symposium on the Foundations of Computer Science*, October 1991, pp. 29–38. The convex hull algorithm given in the present paper, although similar in spirit, is considerably simpler than the one given in the proceedings.

algorithm for the case where d is even. Later, he gave a slightly suboptimal, $O(n^{\lfloor d/2 \rfloor} \log n)$-time, solution for any fixed d [20]. On the probabilistic front, Clarkson and Shor [8] gave a Las Vegas incremental algorithm with optimal expected time. A variant of that method with a particularly simple analysis was proposed recently by Seidel [21].

Our convex hull algorithm is optimal in the worst case, but it is not output-sensitive. On that score the best-known general solution is due to Seidel [20]. Its running time is $O(n^2 + h \log h)$, where h is the facial complexity of the hull. The quadratic overhead can be cut down to $O(n^{2-(2/(\lfloor d/2 \rfloor + 1)) + \varepsilon})$, for any fixed $\varepsilon > 0$, as was recently shown by Matoušek [16]. Whether the complexity can be brought down to the lower bound of $\Omega(h + n \log h)$ remains an outstanding open problem. To date, only in two dimensions [12] and three dimensions [5], [8] are optimal output-sensitive algorithms known. (The algorithm in [8] is probabilistic; its optimal derandomization is given in [5].)

The convex hull algorithm given here can be regarded as an attempt to derandomize Clarkson and Shor's probabilistic incremental method [8]. It is nearly as simple as its randomized counterpart. Its complexity analysis, however, requires more effort. In the process we develop a deterministic version of a Monte Carlo integration method, which is likely to be useful elsewhere. Note that the details of the technique are needed for the analysis of the algorithm but not for its implementation. Whereas most geometric algorithms based on random sampling have been successfully derandomized [1], [3], [4], [13]–[15], this has not been the case of randomized incremental algorithms. We believe that our result is the first such instance, and we anticipate that others will follow from this work.

The term *randomized incremental* refers to a class of geometric algorithms, whose guiding principle is to add the input elements one at a time, in random order, and maintain the current solution naively after each addition—see Clarkson's survey article [7] for a detailed discussion. This is the approach followed by Clarkson and Shor to compute the convex hull of n points. Translated into dual space, we are given a set H of n half-spaces, and we must compute their intersection. At the rth stage, a subset R of r of them has been selected and a triangulation $\mathscr{T}(R^\cap)$ of the polyhedron $R^\cap = \bigcap\{h \in R\}$ has been computed. We also assume that we have, for each face of $\mathscr{T}(R^\cap)$, a *conflict list* that provides all the half-spaces cutting through it. Among the remaining $n - r$ half-spaces, we pick one at random and intersect it with the current intersection $R^\cap$, using the triangulation as a guiding device to speed up the location of the sliced simplices of $\mathscr{T}(R^\cap)$. If the intersection with $R^\cap$ is not trivial, we recompute a new triangulation and update the conflict lists.

The running time of the algorithm is proportional to the total amount of updating among the conflict lists. Intuitively, it is clear that at stage r the faces of the triangulation are cut by $O(n/r)$ half-spaces on average. By the Upper Bound Theorem, the intersection $R^\cap$ has $O(r^{\lfloor d/2 \rfloor})$ faces and its triangulation $\mathscr{T}(R^\cap)$ has on the same order of simplices. Therefore, the total number of half-space/simplex cutting pairs is $O(r^{\lfloor d/2 \rfloor}(n/r))$. It follows that if the next half-space to be added is picked at random, it should cut at most on the order of $(1/(n - r))(n/r)r^{\lfloor d/2 \rfloor}$ faces of $\mathscr{T}(R^\cap)$. If r is not too large, that puts the amount of work for slicing/

retriangulating at stage r at $O(r^{\lfloor d/2 \rfloor -1})$. Assuming that each face is cut by no more than $O(n/r)$ half-spaces, the total amount of updating at stage r is $O(nr^{\lfloor d/2 \rfloor -2})$, which represents the number of cutting pairs destroyed or created. Summing up over all r gives an expected running time of $O(n \log n + n^{\lfloor d/2 \rfloor})$, which is optimal.

Of course, this analysis is not rigorous and does not constitute a proof. However, it gives some intuition for what a fast deterministic algorithm must strive for, namely, keeping the size of each conflict list at stage r within $O(n/r)$. As it turns out, this is too much to ask, but fortunately a relaxed, "lower-moment" condition suffices. To state this condition, we need to define the *energy* of the set R: this is the sum $\sum_e |H_{|e}|^c$, taken over all segments e joining the origin to the vertices of $R^\cap$, where c is a large constant and $|H_{|e}|$ is the number of half-spaces cutting through e. The required condition says that the energy of R should be $O(r^{\lfloor d/2 \rfloor}(n/r)^c)$. Such a low-energy triangulation is called a *semicutting*.

How do we produce a "random-looking" permutation deterministically? Think of the half-spaces as a stack of cards (in arbitrary order). We proceed in $O(\log n)$ rounds: at round t, we remove 2^t cards from the deck by going through the entire deck and selecting cards to be removed. Each card has an initial *score* and we remove only cards whose scores are below average. (The removal of a single card may alter the score of all the remaining cards, so scores need to be recomputed.) Every time a card is thrown out, the corresponding half-space is added to the current structure (which means cutting through the current intersection, retriangulating, and updating the conflict lists). The order in which the cards are thrown out gives us the desired permutation. Note that with the help of the conflict lists and the current triangulation, adding a new half-space is routine. So, it appears that the efficiency of the algorithm rests entirely on the choice of a good scoring function: it must lead to a favorable ordering of the cards and be easy to compute.

At the beginning of round t, we have in hand the intersection $R^\cap$ of a subset R of $r = 2^t$ of the input half-spaces, along with its triangulation $\mathscr{T}(R^\cap)$ and the set of conflict lists. The purpose of the round is to find, among the $n - r$ remaining half-spaces, a "good" subset S of 2^t of them, and compute the new intersection $(R \cup S)^\cap$ along with its associated triangulation and conflict lists. By a good set S, we mean one that makes $\mathscr{T}((R \cup S)^\cap)$ a semicutting.

We can verify that a random choice of S is good with high probability. Thus, a reasonable approach is to simulate a random selection deterministically by using Raghavan and Spencer's *method of conditional probabilities* [18], [22]. The underlying probabilistic model is the hypergeometric distribution, which assumes that all subsets of $H \backslash R$ of size 2^t are equally likely. Let $\mathscr{E}$ be the expected energy of $R \cup S$, where R is fixed and S is random. There must be a choice of $h \in H \backslash R$ for which the expectation of the energy, conditioned upon $h \in S$, is at most $\mathscr{E}$. We choose such an h and iterate in this fashion 2^t times. This gives us a low-energy set $R \cup S$, whose triangulation $\mathscr{T}((R \cup S)^\cap)$ is therefore a semicutting.

This approach easily yields a polynomial-time algorithm for computing $H^\cap$. One major snag, however, is that computing the required conditional expectations exactly within a reasonable amount of time seems completely hopeless. Instead, we must do with very rough estimations of these quantities. We build approximation tools by using recent results [3], [13] from the theory of range spaces of finite

VC-dimension. This allows us to devise a *method of approximate conditional probabilities.*

Besides the main result of this paper, we introduce three simple ideas which might be useful in other situations:

1. A scheme for producing "random-looking" permutations.
2. A deterministic *Monte Carlo* integration method for approximating the sum (i.e., integral) of a function defined over the vertices of a hyperplane arrangement.
3. An elementary error analysis to cope with faulty calculations in the Raghavan–Spencer method.

2. Preliminaries

Computing the convex hull of n points is reducible, by duality, to computing the intersection of n half-spaces [9]. If we choose a polarity with the origin inside the simplex determined by $d + 1$ of the points, then we are guaranteed that the intersection of the half-spaces is bounded and contains the origin O. Let H be a set of n hyperplanes (assumed to be in general position and away from O), and let $H^{\cap}$ be the polyhedron formed by the intersection of the corresponding (origin-enclosing) half-spaces. (Note that we have changed our notation slightly, as H referred to a set of half-spaces in the previous section.) Given a simplex s of any dimension between 0 and d, let $H_{|s}$ be the set of hyperplanes of H that intersect the relative interior of s but do not contain s. We denote by $v(H; s)$ the number of vertices in the portion of the arrangement of H in the relative interior of s. If s is of dimension $k < d$, the k-dimensional arrangement formed by H over the affine hull of s is understood. So, in particular, if s is a line segment, $v(H; s) = |H_{|s}|$. (We use a lower bar to avoid confusion with the cardinality sign.) Finally, we use the notation $\mathscr{A}(H)$ to designate the arrangement formed by H. Because of the previous results in low dimension, we can assume that $d > 3$.

The Geode. We need to define a particular type of triangulation which we use throughout. We adopt the convention that a triangulation consists of disjoint faces which are relative interiors of simplices of all dimensions between 0 and d. (Other definitions can be found, where faces are closed sets.) Of course, the standard properties of a simplicial complex, in particular, the fact that the closures of adjacent faces intersect in the closure of a face, are assumed [4], [6], [9]. Given $R \subseteq H$, a triangulation of $\mathscr{A}(R)$ is obtained by first triangulating recursively the $(d - 1)$-dimensional cross-section of the arrangement made by each hyperplane, and then, for each cell of the arrangement, lifting all the k-faces on its boundary $(0 \leq k \leq d - 1)$ toward a consistently chosen vertex (except for the simplices partitioning the faces incident upon the vertex in question). When it comes to triangulating the polyhedron $R^{\cap}$, however, we must be more specific about the choice of lifting vertices. We define the *geode* of R, denoted $\mathscr{T}(R^{\cap})$, as the particular triangulation of $R^{\cap}$ obtained in the following recursive manner. For $k = 2, 3, \ldots, d$, in that order, triangulate each k-face f of $R^{\frown}$ like this: If $k < d$, let $v_1, \dots, v_m$ be the vertices of f and let v_1 be the one that minimizes the quantity $|H_{|e}|$, where e is the segment $Ov_i\,(1 \leq i \leq m)$. (Break ties by taking the vertex with lexicographically smallest coordinate vector.) Lift toward v_1 the triangulation of each j-face ($j < k$) of $R^{\frown}$ that lies within the boundary of f but is not incident upon v_1. By our choice of v_1 this produces a triangulation of f that is consistent with that of its boundary. (This is because v_1 is also the vertex chosen for triangulating the faces on the boundary of f that are incident upon v_1.) For $k = d$, simply lift toward O the triangulation of $\partial R^{\frown}$ just obtained. An easy inductive argument based on the Upper Bound Theorem shows that the size of the geode is $O(|R|^{\lfloor d/2 \rfloor})$. Our next result motivates this particular choice of triangulation.

Lemma 2.1. *Given the geode of $R \subseteq H$, for any constant $c \geq 1$, a constant $b = b(c)$ exists, such that*

$$\sum_s |H_{|s}|^c \leq \sum_e b|H_{|e}|^c,$$

where the first sum is taken over all faces s (or all dimensions) of the geode and the second one over all segments e connecting O to the vertices of the geode.

The proof of the lemma as well as other proofs are given in the Appendix. For any face $s \in \mathscr{T}(R^{\frown})$ the set $H_{|s}$ is called the *conflict list* of s. By setting $c = 1$ in Lemma 2.1, we easily derive the following result.

Corollary 2.2. *Given a full facial-lattice description of $R^{\frown}$ and the list $H_{|e}$, for all segments e connecting O to the vertices of the geode, we can compute the geode of R and all its conflict lists in time proportional to $|R|^{\lfloor d/2 \rfloor} + \sum_e |H_{|e}|$.*

Approximation Tools. Specializing a definition from [23], we say that a subset R of H is a *$(1/r)$-approximation* for H if, for any line segment e, the densities in R and H of the hyperplanes crossing e differ by less than $1/r$, or, formally,

$$\left| \frac{|H_{|e}|}{|H|} - \frac{|R_{|e}|}{|R|} \right| < \frac{1}{r}.$$

Matoušek [13] has shown how to compute a $(1/r)$-approximation for H of size $O(r^2 \log r)$ in time $nr^{O(1)}$, for any $r \geq 1$. A $(1/r)$-approximation can be used to estimate the number of vertices of $\mathscr{A}(H)$ inside a given simplex.

Lemma 2.3 [3]. *Let R be a $(1/r)$-approximation for a finite set H of hyperplanes in E^d. For any d-dimensional simplex s, we have*

$$\left| \frac{v(H; s)}{|H|^d} - \frac{v(R; s)}{|R|^d} \right| < \frac{1}{r}.$$

Semicuttings. We fix two integer constants c_0 and c_1, which are used throughout. We must think of them as large enough to satisfy all the inequalities in which they are involved; c_1 depends only on the dimension d, while c_0 depends on both d and c_1. Given $R \subseteq H$, if the *energy* satisfies

$$\sum_e |H_e|^{c_1} \leq c_0 |R|^{\lfloor d/2 \rfloor} \left(\frac{n}{|R|} \right)^{c_1},$$

where the sum extends over all the segments e joining O to the vertices of $R^\frown$, then the geode $\mathscr{T}(R^\frown)$ is called a *semicutting* (for H). A side observation, which we often use but do not mention explicitly again, is that we have a similar inequality for lower moments, i.e., for any $c \leq c_1$,

$$\sum_e |H_e|^{c} \leq c_0 |R|^{\lfloor d/2 \rfloor} \left(\frac{n}{|R|} \right)^{c}.$$

We omit the proof, which is a straightforward application of Hölder's inequality.

The Method of Approximate Conditional Probabilities. We briefly recall Raghavan and Spencer's method of conditional probabilities [18], [22] and generalize it to allow errors in the computations. Let X be a discrete set and let χ be a real function whose argument is any subset $R \subseteq X$ of fixed size r. The goal is to determine a particular $S \subseteq X$ of size r at which the value of the function χ does not exceed its expectation, i.e.,

$$\chi(S) \leq \mathbf{E}\chi = \binom{|X|}{r}^{-1} \sum_{R \subseteq X;\, |R| = r} \chi(R).$$

Here is a way to do it. Note that (using subscripts to indicate the relevant probability spaces)

$$\mathbf{E}_R[\chi(R)] = |X|^{-1} \sum_{x \in X} \binom{|X| - 1}{r - 1}^{-1} \sum_{R \subseteq X;\, |R| = r;\, x \in R} \chi(R) = \mathbf{E}_x[\mathbf{E}_R[\chi(R) | x \in R]].$$

Dropping the argument R for notational convenience, the element x of X, denoted s_1, that minimizes $\mathbf{E}[\chi | x \in R]$ satisfies $\mathbf{E}[\chi | s_1 \in R] \leq \mathbf{E}\chi$. Pursuing this idea, we find that the element $x \in X \backslash \{s_1\}$, denoted s_2, that minimizes $\mathbf{E}[\chi | s_1, x \in R]$ is such that

$$\mathbf{E}[\chi | s_1, s_2 \in R] \leq \mathbf{E}[\chi | s_1 \in R] \leq \mathbf{E}\chi.$$

In the end, we find a complete assignment for which

$$\chi(s_1, \ldots, s_r) = \mathbf{E}[\chi | s_1, \ldots, s_r \in R] \leq \mathbf{E}\chi.$$

Note that we can relax the selection criterion by simply requiring that the choice of s_1 satisfies $\mathbf{E}[\chi | s_1 \in R] \le \mathbf{E}\chi$, and, more generally,

$$\mathbf{E}[\chi | s_1, \ldots, s_{i+1} \in R] \le \mathbf{E}_x[\mathbf{E}[\chi | s_1, \ldots, s_i, x \in R]],$$

where the expectation $\mathbf{E}_x$ is taken over all $x \neq s_1, \ldots, s_i$, with all s_j distinct $(1 \le j \le i + 1)$. In practice, this will work only if the relevant conditional expectations can be computed efficiently. Suppose that we cannot compute $\mathbf{E}$ exactly, but, instead, we can obtain only an estimation $\mathbf{E}^*$. Let $s_1^*, \ldots, s_r^*$ be the r elements of X selected in the same manner, i.e., in such a way that

$$\mathbf{E}^*[\chi | s_1^* \in R] \le \mathbf{E}_x[\mathbf{E}^*[\chi | x \in R]]$$

and, for $i > 0$,

$$\mathbf{E}^*[\chi | s_1^*, \ldots, s_{i+1}^* \in R] \le \mathbf{E}_{x \neq s_1^*, \ldots, s_i^*}[\mathbf{E}^*[\chi | s_1^*, \ldots, s_i^*, x \in R]].$$

We also require that

$$\mathbf{E}^*[\chi | s_1^*, \ldots, s_r^* \in R] = \chi(s_1^*, \ldots, s_r^*).$$

The method will work if (i) $\mathbf{E}^*\chi$ is not too far from $\mathbf{E}\chi$ and (ii) the *drift* Δ_i is not too large, where

$$\Delta_0 = \mathbf{E}_x[\mathbf{E}^*[\chi | x \in R]] - \mathbf{E}^*\chi$$

and, for $i > 0$,

$$\Delta_i = \mathbf{E}_{x \neq s_1^*, \ldots, s_i^*}[\mathbf{E}^*[\chi | s_1^*, \ldots, s_i^*, x \in R]] - \mathbf{E}^*[\chi | s_1^*, \ldots, s_i^* \in R].$$

The first expectation is not starred because in our application it can be easily computed exactly. Note also that in the original Raghavan–Spencer method the drift is always null.

Lemma 2.4. $\chi(s_1^*, \ldots, s_r^*) \le \mathbf{E}^*\chi + \sum_{0 \le i \le r-1} \Delta_i$.

Remark. The notation $\mathbf{E}^*$ is used for mnemonic purposes. It should not be confused with a mathematical operator: $\mathbf{E}^*[\chi | x_1, \ldots, x_i]$ is simply a function of $x_1, \ldots, x_i$, with none of the usual properties of an expectation.

3. The Convex Hull Algorithm

To build intuition we begin with a probabilistic version of the algorithm and we briefly discuss its expected complexity. Then we move on to the deterministic algorithm and analyze its performance in detail.

A Probabilistic Solution. To get started we compute a $(1/c_1)$-approximation R for H and naively compute its geode as well as the conflict list $H_{|s}$ of every face s in the geode. This takes linear time. For convenience, we can always add a few hyperplanes to R (e.g., the duals of the $d+1$ points chosen at the beginning of Section 2), if necessary, so as to make the geode bounded but still enclosing $H^{\cap}$. By making c_0 large enough, we can always ensure that the geode is a semicutting. Furthermore, note that $|H_{|e}| < n/c_1$ for any segment e connecting O to a vertex of $R^{\cap}$. This concludes the start-up phase.

We are now ready to pick up the algorithm in mid-action. At the outset of round t we have already chosen a subset $R \subseteq H$ of size r whose geode $\mathscr{T}(R^{\cap})$ is a semicutting and whose conflict lits $H_{|s}$ $(s \in \mathscr{T}(R^{\cap}))$ have already been computed. To conduct the round we pick a random sample S of r hyperplanes in $H \backslash R$, and we insert them into the current intersection. Thus, by going from R to $R \cup S$ in one round, we double the number of hyperplanes inserted, which ensures that in round t on the order of 2^t hyperplanes will have been inserted.

For each $s \in \mathscr{T}(R^{\cap})$, we identify the set $S_{|s}$ of cutting hyperplanes in S, and we form the polytope $s \cap (S_{|s})^{\cap}$ naively by, say, computing $\mathscr{A}(S)$ explicitly [9] and clipping the relevant cell within s. We complete the work by computing the geode of $R \cup S$ and its conflict lists. It is easy to achieve a running time on the order of

$$r^{\lfloor d/2 \rfloor} + \sum_{s \in \mathscr{T}(R^{\cap})} |H_{|s}| \times |S_{|s}|^d.$$

By Markov's inequality, the probability that the geode of $R \cup S$ is a semicutting is at least a positive constant, provided that the expectation (for R fixed) of $\sum_v |H_{|Ov}|^{c_1}$, summing over all vertices v of $(R \cup S)^{\cap}$, is at most on the order of

$$|R \cup S|^{\lfloor d/2 \rfloor} \left(\frac{n}{|R \cup S|} \right)^{c_1},$$

which is not difficult to verify. We can also easily check that, with high probability,

$$\sum_{s \in \mathscr{T}(R^{\cap})} |S_{|s}|^{O(1)} = O(r^{\lfloor d/2 \rfloor}).$$

Intuitively, a typical face s should be cut by $O(n/r)$ hyperplanes, so at most a constant number of those should end up in S. It follows from Lemma 2.1 and Cauchy–Schwarz's inequality that the expected time for computing the geode of $R \cup S$ is $O((n/r)r^{\lfloor d/2 \rfloor})$. Since the sizes of R, $R \cup S$, etc., grow geometrically and $d > 3$, we find that the total expected running time of the algorithm is $O(n^{\lfloor d/2 \rfloor})$.

This discussion only illustrates the mechanics of the probabilistic algorithm, so we do not need to justify our claims. However, we are now ready to describe its derandomization in detail, and provide a complete analysis of its complexity.

The Deterministic Algorithm. The first round does not use randomization, so let us turn directly to the construction of S in round t. Our assumptions about R are the same as in the probabilistic case, i.e., $\mathcal{T}(R^\frown)$ is a semicutting; for convenience, however, we assume that $r = |R|$ is less than n/c_1. We can also trivially assume that r is large enough, say, $r > c_1$. For natural reasons, we want to ensure that:

1. The faces of the geode $\mathcal{T}(R^\frown)$ should not be cut by too many hyperplanes.
2. The next current set of inserted hyperplanes, $R \cup S$, should produce a semicutting.

To achieve both conditions, it would be natural to enhance the definition of the energy $\mathcal{E}$ of $R \cup S$ given earlier by setting $\mathcal{E} = \mathcal{E}_1 + \mathcal{E}_2$, where each $\mathcal{E}_i$ is the (normalized) randomized variable:

$$\mathcal{E}_1 = \frac{1}{c_0^2 r^{\lfloor d/2 \rfloor}} \sum_{s \in \mathcal{T}(R^\frown)} |S_{|s}|^{c_1}$$

and

$$\mathcal{E}_2 = \frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_e |H_{|e}|^{c_1},$$

where the sum extends over all the segments e joining O and the vertices of $(R \cup S)^\frown$. (Note that, from now on, $s \in \mathcal{T}(R^\frown)$ refers to a face of the geode that is either d-dimensional, or, else, that is not incident upon O; indeed, the nonfull dimensional faces incident upon O can be assumed not to contain any vertices of $\mathcal{A}(H)$ and thus can be ignored.) If we can show that the expectation of $\mathcal{E}$ is less than 1, then we are in a position to apply the method of conditional probabilities and derive a set S that satisfies both desired conditions.

The problem is that the expectation of $\mathcal{E}_2$ is very unwieldy to compute. So we must substitute approximations, and define a random variable $\mathcal{E}_2^*$, whose expectation $\mathbf{E}[\mathcal{E}_2^*]$ provides a fairly accurate estimation of $\mathbf{E}[\mathcal{E}_2]$. Using the terminology of the last section, we define the *estimated* expected energy of $R \cup S$ to be

$$\mathbf{E}^*[\mathcal{E}] = \mathbf{E}[\mathcal{E}_1] + \mathbf{E}[\mathcal{E}_2^*].$$

To define $\mathcal{E}_2^*$, we consider each face $s \in \mathcal{T}(R^\frown)$ in turn, and compute a $(1/\rho_s)$-approximation A_s for $H_{|s}$ of size $O(\rho_s^2 \log \rho_s)$, where

$$\rho_s = 2^{c_0}\left[\frac{r}{n}|H_{|s+}|\right]^{\sqrt{c_1}}.$$

We use the notation $s+$ to designate the relative interior of the convex hull of O and s. Note that $s+ \in \mathcal{T}(R^\frown)$ and that $|H_{|s}| \leq |H_{|s+}|$. Given a point v, let φ_v be the number of hyperplanes of R passing through v and let δ_v be a shorthand for $|H_{|o_v}|$. Lemma 2.3 suggests that a vertex $v \in \mathscr{A}(A_s)$ should "account" for roughly $(|H_{|s}|/|A_s|)^{d-\varphi_v}$ vertices of $\mathscr{A}(H)$, so it is natural that we should set

$$\mathscr{E}_2^* = \frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_{s \in \mathscr{T}(R^{\frown})} \sum_{v} \left(\frac{|H_{|s}|}{|A_s|} \right)^{d-\varphi_v} \delta_v^{c_1},$$

where the inner sum extends over all vertices v of $(R \cup S)^{\frown}$ that lie in s and are also vertices of $\mathscr{A}(R \cup A_s)$. Note that in order to account for vertices on the boundary of $R^{\frown}$, we do not require v to be a vertex of $\mathscr{A}(A_s)$ but only of $\mathscr{A}(R \cup A_s)$.

Lemma 3.1. $\mathbf{E}^*[\mathscr{E}] < \frac{1}{2}$.

Here is how the tth round is conducted. We select hyperplanes in $H \backslash R$ one at a time until we have a collection $S = S_r$ of size r. Assume that we have already chosen a subset S_k of size k, with $S_0 = \varnothing$. The *score* of a hyperplane $h \in H \backslash (R \cup S_k)$ is equal to our estimation, denoted $\mathbf{E}^*[\mathscr{E} \mid S_k \cup \{h\} \subseteq S]$, of the expectation of $\mathscr{E}$ conditioned upon having all the hyperplanes of $S_k \cup \{h\}$ in the random set S. Although ideally we would like to pick the hyperplane of lowest score, this seems too difficult to do, so we content ourselves with any one of a reasonably low score. We define the score $\zeta_k(h)$ of a hyperplane h at step k to be

$$\zeta_k(h) = \mathbf{E}^*[\mathscr{E} \mid S_k \cup \{h\} \subseteq S] \overset{\text{def}}{=} \mathbf{E}[\mathscr{E}_1 \mid S_k \cup \{h\} \subseteq S] + \mathbf{E}[\mathscr{E}_2^*(S_k \cup \{h\})].$$

What is the random variable $\mathscr{E}_2^*(S_k \cup \{h\})$? For notational convenience we define $\mathscr{E}_2^*(S_k)$, from which the definition of $\mathscr{E}_2^*(S_k \cup \{h\})$ follows trivially. At the beginning of the round, we have $\mathbf{E}^*[\mathscr{E}] = \mathbf{E}[\mathscr{E}_1] + \mathbf{E}[\mathscr{E}_2^*]$, so it is natural to set $\mathscr{E}_2^*(S_0) = \mathscr{E}_2^*$. Now, for $k > 0$, let φ_v denote the number of hyperplanes of $R \cup S_k$ passing through vertex v. (Note that since $S_0 = \varnothing$, this is consistent with our earlier definition of φ_v as the number of hyperplanes of R passing through v.) We define

$$\mathscr{E}_2^*(S_k) = \frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_{s \in \mathscr{T}(R^{\frown})} \sum_{v} \left(\frac{|H_{|s}|}{|A_s|} \right)^{d-\varphi_v} \delta_v^{c_1},$$

where the inner sum extends over all vertices v of $(R \cup S)^{\frown}$ that lie in s and are also vertices of $\mathscr{A}(R \cup S_k \cup A_s)$; we define $S = S_k \cup T$, where T is a random subset of $H \backslash (R \cup S_k)$ of size $r - k$.

Important Remark. Except for $k = 0$, $\mathbf{E}[\mathscr{E}_2^*(S_k)]$ is not the same as the conditional expectation $\mathbf{E}[\mathscr{E}_2^* \mid S_k \subseteq S]$. Note that it cannot be smaller. The reason we use $\mathbf{E}[\mathscr{E}_2^*(S_k)]$ instead of $\mathbf{E}[\mathscr{E}_2^* \mid S_k \subseteq S]$ is that to find a set S such that $\mathscr{E}_2^*$ is small is in itself quite useless: it only guarantees that a small subset of the vertices of $(R \cup S)^{\frown}$ (those in the arrangement of $R \cup A_s$) has low "energy," which does not give us a semicutting. Instead, we must introduce a *brand-new* random variable $\mathscr{E}_2^*(S_k)$ at each step, so that in the end every vertex of $(R \cup S)^\frown$ is counted in the inner sum for $\mathscr{E}_2^*(S_r)$. Indeed, we can verify that

$$\mathscr{E}_2^*(S_r) = \frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_e |H_{|e}|^{c_1},$$

where the sum extends over all segments e connecting O to the vertices of $(R \cup S_r)^\frown$. It follows that the energy of $R \cup S_r$, defined earlier as $\mathscr{E}_1 + \mathscr{E}_2$, or, to be more rigorous, as $\mathbf{E}[\mathscr{E}_1 + \mathscr{E}_2 | S = S_r]$, is also equal to $\mathbf{E}[\mathscr{E}_1 | S = S_r] + \mathscr{E}_2^*(S_r)$.

Let us show how to calculate the score $\zeta_k(h)$. Again, for notational convenience, we only discuss the calculation of $\mathbf{E}^*[\mathscr{E} | S_k \subseteq S]$. We can express $\mathbf{E}[\mathscr{E}_1 | S_k \subseteq S]$ as

$$\frac{1}{c_0^2 r^{\lfloor d/2 \rfloor}} \sum_{s \in \mathscr{T}(R^\frown)} \sum_{j \geq 0} (|S_{k|s}| + j)^{c_1} \binom{n - r - k - |H_{|s} \backslash S_k|}{r - k - j} \binom{|H_{|s} \backslash S_k|}{j} \bigg/ \binom{n - r - k}{r - k},$$

with the usual convention that

$$\binom{a}{b} = 0 \quad \text{if} \quad b < 0 \quad \text{or} \quad b > a.$$

Indeed, we pick $r - k$ hyperplanes out of $n - r - k$, and we distinguish among those (j) crossing s and the others: note that the hyperplanes crossing s form the set $H_{|s} \backslash (R \cup S_k)$, which is the same as $H_{|s} \backslash S_k$. Similarly, we write $\mathbf{E}[\mathscr{E}_2^*(S_k)]$ as

$$\frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_{s \in \mathscr{T}(R^\frown)} \sum_{v \in \mathscr{A}(R \cup S_k \cup A_s) \cap s} \chi_v \left(\frac{|H_{|s}|}{|A_s|}\right)^{d - \varphi_v} \delta_v^{c_1}$$
$$\times \binom{n - r - k - \delta_v - d + \varphi_v}{r - k - d + \varphi_v} \bigg/ \binom{n - r - k}{r - k},$$

where $\chi_v = 1$ if $H_{|Ov} \cap S_k = \varnothing$ and 0 otherwise. Here we express the fact that, in order to contribute to $\mathbf{E}[\mathscr{E}_2^*(S_k)]$, the vertex v must not be separated from O by any hyperplane of S_k nor by any selected hyperplane, and its $d - \varphi_v$ passing hyperplanes not in $R \cup S_k$ must all be selected. (Recall the redefinition of φ_v to include S_k as well as R.)

At step k we need to select a hyperplane $h_0 \in H \backslash (R \cup S_k)$ whose score does not exceed the average score, i.e.,

$$\zeta_k(h_0) \leq \frac{1}{n - r - k} \sum_{h \in H \backslash (R \cup S_k)} \zeta_k(h).$$

We show how to do this in the Appendix. We prove the following:

Lemma 3.2. *Round t can be executed in time at most proportional to*

$$\frac{n^2}{\log n}+\frac{n^2}{r^{2/3}}+\sum_{s\in\mathcal{T}(R^\cap)}|H_{|s}|^{2-\nu}(|S_{|s}|+1)^{d+2}+\left(\frac{r}{n}\right)^{c_1^{2/3}}\sum_{s\in\mathcal{T}(R^\cap)}|H_{|s}|^{2-\nu+c_1^{2/3}}$$

for some fixed $0<\nu<1$.

As in the randomized case, once $S=S_r$ has been determined, we compute the geode $\mathcal{T}((R\cup S)^\cap)$ and update the conflict lists. To do that, for each $s\in\mathcal{T}(R^\cap)$ we form the arrangement of $S_{|s}$ and the $d+1$ hyperplanes defining s [9], and we retrieve the cell $s\cap(R\cup S)^\cap$. These cells glue naturally together around adjacent simplices s of $\mathcal{T}(R^\cap)$. Therefore, it is routine to compute $(R\cup S)^\cap$. By Corollary 2.2, we can also update the conflict lists without difficulty in time within the bound of Lemma 3.2.

We are now ready to analyze the complexity of the whole algorithm. Because of our selection criterion, we can use Lemma 2.4 to bound the energy, $\mathcal{E}_1+\mathcal{E}_2^*(S)$, of $R\cup S$ by

$$\mathbf{E}^*\mathcal{E}+\sum_{0\le k\le r-1}\Delta_k,$$

where

$$\Delta_k=-\mathbf{E}^*[\mathcal{E}\,|\,S_k\subseteq S]+\frac{1}{n-r-k}\sum_{h\in H\setminus(R\cup S_k)}\mathbf{E}^*[\mathcal{E}\,|\,S_k\cup\{h\}\subseteq S].$$

Our next result bounds the drift Δ_k:

Lemma 3.3.

$$\Delta_k\le\frac{1}{c_0^2r^{1+\lfloor d/2\rfloor/c}}\left(\sum_{s\in\mathcal{T}(R^\cap)}|S_{k|s}|^{dc}\right)^{1/c},$$

where $c=c_1^{2/3}$.

From there a simple inductive argument shows that the energy of $R\cup S$ at the end of round t is less than 1, i.e., $\mathbf{E}^*[\mathcal{E}\,|\,S=S_r]<1$.

Lemma 3.4. *The energy of* $R\cup S$ *at the end of round t is less than* 1.

The lemma immediately implies that $\mathcal{T}((R\cup S)^\cap)$ is a semicutting, as desired. It also shows that

$$\sum_{s\in\mathcal{T}(R^\cap)}|S_{|s}|^{c_1}<c_0^2r^{\lfloor d/2\rfloor}.$$

Since $\mathscr{T}(R^\frown)$ is a semicutting, Cauchy–Schwarz's inequality implies that (up to constant factors)

$$\sum_{s \in \mathscr{T}(R^\frown)} |H_{|s}|^{2-\nu}(|S_{|s}| + 1)^{d+2} + \left(\frac{r}{n}\right)^{c_1^{2/3}} \sum_{s \in \mathscr{T}(R^\frown)} |H_{|s}|^{2-\nu+c_1^{2/3}} = O(1) + r^{\lfloor d/2 \rfloor}\left(\frac{n}{r}\right)^{2-\nu}.$$

Thus, Lemma 3.2 shows that the cost of round t is

$$O(n^2/\log n + n^2/r^{2/3} + n^{2-\nu} r^{\lfloor d/2 \rfloor - 2 + \nu}),$$

where $r = \Theta(2^t)$. Summing up over all t gives $O(n^{\lfloor d/2 \rfloor})$. Recall our earlier assumption that $r < n/c_1$. When we reach $r \geq n/c_1$, we simply include in the next set S the remaining hyperplanes, and finish up the work naively. The time needed for this last phase is at most proportional to

$$\sum_{s \in \mathscr{T}(R^\frown)} |S_{|s}|^d = \sum_{s \in \mathscr{T}(R^\frown)} |H_{|s}|^d = O(1) \times r^{\lfloor d/2 \rfloor}\left(\frac{n}{r}\right)^d = O(n^{\lfloor d/2 \rfloor}),$$

which completes the proof of our main result.

Theorem 3.5. *It is possible to compute the convex hull of n points in d-space deterministically in $O(n \log n + n^{\lfloor d/2 \rfloor})$ time, which is optimal.*

There are several ways of interpreting a Voronoi diagram of n points in d-space as a convex hull or an intersection of half-spaces in $(d + 1)$-space [2], [9], [10]. We have the immediate corollary:

Theorem 3.6. *The Voronoi diagram of n points in d-space can be computed in time $O(n \log n + n^{\lceil d/2 \rceil})$, which is optimal.*

Acknowledgments

I wish to thank J. Matoušek for several helpful suggestions and the referees for many useful comments.

Appendix

Lemma 2.1. *Given the geode of $R \subseteq H$, for any constant $c \geq 1$, a constant $b = b(c)$ exists, such that*

$$\sum_s |H_{|s}|^c \leq \sum_e b |H_{|e}|^c,$$

where the first sum is taken over all faces s (of all dimensions) of the geode and the second one over all segments e connecting O to the vertices of the geode.

Proof. Because it cannot pass through O, any hyperplane of H meeting a k-face s of the geode without containing it must intersect (without containing) the relative interior of at least one of the $k+1$ segments connecting its vertices to O. Thus, it suffices to prove the inequality obtained by substituting W_s for $|H_{|s}|$, where W_s is the number of hyperplanes crossing at least one of those segments outside of its endpoints. We show by induction on k that, for any k-face f of $R^\frown$, the sum $\sum_s W_s^c$, denoted A_f, where s ranges over the faces of the geode lying within the closure of f, is at most

$$d^k(2^c+1)^k \sum_{e \in E} |H_{|e}|^c,$$

where E is the set of edges joining O to the vertices of f. The case $k=d$, where f is the interior of $R^\frown$, gives the lemma.

The case $k=0$ is obvious, so let assume that $k>0$. We observe that, by our choice of the lifting vertex, we have $A_f \le (2^c+1)\sum_g A_g$, where g ranges over all the $(k-1)$-faces of $R^\frown$ incident upon f. (Note that g ranges over faces of $R^\frown$, which are not necessarily faces of the geode.) The term 1 comes from the contribution of the faces incident upon f, while the term 2^c accounts (conservatively) for the effect of the lifting vertex to the contribution of the geode faces within f. By induction, we have

$$\sum_g A_g \le \mu d^{k-1}(2^c+1)^{k-1} \sum_{e \in E} |H_{|e}|^c,$$

where μ represents the maximum multiplicity of a segment e in the counting. However, the general position of H, and hence of R, ensures that a segment e is adjacent to d hyperplanes, and, therefore, k $(k-1)$-faces incident upon f. It follows that $\mu \le d$, which completes the inductive proof. □

Lemma 2.4. $\chi(s_1^*, \dots, s_r^*) \le \mathbf{E}^*\chi + \sum_{0 \le i \le r-1} \Delta_i$.

Proof. For $i>0$, we have

$$\begin{aligned} \mathbf{E}^*[\chi | s_1^*, \dots, s_{i+1}^* \in R] &\le \mathbf{E}_{x \ne s_1^*, \dots, s_i^*}[\mathbf{E}^*[\chi | s_1^*, \dots, s_i^*, x \in R]] \\ &\le \mathbf{E}^*[\chi | s_1^*, \dots, s_i^* \in R] + \Delta_i, \end{aligned}$$

with a similar inequality for $i=0$. Since $\chi(s_1^*, \dots, s_r^*) = \mathbf{E}^*[\chi | s_1^*, \dots, s_r^* \in R]$, the lemma follows by induction. □

Lemma 3.1. $\mathbf{E}^*[\mathscr{E}] < \frac{1}{2}$.

Proof. Given $s \in \mathscr{T}(R^\frown)$, let $m = |H_{|s}|$ and $k_0 = 2c_1\lceil mr/n \rceil$. To evaluate the expectation of $|S_{|s}|^{c_1}$ we put a threshold at $k=k_0$, and, for $k>k_0$, we consider the event where, out of the r random hyperplanes, exactly k of them intersect s. We have

$$\mathbf{E}|S_{|s}|^{c_1} \le k_0^{c_1} + \sum_{k_0 < k \le r} k^{c_1} \binom{n-r-m}{r-k} \binom{m}{k} \Big/ \binom{n-r}{r}.$$

Note that, because of the start-up phase, m can be at most a small fraction of n. The same is true of r, so all the binomial coefficients are strictly positive, except possibly for $\binom{m}{k}$, for which we adopt the convention that it is zero if $m < k$.

$$\begin{aligned}\mathbf{E}|S_{|s}|^{c_1} &\le k_0^{c_1} + \sum_{k_0<k\le r} k^{c_1}\left(\frac{r}{n-2r+k}\right)^k\left(1-\frac{m}{n-r}\right)^{r-k}\left(\frac{em}{k}\right)^k\\ &\le k_0^{c_1} + \sum_{k>k_0} k^{c_1}\left(\frac{c_1 mr}{kn}\right)^k\\ &\le k_0^{c_1} + \sum_{k>k_0} k^{c_1}2^{-k}\\ &= O(1)\times\left(\frac{r|H_{|s}|}{n}\right)^{c_1} + O(1).\end{aligned}$$

Note that the constants hidden behind $O(1)$ do not depend on c_0. Therefore, by Lemma 2.1 and the fact that $\mathscr{T}(R^\frown)$ is a semicutting,

$$\begin{aligned}\mathbf{E}[\mathscr{E}_1] &= O\left(\frac{1}{c_0^2}\right) + O(1)\times\frac{1}{c_0^2 r^{\lfloor d/2\rfloor}}\left(\frac{r}{n}\right)^{c_1}\sum_{s\in\mathscr{T}(R^\frown)}|H_{|s}|^{c_1}\\ &= O\left(\frac{1}{c_0^2}\right) + O(1)\times\frac{1}{c_0^2 r^{\lfloor d/2\rfloor}}\left(\frac{r}{n}\right)^{c_1} c_0 r^{\lfloor d/2\rfloor}\left(\frac{n}{r}\right)^{c_1}.\end{aligned}$$

It follows that

$$\mathbf{E}[\mathscr{E}_1] < \tfrac{1}{4}. \tag{A.1}$$

To bound $\mathbf{E}[\mathscr{E}_2^*]$ is more difficult and we must begin with a technical extension of a result of Clarkson and Shor [8] concerning k-levels in arrangements of hyperplanes. We say that a point p is at *level* k if the relative interior of the segment Op intersects exactly k hyperplanes of H. Let $f_k(H)$ denote the number of vertices of $\mathscr{A}(H)$ at level k. It was shown in [8] that the prefix sum

$$f_{\le k}(H) = \sum_{0\le i\le k} f_i(H)$$

is in $O(n^{\lfloor d/2\rfloor}k^{\lceil d/2\rceil})$. For any $j \le d$, let $f_k(H, R, j)$ denote the number of vertices of $\mathscr{A}(H)$ at level k formed by the intersection of any j-face of $R^\frown$ and j hyperplanes of $H\backslash R$. (Note that the level function remains unchanged.) We prove that

$$f_{\le k}(H, R, j) = O(1)\times(k+1)^j\left(r+\frac{n}{k+1}\right)^{\lfloor d/2\rfloor}. \tag{A.2}$$

Let $t = \lfloor n/(2k+2) \rfloor$; we can obviously assume that t is large enough. Let T be a random sample of $H \backslash R$ obtained by picking each hyperplane independently with probability t/n, and let V be the number of vertices of the polyhedron $R^{\cap} \cap T^{\cap}$ that are formed by the intersection of a j-face of $R^{\cap}$ with j hyperplanes of $H \backslash R$. With high probability, T has on the order of t hyperplanes, so we should expect the polyhedron $R^{\cap} \cap T^{\cap}$ to be bounded by $O(r+t)$ hyperplanes, which would then give an upper bound of $O((r+t)^{\lfloor d/2 \rfloor})$ on its number of vertices, and, hence, on V. Indeed, using elementary tail estimates for the binomial distribution, we leave it as an exercise to show that the expected value of V is $O((r+t)^{\lfloor d/2 \rfloor})$. We can also estimate $\mathbf{E}V$ from below: because $t/n \leq \frac{1}{2}$,

$$\mathbf{E}V = \sum_{i \geq 0} f_i(H, R, j)\left(\frac{t}{n}\right)^j\left(1 - \frac{t}{n}\right)^i \geq e^{-2kt/n}\left(\frac{t}{n}\right)^j f_{\leq k}(H, R, j) \geq e^{-1}\left(\frac{t}{n}\right)^j f_{\leq k}(H, R, j),$$

from which (A.2) follows.

Using the fact that $\delta_v < n/c_1$ (because of the start-up phase) and $r < n/c_1$, we easily derive an upper bound on the expectation of $\mathscr{E}_2^*$. For a given face s of $\mathscr{T}(R^{\cap})$ we consider every vertex v of $\mathscr{A}(R \cup A_s)$ inside s, and we express the probability that v is a vertex of $(R \cup S)^{\cap}$ for random S: this means that the $d - \varphi_v$ hyperplanes through v that are not already in R are picked and that the δ_v separating hyperplanes are not chosen. We have

$$\begin{aligned}\mathbf{E}[\mathscr{E}_2^*] &= \frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_{s \in \mathscr{T}(R^{\cap})} \sum_{v \in s \cap \mathscr{A}(R \cup A_s)} \left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v} \delta_v^{c_1} \\ &\qquad \times \binom{n - r - \delta_v - d + \varphi_v}{r - d + \varphi_v} \Big/ \binom{n-r}{r} \\ &\leq \frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_{s \in \mathscr{T}(R^{\cap})} \sum_{v \in s \cap \mathscr{A}(R \cup A_s)} \left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v} \delta_v^{c_1} \\ &\qquad \times \left(\frac{r}{n - r - \delta_v}\right)^{d-\varphi_v}\left(1 - \frac{\delta_v}{n-r}\right)^r.\end{aligned}$$

We fix j $(0 \leq j \leq d)$ and define

$$F_j = \sum_s \sum_v \delta_v^{c_1}\left(\frac{r}{n - r - \delta_v} \times \frac{|H_{|s}|}{|A_s|}\right)^{d-j}\left(1 - \frac{\delta_v}{n-r}\right)^r,$$

where the summation in s extends over all $(d-j)$-faces of $\mathscr{T}(R^{\cap})$, and the one in v ranges over all the vertices in s in the $(d-j)$-dimensional arrangement formed by A_s over the affine hull of s. Because of general position we can assume that the $(d-j)$-dimensional faces of $\mathscr{T}(R^{\cap})$ lying in fewer than j hyperplanes of R do not contain any vertices v and thus can be ignored. This implies that all the vertices v involved in the sum satisfy $\varphi_v = j$. Because of previous restrictions on δ_v and r, we have

$$F_j = O(1) \times \sum_s \left(\frac{r|H_{|s}|}{n|A_s|}\right)^{d-j} \sum_v \delta_v^{c_1}\left(1 - \frac{\delta_v}{n}\right)^r,$$

where the factor $O(1)$ does not depend on c_0.

We wish to estimate the inner sum in the expression of F_j as a sum over vertices of the $(d-j)$-dimensional arrangement induced by H over s. To do so, we use an approach similar to the finite-element method in numerical analysis. We subdivide the summation domain into clusters over which simplifying assumptions can be made about the summands. This can be regarded as a deterministic *Monte Carlo* integration method.

Let $\mathscr{T}_s$ be a triangulation within s of a $(1/v_s)$-approximation for $H_{|s}$ of size $O(v_s^2 \log v_s)$, where $v_s = \lceil r|H_{|s+}|/n \rceil$. (The weaker notion of an ε-net is sufficient here, but for the reader's sake we probably do not need to add any new definitions at this point. Note that $\mathscr{T}_s$ need not be computed: it is only used to prove the lemma.) The triangulation partitions s into faces (i.e., relatively open simplices) of all dimensions between 0 and $d-j$. Let us perturb $\mathscr{T}_s$ slightly so that all the vertices v (in the definition of F_j) lie only in $(d-j)$-dimensional faces of $\mathscr{T}_s$. Note that, as a result of the perturbation, given any two points p, q inside the same $(d-j)$-dimensional face of $\mathscr{T}_s$, we have (recall that $|H_{|s}| \le |H_{|s+}|$)

$$|\delta_p - \delta_q| < \frac{|H_{|s}|}{v_s} + O(1) \le \frac{2n}{r}.$$

In this way we can write

$$\sum_v = \sum_\sigma \sum_{v \in \sigma \cap \mathscr{A}(R \cup A_s)},$$

where σ extends over all the $(d-j)$-dimensional faces of $\mathscr{T}_s$. By Lemma 2.3 (applied in dimension $d-j$),

$$\sum_{v \in \sigma \cap \mathscr{A}(R \cap A_s)} 1 < \frac{|A_s|^{d-j}}{\rho_s} + \left(\frac{|A_s|}{|H_{|s}|}\right)^{d-j} \sum_{v \in \sigma \cap \mathscr{A}(H)} 1.$$

Note that ρ_s can always be assumed to be nonzero since any simplex s such that $H_{|s+} = \varnothing$ can be ignored from the summation for F_j. Returning to F_j, we can bound the inner sum as follows:

$$\begin{aligned}\sum_v \delta_v^{c_1}\left(1 - \frac{\delta_v}{n}\right)^r &= \sum_\sigma \sum_{v \in \sigma} \delta_v^{c_1}\left(1 - \frac{\delta_v}{n}\right)^r \\ &\le \sum_\sigma \sum_{v \in \sigma} \left(\delta_w + \frac{2n}{r}\right)^{c_1}\left(1 - \frac{\max\{0, \delta_w - 2n/r\}}{n}\right)^r,\end{aligned}$$

where, for each v, we are allowed to choose *any* point w in σ. It immediately follows that $F_j = O(1) \times (F'_j + F''_j)$, where the factor $O(1)$ does not depend on c_0, and

$$F'_j = \left(\frac{r}{n}\right)^{d-j} \sum_s \sum_{v \in s \cap \mathscr{A}(H)} \left(\delta_v + \frac{4n}{r}\right)^{c_1} \left(1 - \frac{\max\{0, \delta_v - 4n/r\}}{n}\right)^r$$

and

$$F''_j = \left(\frac{r}{n}\right)^{d-j} \sum_s \sum_\sigma \frac{|H_{|s}|^{d-j}}{\rho_s} \left(\delta_w + \frac{2n}{r}\right)^{c_1},$$

where w is any vertex in σ.

$$\begin{aligned} F'_j &\leq \left(\frac{r}{n}\right)^{d-j} \sum_{v \in \mathscr{A}(H);\, v \in R^\frown;\, \varphi_v = j} \left(\delta_v + \frac{4n}{r}\right)^{c_1} e^{-\max\{0, \delta_v - 4n/r\} r/n} \\ &= O(1) \times \left(\frac{r}{n}\right)^{d-j} \sum_{k \geq 0} \left(k + \frac{4n}{r}\right)^{c_1} f_k(H, R, d-j) e^{-rk/n}. \end{aligned}$$

(Note that $v \in R^\frown$ does *not* mean that v is a vertex of $R^\frown$.) Let $k_0 = \lceil 4n/r \rceil$; the contribution to the sum of the first k_0 terms is easily seen to be $O(1) \times r^{\lfloor d/2 \rfloor}(n/r)^{c_1}$ because of (A.2). To handle the second part, we use Lemma 2.1 but now we apply summation by parts beforehand. In other words, using the identity

$$\sum_{k=k_0}^{N} u_k v_k = \sum_{k=k_0}^{N} (u_k - u_{k+1}) \sum_{i=k_0}^{k} v_i + u_{N+1} \sum_{i=k_0}^{N} v_i,$$

with $u_k = k^{c_1} e^{-rk/n}$ and $v_k = f_k(H, R, d-j)$, we derive a similar upper bound, from which it follows that

$$F'_j = O(1) \times r^{\lfloor d/2 \rfloor} \left(\frac{n}{r}\right)^{c_1}. \tag{A.3}$$

Here are the details: Let

$$A = \sum_{k \geq k_0} k^{c_1} f_k(H, R, d-j) e^{-rk/n}.$$

We have

$$A \leq \sum_{k \geq k_0} (k^{c_1} e^{-rk/n} - (k+1)^{c_1} e^{-r(k+1)/n})(f_{\leq k}(H, R, d-j) - f_{\leq k_0 - 1}(H, R, d-j)),$$

since the term $u_{N+1} \sum_{k_0}^{N} v_i$ tends to 0 as N goes to infinity.

Note that

$$k^{c_1}e^{-rk/n} - (k+1)^{c_1}e^{-r(k+1)/n} \le k^{c_1}e^{-rk/n}(1 - e^{-r/n}) \le \frac{r}{n}\, k^{c_1}e^{-rk/n},$$

therefore each of the following inequalities hold up to within a constant factor:

$$\begin{aligned} A &\le \left(\frac{r}{n}\right) \sum_{k \ge k_0} k^{c_1}e^{-rk/n} f_{\le k}(H, R, d-j) \\ &\le \left(\frac{r}{n}\right) r^{\lfloor d/2 \rfloor} \sum_{k \ge k_0} k^{c_1+d-j}e^{-rk/n} \\ &\le \left(\frac{r}{n}\right) r^{\lfloor d/2 \rfloor} \int_{k_0}^{\infty} x^{c_1+d-j}e^{-rx/n}\,dx \\ &\le \left(\frac{n}{r}\right)^{c_1+d-j} r^{\lfloor d/2 \rfloor} \int_{rk_0/n}^{\infty} y^{c_1+d-j}e^{-y}\,dy \\ &= O(1) \times \left(\frac{n}{r}\right)^{c_1+d-j} r^{\lfloor d/2 \rfloor}, \end{aligned}$$

from which (4.3) follows. For any w in s, we have $\delta_w \le |H_{|s+}|$, where, as we recall, $s+$ is the relative interior of the convex hull of O and s. Note that $|H_{|s}| \le |H_{|s+}|$.

$$F_j'' = O(1) \times \left(\frac{r}{n}\right)^{d-j} \sum_s \frac{|H_{|s+}|^{d-j}}{\rho_s} (1 + v_s^2 \log v_s)^{d-j} \left(|H_{|s+}| + \frac{2n}{r}\right)^{c_1}.$$

We distinguish between several cases. If $|H_{|s+}| \le n/r$, then $\rho_s = 2^{c_0}$ and $v_s = 1$, and

$$F_j'' = O(1) \times \left(\frac{r}{n}\right)^{d-j} \sum_s 2^{-c_0} \left(\frac{n}{r}\right)^{c_1+d-j} \le r^{\lfloor d/2 \rfloor} \left(\frac{n}{r}\right)^{c_1}$$

for c_0 large enough. Of course, the sum is understood to range only over those s for which the condition $|H_{|s+}| \le n/r$ holds.

If $|H_{|s+}| > n/r$, then we have (summing over all relevant $s \in \mathcal{T}(R^{\frown})$)

$$F_j'' = O(1) \times \left(\frac{r}{n}\right)^{d-j} \sum_{s \in \mathcal{T}(R^{\frown})} 2^{-c_0} \left(\frac{r|H_{|s+}|}{n}\right)^{3d-\sqrt{c_1}} |H_{|s+}|^{c_1+d-j}.$$

The simplex $s+$ belongs to $\mathcal{T}(R^{\frown})$. Note that it can be counted at most twice if we extend the summation to all s in $\mathcal{T}(R^{\frown})$ (of all dimensions), therefore

$$F_j'' = O(1) \times \left(\frac{r}{n}\right)^{4d-j-\sqrt{c_1}} \sum_{s \in \mathcal{T}(R^{\frown})} 2^{-c_0} |H_{|s}|^{c_1+4d-j-\sqrt{c_1}}$$

By Cauchy–Schwarz, it follows that

$$F_j'' \leq r^{\lfloor d/2 \rfloor} \left(\frac{n}{r}\right)^{c_1}.$$

Since the $O(1)$ factor in (A.3) does not depend on c_0, we derive from (A.1), (A.3) that

$$\mathbf{E}^*[\mathscr{E}] = \mathbf{E}[\mathscr{E}_1] + \mathbf{E}[\mathscr{E}_2^*] < \frac{1}{4} + \frac{O(1)}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_{0 \leq j \leq d} (F_j' + F_j'') < \frac{1}{4} + \frac{O(1)}{c_0} < \frac{1}{2}.$$

□

Lemma 3.2. *Round t can be executed in time at most proportional to*

$$\frac{n^2}{\log n} + \frac{n^2}{r^{2/3}} + \sum_{s \in \mathscr{T}(R^\frown)} |H_{|s}|^{2-\nu}(|S_{|s}| + 1)^{d+2} + \left(\frac{r}{n}\right)^{c_1^{2/3}} \sum_{s \in \mathscr{T}(R^\frown)} |H_{|s}|^{2-\nu+c_1^{2/3}}$$

for some fixed $0 < \nu < 1$.

Proof. Our first step must be to compute the approximations A_s for each simplex $s \in \mathscr{T}(R^\frown)$. Conservatively, this takes time proportional to

$$\sum_{s \in \mathscr{T}(R^\frown)} |H_{|s+}| + \left(\frac{r}{n}\right)^{c_1^{2/3}} \sum_{s \in \mathscr{T}(R^\frown)} |H_{|s+}|^{1+c_1^{2/3}}.$$

Recall that at step k we must find a hyperplane h_0 whose score $\zeta_k(h_0)$ does not exceed the average score

$$\frac{1}{n-r-k} \sum_{h \in H \setminus (R \cup S_k)} \zeta_k(h).$$

To speed up the process, we set up a rudimentary priority queue structure in the form of a tree of height 2, whose leaves are in bijection with the hyperplanes of $H \setminus (R \cup S_k)$, and whose internal nodes have degree at most $\sqrt{n}$. Our first attempt might be to store at each node z the average score of the hyperplanes in the set $L(z)$ associated with the set of leaves below z, i.e.,

$$\frac{1}{|L(z)|} \sum_{h \in L(z)} \zeta_k(h).$$

Indeed, this would allow us to home in fairly quickly toward a good h_0. Updating the tree would be too costly, however, so instead we store only partial scoring information at the nodes. Recall that $\mathbf{E}[\mathscr{E}_1 | S_k \subseteq S]$ is equal to

$$\frac{1}{c_0^2 r^{\lfloor d/2 \rfloor}} \sum_{s \in \mathscr{T}(R^\frown)} \sum_{j \geq 0} (|S_{k|s}| + j)^{c_1} \binom{n - r - k - |(H_{|s} \backslash S_k|}{r - k - j} \binom{|H_{|s} \backslash S_k|}{j} \Big/ \binom{n - r - k}{r - k},$$

and $\mathbf{E}[\mathscr{E}_2^*(S_k)]$ is equal to

$$\frac{1}{c_0 r^{\lfloor d/2 \rfloor} (n/r)^{c_1}} \sum_{s \in \mathscr{T}(R^\frown)} \sum_{v \in \mathscr{A}(R \cup S_k \cup A_s) \cap s} \chi_v \left(\frac{|H_{|s}|}{|A_s|} \right)^{d - \varphi_v} \delta_v^{c_1}$$
$$\times \binom{n - r - k - \delta_v - d + \varphi_v}{r - k - d + \varphi_v} \Big/ \binom{n - r - k}{r - k},$$

where $\chi_v = 1$ if $H_{|Ov} \cap S_k = \varnothing$ and 0 otherwise. After regrouping, $\mathbf{E}[\mathscr{E}_1 | S_k \subseteq S]$ can be expressed as

$$\sum_{i, \alpha} N_1(S_k, i, \alpha) F(k, i, \alpha),$$

where

$$F(k, i, \alpha) = \sum_j j^\alpha \binom{n - r - k - i}{r - k - j} \binom{i}{j} \Big/ \binom{n - r - k}{r - k}$$

and

$$N_1(S_k, i, \alpha) = \frac{1}{c_0^2 r^{\lfloor d/2 \rfloor}} \sum_s \binom{c_1}{\alpha} |S_{k|s}|^{c_1 - \alpha},$$

where the sum ranges over all $s \in \mathscr{T}(R^\frown)$ such that $|H_{|s} \backslash S_k| = i$. We must adopt the convention that $|S_{k|s}|^0 = 1$, even if $|S_{k|s}| = 0$. Note that $0 \leq i \leq \max_s |H_{|s} \backslash S_k|$ and $0 \leq \alpha \leq c_1$. Similarly, we can write

$$\mathbf{E}[\mathscr{E}_2^*(S_k)] = \sum_{i, j} N_2(S_k, i, j) G(k, i, j),$$

where

$$G(k, i, j) = \binom{n - r - k - i}{r - k - j} \Big/ \binom{n - r - k}{r - k},$$

with $0 \leq i \leq \max_s |H_{|s} \backslash S_k|$ and $0 \leq j \leq d$, and

$$N_2(S_k, i, j) = \frac{1}{c_0 r^{\lfloor d/2 \rfloor} (n/r)^{c_1}} \sum_{s \in \mathscr{T}(R^\frown)} \sum_v \left(\frac{|H_{|s}|}{|A_s|} \right)^j \delta_v^{c_1},$$

where the inner sum ranges over all $v \in \mathscr{A}(R \cup S_k \cup A_s) \cap s$ such that (i) $H_{|O_v} \cap S_k = \emptyset$, (ii) $d - \varphi_v = j$, and (iii) $\delta_v = i - j$. We do this to separate quantities (i.e., F, G) that are purely combinatorial and depend on k from quantities that are essentially geometric.

We now have

$$\zeta_k(h) = \sum_{i,\alpha} N_1(S_k \cup \{h\}, i, \alpha)F(k+1, i, \alpha) + \sum_{i,j} N_2(S_k \cup \{h\}, i, j)G(k+1, i, j).$$

We can now write $\zeta_k(h)$ as the sum $\zeta_k + \Delta_k(h)$ of two terms, the first of which does not depend on h. We have

$$\zeta_k = \sum_{i,\alpha} N_1(S_k, i, \alpha)F(k+1, i, \alpha) + \sum_{i,j} N_2(S_k, i, j)G(k+1, i, j),$$

or in vector form

$$\zeta_k = \tilde{N}_1 \cdot \tilde{F} + \tilde{N}_2 \cdot \tilde{G}.$$

The vectors $\tilde{N}_1$, $\tilde{F}$ are indexed by (i, α), or more simply by a single index running between 1 and $(c_1 + 1)(\max_s |H_{|s}| + 1)$. In this way the two vectors have a fixed length during the entire round. Similarly, the vectors $\tilde{N}_2$, $\tilde{G}$ can be made to have a fixed length equal to $(d + 1)(\max_s |H_{|s}| + 1)$. We also have

$$\Delta_k(h) = \sum_{i,\alpha} (N_1(S_k \cup \{h\}, i, \alpha) - N_1(S_k, i, \alpha))F(k+1, i, \alpha)$$
$$+ \sum_{i,j} (N_2(S_k \cup \{h\}, i, j) - N_2(S_k, i, j))G(k+1, i, j),$$

or in vector form

$$\Delta_k(h) = \Delta\tilde{N}_1(h) \cdot \tilde{F} + \Delta\tilde{N}_2(h) \cdot \tilde{G}.$$

We store the two vectors $\tilde{N}_1$ and $\tilde{N}_2$ separately. Each node z of the tree stores the two vectors $(i = 1, 2)$

$$\tilde{M}_i(z) = \frac{1}{|L(z)|} \sum_{h \in L(z)} \Delta\tilde{N}_i(h),$$

where we recall that $L(z)$ is the set of hyperplanes associated with the set of leaves below z.

To find h_0 we evaluate ζ_k and go down the tree until we reach a leaf. At internal node z, we compute the average

$$\frac{1}{|L(z_i)|} \sum_{h \in L(z_i)} \zeta_k(h) = \zeta_k + \frac{1}{|L(z_i)|} \sum_{h \in L(z_i)} (\Delta\tilde{N}_1(h) \cdot \tilde{F} + \Delta\tilde{N}_2(h) \cdot \tilde{G})$$
$$= \zeta_k + \tilde{M}_1(z_i) \cdot \tilde{F} + \tilde{M}_2(z_i) \cdot \tilde{G}$$

for each child z_i and branch toward the child with the smallest average. This clearly takes us to a leaf whose score is at most the average. Note that the computation at each z_i requires only two inner products since we have stored the relevant vectors at the nodes (assuming that all vectors $\tilde{F}$, $\tilde{G}$ are also available). Once h_0 has been found, we delete the corresponding leaf and update all the vectors $\tilde{N}_i$, $\Delta\tilde{N}_i(h)$ that need updating as well as the relevant nodes of the tree.

Let us review each step of the process in detail. First, how hard is it to precompute all the vectors $\tilde{F}$, $\tilde{G}$? Since $\mathscr{T}(R^\frown)$ is a semicutting,

$$\sum_{s \in \mathscr{T}(R^\frown)} |H_{|s}|^{c_1} = O(1) \times r^{\lfloor d/2 \rfloor}\left(\frac{n}{r}\right)^{c_1},$$

and therefore $\max_s |H_{|s}| = O(n/r^{1-\varepsilon})$ for an arbitrarily small constant $\varepsilon = \varepsilon(c_1) > 0$. So, all the necessary binomial coefficients can be precomputed, as well as the vectors $\tilde{F}$ needed in round t, in time proportional to

$$r\left(\frac{n}{r^{1-\varepsilon}}\right) \times \min\left\{r, \frac{n}{r^{1-\varepsilon}}\right\},$$

which is (conservatively) $O(n^2/\log n)$, for, say, $\varepsilon = \frac{1}{3}$. All the vectors $\tilde{G}$ are easily obtained in time $O(nr^\varepsilon) = O(n^{1+\varepsilon})$. Note that computing binomial coefficients exactly might be a problem if $O(\log n)$-bit precision words are used. However, we can compute them with a relative error of $1/n^c$, for an arbitrarily large constant $c > 0$, which, we can easily verify, is good enough for our purposes.

At the beginning of round t, the vector $\tilde{N}_1$ can be obtained trivially by exploring each conflict list, which takes time

$$\sum_{s \in \mathscr{T}(R^\frown)} |H_{|s}|.$$

(To simplify the notation all our analysis is done up to within constant factors.) For each $s \in \mathscr{T}(R^\frown)$ we set up a data structure for computing δ_v efficiently. This consists of a triangulation of the arrangement formed by a $(1/|H_{|s+}|^c)$-approximation for $H_{|s+}$, for some small positive constant c. For each vertex w of the triangulation we compute δ_w by examining the entire set $H_{|s+}$. Then any δ_v $(v \in s+)$ can be obtained very simply by locating the face σ enclosing v (in the computed triangulation) and examining each hyperplane cutting it. Knowing the value δ_w of, say, the closest vertex w of σ to O, it is then straightforward to derive δ_v. The preprocessing time is $O(|H_{|s+}|^{2-\nu})$ and each query can be answered in time $O(|H_{|s+}|^{1-\nu})$ for some fixed $0 < \nu < 1$. It follows that, counting preprocessing, $\tilde{N}_2$ can be found in time

$$\sum_{s \in \mathscr{T}(R^\frown)} |H_{|s+}|^{2-\nu} + \sum_{s \in \mathscr{T}(R^\frown)} |H_{|s+}|^{1-\nu} \times |A_s|^d.$$

At the beginning of round t, to compute $\Delta\tilde{N}_i(h)$ $(i = 1, 2)$ requires only the examination of the simplices $s \in \mathscr{T}(R^{\cap})$ cut by h or separated from O by h, and thus takes time

$$nr^{\varepsilon - 1} + \sum_{s: s^+ \cap h \neq \varnothing} |H_{|s+}|^{1-\nu} \times |A_s|^d.$$

The term $nr^{\varepsilon - 1}$ comes from the fact that the entire vector, represented as an array, needs to be initialized and its size is $\max_s |H_{|s}| = O(nr^{\varepsilon - 1})$. We omit the details of the data structure needed to find all the relevant simplices s because they are straightforward. So, to set up the tree in its entirety takes time

$$n^2 r^{\varepsilon - 1} + \sum_{s \in \mathscr{T}(R^{\cap})} |H_{|s+}|^{2-\nu} \times |A_s|^d.$$

Once all the relevant vectors are available, h_0 is obtained by computing $O(n^{1/2})$ inner products, which takes $O(n^{3/2} r^{\varepsilon - 1})$ time.

Once h_0 has been found, we delete its corresponding leaf from the tree. To update $\tilde{N}_i$ $(i = 1, 2)$ requires only work within the simplices s cut or separated from O by h_0. Let us consider the update in the midst of the round. For $\tilde{N}_2$, a straightforward solution involves examining each vertex formed by $R \cup S_k \cup A_s$ within s and checking whether it is separated from O by $S_k \cup \{h_0\}$. The distance δ_v for each such vertex v is computed using the data structure. Note that φ_v might change. An obvious upper bound on the time for updating both $\tilde{N}_1$ and $\tilde{N}_2$ is

$$nr^{\varepsilon - 1} + \sum_{s: s^+ \cap h_0 \neq \varnothing} |H_{|s+}|^{1-\nu} \times |S_{|s} \cup A_s|^{d+1}.$$

To update $\Delta\tilde{N}_i(h)$ $(i = 1, 2)$ we must examine each $s \in \mathscr{T}(R^{\cap})$ that is cut or separated from O by both h and h_0, and modify the vector $\Delta\tilde{N}_i(h)$ accordingly. Care must be taken not to look at the whole vector unless this is needed. In other words, we update only those entires that are no longer valid, without looking at those that are not affected by the insertion of h_0. Thus, the time required is at most

$$\sum_s |H_{|s+}|^{1-\nu} \times |S_{|s} \cup A_s|^{d+1},$$

where the sum ranges over all $s \in \mathscr{T}(R^{\cap})$ that are cut or separated from O by both h and h_0. Summing over all h gives us an upper bound on the time required to update all $\Delta\tilde{N}_i(h)$ as well as the information stored at the internal nodes of the tree. (Again, we do not look at the entire vectors stored at the nodes, but only at the entries to be updated.) The upper bound is

$$\sum_{s: s^+ \cap h_0 \neq \varnothing} |H_{|s+}|^{2-\nu} \times |S_{|s} \cup A_s|^{d+1}.$$

Finally, taken over the entire round, the updating time amounts to

$$nr^{\varepsilon} + \sum_{s \in \mathscr{T}(R^{\frown})} |S_{|s+}| \times |H_{|s+}|^{2-\nu} \times |S_{|s} \cup A_s|^{d+1}.$$

We conclude that round t can be executed in time

$$\frac{n^2}{\log n} + n^{3/2}r^{\varepsilon} + n^2 r^{\varepsilon - 1} + nr^{\varepsilon} + \sum_{s \in \mathscr{T}(R^{\frown})} |H_{|s+}|^{2-\nu} \times |S_{|s+} \cup A_s|^{d+2}.$$

Up to within a constant factor we have

$$\begin{aligned} &\sum_{s \in \mathscr{T}(R^{\frown})} |H_{|s+}|^{2-\nu} \times |S_{|s+} \cup A_s|^{d+2} \\ &\qquad \le \sum_{s} |H_{|s+}|^{2-\nu} \times |S_{|s+}|^{d+2} + \sum_{s} |H_{|s+}|^{2-\nu} \times |A_s|^{d+2}. \end{aligned}$$

From the relation

$$|A_s| \le \left(\frac{r|H_{|s+}|}{n}\right)^{3\sqrt{c_1}} + 1,$$

again understood up to within a constant factor, we derive

$$\begin{aligned} &\sum_{s \in \mathscr{T}(R^{\frown})} |H_{|s+}|^{2-\nu} \times |S_{|s+} \cup A_s|^{d+2} \\ &\qquad \le \sum_{s} |H_{|s+}|^{2-\nu} \times |S_{|s+}|^{d+2} \\ &\qquad\qquad + \left(\frac{r}{n}\right)^{3(d+2)\sqrt{c_1}} \sum_{s} |H_{|s+}|^{2-\nu+3(d+2)\sqrt{c_1}} + \sum_{s} |H_{|s+}|^{2-\nu}, \end{aligned}$$

since $|H_{|s}| \le |H_{|s+}|$. All subscripts $s+$ can be replaced by s without losing more than a constant factor. Setting $\varepsilon = \frac{1}{3}$ and taking into account the cost of computing the initial approximations A_s (and making some obvious simplifications) establishes the lemma. □

Lemma 3.3.

$$\Delta_k \le \frac{1}{c_0^2 r^{1+\lfloor d/2 \rfloor / c}} \left(\sum_{s \in \mathscr{T}(R^{\frown})} |S_{k|s}|^{dc} \right)^{1/c},$$

where $c = c_1^{2/3}$.

Proof. To begin with, notice that $\mathscr{E}_1$ has no drift, therefore

$$\Delta_k = -\mathbf{E}[\mathscr{E}_2^*(S_k)] + \frac{1}{n-r-k} \sum_{h \in H \setminus (R \cup S_k)} \mathbf{E}[\mathscr{E}_2^*(S_k \cup \{h\})].$$

Let $\alpha = n - r - k$ and $\beta = r - k$; we have

$$\Delta_k = \frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_{s \in \mathscr{T}(R^{\cap})} B_s,$$

where

$$B_s = \sum_{h \in H \backslash (R \cup S_k)} \sum_v \frac{1}{\alpha} \left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v} \delta_v^{c_1} \binom{\alpha - \delta_v - d + \varphi_v - 1}{\beta - d + \varphi_v - 1} \Big/ \binom{\alpha - 1}{\beta - 1}$$
$$- \sum_v \left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v} \delta_v^{c_1} \binom{\alpha - \delta_v - d + \varphi_v}{\beta - d + \varphi_v} \Big/ \binom{\alpha}{\beta}.$$

In the first sum, v ranges over all the vertices of $\mathscr{A}(R \cup S_k \cup A_s \cup \{h\})$ within $s \cap (S_k \cup \{h\})^{\cap}$, while in the second one, v ranges over all the vertices of $\mathscr{A}(R \cup S_k \cup A_s)$ within $s \cap S_k^{\cap}$. Note that φ_v is defined with respect to $R \cup S_k \cup \{h\}$ in the first sum and with respect to $R \cup S_k$ in the second sum. We further derive

$$\Delta_k = \frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_{s \in \mathscr{T}(R^{\cap})} \sum_v F_v \delta_v^{c_1},$$

where v ranges over all the vertices within $s \cap S_k^{\cap}$ formed by any of the arrangements $\mathscr{A}(R \cup S_k \cup A_s \cup \{h\})$ for $h \in H \backslash (R \cup S_k \cup A_s)$. To express F_v, we distinguish between two cases. (In the following, φ_v is defined with respect to $R \cup S_k$.) If v is a vertex of $\mathscr{A}(R \cup S_k \cup A_s)$, then what is the contribution of v to the positive sum in B_s? The δ_v hyperplanes h cutting Ov contribute nothing; the $d - \varphi_v$ hyperplanes passing through v, not in $R \cup S_k$, each contribute one term, with φ_v raised by 1. Finally, the other $\alpha - \delta_v - d + \varphi_v$ hyperplanes contribute one term each, with φ_v unchanged:

$$F_v = \left(\frac{\alpha - \delta_v - d + \varphi_v}{\alpha}\right)\left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v} \binom{\alpha - \delta_v - d + \varphi_v - 1}{\beta - d + \varphi_v - 1} \Big/ \binom{\alpha - 1}{\beta - 1}$$
$$+ \left(\frac{d - \varphi_v}{\alpha}\right)\left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v-1} \binom{\alpha - \delta_v - d + \varphi_v}{\beta - d + \varphi_v} \Big/ \binom{\alpha - 1}{\beta - 1}$$
$$- \left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v} \binom{\alpha - \delta_v - d + \varphi_v}{\beta - d + \varphi_v} \Big/ \binom{\alpha}{\beta}.$$

We derive

$$F_v = \left(\frac{d - \varphi_v}{\alpha}\right)\left(1 - \frac{|H_{|s}|}{|A_s|}\right)\left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v-1} \binom{\alpha - \delta_v - d + \varphi_v}{\beta - d + \varphi_v} \Big/ \binom{\alpha - 1}{\beta - 1},$$

and, hence,

$$F_v = \binom{d-\varphi_v}{\beta}\left(1 - \frac{|H_{|s}|}{|A_s|}\right)\left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v-1}\binom{\alpha-\delta_v-d+\varphi_v}{\beta-d+\varphi_v}\bigg/\binom{\alpha}{\beta}. \qquad \text{(A.4)}$$

If v is not a vertex of $\mathscr{A}(R \cup S_k \cup A_s)$, then it must be formed by the intersection of $d-1$ hyperplanes in $R \cup S_k \cup A_s$ and one other hyperplane h not in $R \cup S_k \cup A_s$. This hyperplane must be chosen in order for v to contribute to B_s; note that picking h raises φ_v by 1. No contribution can be made to the negative sum.

$$\begin{aligned} F_v &= \frac{1}{\alpha}\left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v-1}\binom{\alpha-\delta_v-d+\varphi_v}{\beta-d+\varphi_v}\bigg/\binom{\alpha-1}{\beta-1} \\ &= \frac{1}{\beta}\left(\frac{|H_{|s}|}{|A_s|}\right)^{d-\varphi_v-1}\binom{\alpha-\delta_v-d+\varphi_v}{\beta-d+\varphi_v}\bigg/\binom{\alpha}{\beta}. \end{aligned}$$

Note that we use the convention that

$$\binom{a}{b} = 0 \qquad \text{if} \quad b < 0.$$

Let $\Delta_k(j)$ be the contribution to Δ_k of the vertices v such that $j = d - \varphi_v$. These vertices lie on j-dimensional flats defined by hyperplanes in $R \cup S_k$. Any $s \in \mathscr{T}(R^\cap)$ that contains such vertices must be of dimension at least j. Since these are also vertices of $s \cap S_k^\cap$, it is possible (by triangulation of $s \cap \mathscr{A}(R \cup S_{k|s})$) to identify $O(|S_{k|s}|^{\lfloor d/2 \rfloor})$ disjoint j-dimensional simplices that collectively contain all these vertices; actually, we use the conservative bound $O(|S_{k|s}|^d)$ for convenience. Let σ be one of them, and let e be a line segment formed by the intersection of σ with $j-1$ hyperplanes of A_s. By (A.4), it is easy to see that $\Delta_k(j) = 0$, if $j = 0$ or $j > \beta$, so let us assume that $0 < j \leq \beta$. A vertex v of $\mathscr{A}(R \cup S_k \cup A_s) \cap \sigma \cap S_k^\cap$ appears on j distinct edges e, therefore, by summing over all e, it will be counted j times if it is a vertex of $\mathscr{A}(R \cup S_k \cup A_s)$ and only once if it is not, i.e., if it lies on exactly one hyperplane not in $R \cup S_k \cup A_s$. Thus, we can immediately verify that

$$\begin{aligned} \Delta_k(j) = {} & \frac{1}{c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_s \sum_\sigma \sum_e \\ & \times \left\{ \sum_{v \in A_{s|e}} \binom{\delta_v^{c_1}}{j}\binom{j}{\beta}\left(1 - \frac{|H_{|s}|}{|A_s|}\right)\left(\frac{|H_{|s}|}{|A_s|}\right)^{j-1}\binom{\alpha-\delta_v-j}{\beta-j}\bigg/\binom{\alpha}{\beta} \right. \\ & \left. \quad + \sum_{v \in (H \setminus A_s)_{|e}} \frac{\delta_v^{c_1}}{\beta}\left(\frac{|H_{|s}|}{|A_s|}\right)^{j-1}\binom{\alpha-\delta_v-j}{\beta-j}\bigg/\binom{\alpha}{\beta} \right\} \end{aligned}$$

In the first (resp. second) inner sum v ranges over all vertices formed by the intersection of e with the hyperplanes in A_s (resp. not in A_s). Therefore,

$$\Delta_k(j) = \frac{1}{c_0 r^{\lfloor d/2 \rfloor} (n/r)^{c_1}} \sum_s \sum_\sigma \sum_e \frac{1}{\beta} \left(\frac{|H_{|s}|}{|A_s|} \right)^{j-1} G_e,$$

where

$$\begin{aligned} G_e &= \sum_{v \in A_{s|e}} \delta_v^{c_1} \left(1 - \frac{|H_{|s}|}{|A_s|}\right) \binom{\alpha - \delta_v - j}{\beta - j} \Big/ \binom{\alpha}{\beta} + \sum_{v \in (H \setminus A_s)_{|e}} \delta_v^{c_1} \binom{\alpha - \delta_v - j}{\beta - j} \Big/ \binom{\alpha}{\beta} \\ &= \sum_{v \in H_{|e}} \delta_v^{c_1} \binom{\alpha - \delta_v - j}{\beta - j} \Big/ \binom{\alpha}{\beta} - \sum_{v \in A_{s|e}} \delta_v^{c_1} \left(\frac{|H_{|s}|}{|A_s|}\right) \binom{\alpha - \delta_v - j}{\beta - j} \Big/ \binom{\alpha}{\beta}. \end{aligned}$$

To find an upper bound on G_e we are now ready to use the sampling-based integration method introduced in the proof of Lemma 3.1. Note that we can trivially assume that $H_{|s+} \neq \varnothing$ (else s can be ignored), therefore $\rho_s > 0$. Let

$$x_s = \left\lceil \frac{|H_{|s+}|}{\sqrt{\rho_s}} \right\rceil.$$

By subdividing e into subsegments e' such that $H_{|e'}$ has size x_s or less, it appears that

$$\sum_{v \in H_{|e'}} 1 \leq \frac{|H_{|s}|}{\rho_s} + \left(\frac{|H_{|s}|}{|A_s|}\right) \sum_{v \in A_{s|e'}} 1,$$

therefore

$$\begin{aligned} &\sum_{v \in H_{|e'}} \delta_v^{c_1} \binom{\alpha - \delta_v - j}{\beta - j} \Big/ \binom{\alpha}{\beta} \\ &\qquad \leq \sum_{v \in A_{s|e'}} (\delta_v + x_s)^{c_1} \left(\frac{|H_{|s}|}{|A_s|}\right) \binom{\alpha - \delta_v + x_s - j}{\beta - j} \Big/ \binom{\alpha}{\beta} + \left(\frac{|H_{|s}|}{\rho_s}\right) \delta_w^{c_1} \binom{\alpha - j}{\beta - j} \Big/ \binom{\alpha}{\beta}, \end{aligned}$$

where $\delta_w = \max_v \{\delta_v | v \in H_{|e'}\}$. Since $\delta_w \leq |H_{|s+}|$, it follows that

$$\begin{aligned} \sum_{v \in H_{|e}} \delta_v^{c_1} \binom{\alpha - \delta_v - j}{\beta - j} \Big/ \binom{\alpha}{\beta} &\leq \sum_{v \in A_{s|e}} (\delta_v + x_s)^{c_1} \left(\frac{|H_{|s}|}{|A_s|}\right) \binom{\alpha - \delta_v + x_s - j}{\beta - j} \Big/ \binom{\alpha}{\beta} \\ &\quad + \left\lceil \frac{|H_{|e}|}{x_s} \right\rceil \left(\frac{|H_{|s}|}{\rho_s}\right) |H_{|s+}|^{c_1} \binom{\alpha - j}{\beta - j} \Big/ \binom{\alpha}{\beta}. \end{aligned}$$

From the fact that $\rho_s = 2^{c_0}\lceil r|H_{|s+}|/n\rceil^{\sqrt{c_1}}$, we easily verify that $x_s \le n/(4r)$, and, therefore,

$$\left(1 + \frac{x_s}{\alpha - \delta_v - \beta + 1}\right)^{\beta - j} \le \left(1 + \frac{2x_s}{n}\right)^r \le 1 + \frac{c_1 x_s r}{n}.$$

Consequently,

$$\binom{\alpha - \delta_v + x_s - j}{\beta - j}\bigg/\binom{\alpha}{\beta} \le \left(1 + \frac{x_s}{\alpha - \delta_v - \beta + 1}\right)^{\beta - j}\binom{\alpha - \delta_v - j}{\beta - j}\bigg/\binom{\alpha}{\beta}$$
$$\le \left(1 + \frac{c_1 x_s r}{n}\right)\binom{\alpha - \delta_v - j}{\beta - j}\bigg/\binom{\alpha}{\beta}.$$

We have

$$\left\lceil \frac{|H_{|e}|}{x_s} \right\rceil \le \frac{|A_{s|e}| \times |H_{|s}|}{x_s |A_s|} + \frac{|H_{|s}|}{x_s \rho_s} + 1,$$

therefore

$$\sum_{v \in H_{|e}} \delta_v^{c_1}\binom{\alpha - \delta_v - j}{\beta - j}\bigg/\binom{\alpha}{\beta}$$
$$\le \sum_{v \in A_{s|e}} \delta_v^{c_1}\left(\frac{|H_{|s}|}{|A_s|}\right)\left(1 + \frac{c_1 x_s r}{n}\right)\binom{\alpha - \delta_v - j}{\beta - j}\bigg/\binom{\alpha}{\beta}$$
$$+ \sum_{v \in A_{s|e}} ((\delta_v + x_s)^{c_1} - \delta_v^{c_1})\left(\frac{|H_{|s}|}{|A_s|}\right)\left(1 + \frac{c_1 x_s r}{n}\right)\binom{\alpha - \delta_v - j}{\beta - j}\bigg/\binom{\alpha}{\beta}$$
$$+ \frac{|H_{|s}|}{x_s \rho_s}\left(\frac{|A_{s|e}| \times |H_{|s}|}{|A_s|} + \frac{|H_{|s}|}{\rho_s} + x_s\right)\left(\frac{\beta}{\alpha}\right)^j |H_{|s+}|^{c_1}.$$

If we substitute the right-hand side of the inequality into the expression for G_e, we find that the 1 of the factor $(1 + c_1 x_s r/n)$ cancels out. Specifically, we derive

$$G_e \le \sum_{v \in A_{s|e}} \delta_v^{c_1}\left(\frac{|H_{|s}|}{|A_s|}\right)\left(\frac{c_1 x_s r}{n}\right)\binom{\alpha - \delta_v - j}{\beta - j}\bigg/\binom{\alpha}{\beta}$$
$$+ \sum_{v \in A_{s|e}} \sum_{0 \le i < c_1} \binom{c_1}{i}\delta_v^i x_s^{c_1 - i}\left(\frac{|\mathrm{H}_{|s}|}{|A_s|}\right)\left(1 + \frac{c_1 x_s r}{n}\right)\binom{\alpha - \delta_v - j}{\beta - j}\bigg/\binom{\alpha}{\beta}$$
$$+ \frac{1}{x_s \rho_s}\left(\frac{\beta}{\alpha}\right)^j\left(\frac{|A_{s|e}|}{|A_s|}\right)|H_{|s+}|^{c_1 + 2} + \frac{1}{x_s \rho_s^2}\left(\frac{\beta}{\alpha}\right)^j |H_{|s+}|^{c_1 + 2} + \frac{1}{\rho_s}\left(\frac{\beta}{\alpha}\right)^j |H_{|s+}|^{c_1 + 1}.$$

Note that $c_1 x_s r/n$ is less than a constant. Since there are $O(|A_{|s}|^{j-1})$ segments e for each j-dimensional σ and $O(|S_{k|s}|^d)$ simplices σ for each s, we now find that

$$\Delta_k(j) \le \frac{c_1}{\beta c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \sum_{s \in \mathcal{T}(R^\frown)} \left(\frac{|\mathrm{H}_{|s}|}{|A_s|}\right)^{j-1} |S_{k|s}|^d$$
$$\times \left\{\left(\frac{c_1 x_s r}{n}\right)\left(\frac{\beta}{\alpha}\right)^j |H_{|s+}|^{c_1+1} \times |A_s|^{j-1}\right.$$
$$+ c_1 2^{c_1}\left(\frac{\beta}{\alpha}\right)^j x_s |H_{|s+}|^{c_1} \times |A_s|^{j-1} + \frac{1}{x_s \rho_s}\left(\frac{\beta}{\alpha}\right)^j |H_{|s+}|^{c_1+2} \times |A_s|^{j-1}$$
$$\left. + \frac{1}{x_s \rho_s^2}\left(\frac{\beta}{\alpha}\right)^j |H_{|s+}|^{c_1+2} \times |A_s|^{j-1} + \frac{1}{\rho_s}\left(\frac{\beta}{\alpha}\right)^j |H_{|s+}|^{c_1+1} \times |A_s|^{j-1}\right\}.$$

If $x_s = 1$, then $\rho_s \ge |H_{|s+}|^2 \ge |H_{|s}|^2$ and therefore $A_s = H_{|s}$. This implies that $G_e = 0$ for $e \subseteq s$, and so we can assume that $x_s > 1$ in the summation below.

$$\Delta_k(j) \le \frac{2c_1}{\beta c_0 r^{\lfloor d/2 \rfloor}(n/r)^{c_1}} \left(\frac{\beta}{\alpha}\right)^j \sum_{s \in \mathcal{T}(R^\frown)} |S_{k|s}|^d$$
$$\times \left\{\left(\frac{c_1 x_s r}{n}\right)|H_{|s+}|^{c_1+j} + c_1 2^{c_1} x_s |H_{|s+}|^{c_1+j-1}\right.$$
$$\left. + \left(\frac{1}{x_s \rho_s}\right)|H_{|s+}|^{c_1+j+1} + \left(\frac{1}{\rho_s}\right)|H_{|s+}|^{c_1+j}\right\}.$$

Since $x_s > 1$, we have

$$x_s \le 2^{1-c_0/2}\left(\frac{n}{r}\right)^{\sqrt{c_1}/2} |H_{|s+}|^{1-\sqrt{c_1}/2}.$$

From the inequalities,

$$x_s \rho_s \ge 2^{c_0/2}\left(\frac{r}{n}\right)^{\sqrt{c_1}/2} |H_{|s+}|^{\sqrt{c_1}/2+1}$$

and

$$\rho_s \ge 2^{c_0}\left(\frac{r}{n}\right)^{\sqrt{c_1}} |H_{|s+}|^{\sqrt{c_1}},$$

we derive

$$\Delta_k(j) \le \frac{1}{c_0^4 r^{\lfloor d/2 \rfloor}}\left(\frac{1}{\beta}\right)\left(\frac{\beta}{\alpha}\right)^j\left(\frac{n}{r}\right)^{\sqrt{c_1}/2-c_1-1} \sum_{s \in \mathcal{T}(R^\frown)} |S_{k|s}|^d \times |H_{|s+}|^{c_1+j+1-\sqrt{c_1}/2}$$
$$+ \frac{1}{c_0^4 r^{\lfloor d/2 \rfloor}}\left(\frac{1}{\beta}\right)\left(\frac{\beta}{\alpha}\right)^j\left(\frac{n}{r}\right)^{\sqrt{c_1}/2-c_1} \sum_{s \in \mathcal{T}(R^\frown)} |S_{k|s}|^d \times |H_{|s+}|^{c_1+j-\sqrt{c_1}/2}$$
$$+ \frac{1}{c_0^4 r^{\lfloor d/2 \rfloor}}\left(\frac{1}{\beta}\right)\left(\frac{\beta}{\alpha}\right)^j\left(\frac{n}{r}\right)^{\sqrt{c_1}-c_1} \sum_{s \in \mathcal{T}(R^\frown)} |S_{k|s}|^d \times |H_{|s+}|^{c_1+j-\sqrt{c_1}}.$$

By Hölder's inequality,

$$\sum_s |S_{k|s}|^d \times |H_{|s+}|^{c_1+j+1-\sqrt{c_1}/2} \le \left(\sum_s |S_{k|s}|^{dc_1^{2/3}}\right)^{1/c_1^{2/3}} \left(\sum_s |H_{|s+}|^{p_1}\right)^q,$$

where $p_1 = c_1^{2/3}(c_1 + j + 1 - \sqrt{c_1}/2)/(c_1^{2/3} - 1)$ and $q = 1 - 1/c_1^{2/3}$. Similarly,

$$\sum_s |S_{k|s}|^d \times |H_{|s+}|^{c_1+j-\sqrt{c_1}/2} \le \left(\sum_s |S_{k|s}|^{dc_1^{2/3}}\right)^{1/c_1^{2/3}} \left(\sum_s |H_{|s+}|^{p_2}\right)^q,$$

where $p_2 = c_1^{2/3}(c_1 + j - \sqrt{c_1}/2)/(c_1^{2/3} - 1)$. Finally, we have

$$\sum_s |S_{k|s}|^d \times |H_{|s+}|^{c_1+j-\sqrt{c_1}} \le \left(\sum_s |S_{k|s}|^{dc_1^{2/3}}\right)^{1/c_1^{2/3}} \left(\sum_s |H_{|s+}|^{p_3}\right)^q,$$

where $p_3 = c_1^{2/3}(c_1 + j - \sqrt{c_1})/(c_1^{2/3} - 1)$. Because $\mathscr{T}(R^\frown)$ is a semicutting and $p_3 < p_2 < p_1 \le c_1$ (recall that c_1 can be chosen as large as we want) we have, for some constant $b = b(c_1)$,

$$\sum_{s \in \mathscr{T}(R^\frown)} |H_{|s+}|^{p_i} = O(1) \times \sum_{s \in \mathscr{T}(R^\frown)} |H_{|s}|^{p_i} \le bc_0 r^{\lfloor d/2 \rfloor}\left(\frac{n}{r}\right)^{p_i},$$

therefore

$$\Delta_k(j) \le \frac{1}{c_0^3}\left(\frac{1}{\beta}\right)\left(\frac{\beta}{\alpha}\right)^j\left(\frac{n}{r}\right)^j r^{\lfloor d/2 \rfloor(q-1)}\left(\sum_{s \in \mathscr{T}(R^\frown)} |S_{k|s}|^{dc_1^{2/3}}\right)^{1/c_1^{2/3}}.$$

Since we have assumed that $j > 0$,

$$\frac{1}{r-k}\left(\frac{r-k}{n-r-k}\right)^j\left(\frac{n}{r}\right)^j = \frac{n}{(n-r-k)r}\left(\frac{n(r-k)}{r(n-r-k)}\right)^{j-1} = O\left(\frac{1}{r}\right),$$

and, hence,

$$\Delta_k(j) \le \frac{c_1}{c_0^3} r^{\lfloor d/2 \rfloor(q-1)-1}\left(\sum_{s \in \mathscr{T}(R^\frown)} |S_{k|s}|^{dc_1^{2/3}}\right)^{1/c_1^{2/3}}.$$

Adding up $\Delta_k(j)$ for all $j \le d$ completes the proof. □

Lemma 3.4. *The energy of $R \cup S$ at the end of round t is less than* 1.

Proof. It suffices to prove that

$$\mathbf{E}^*[\mathscr{E} \mid S_k \subseteq S] < \tfrac{1}{2}\left(1 + \frac{k}{r}\right),$$

which we do by induction. The case $k = 0$ follows from Lemma 3.1. Assume that it holds up to $k < r$. From the analysis of Lemma 2.4 and from Lemma 3.3, we derive

$$\begin{aligned}\mathbf{E}^*[\mathscr{E} \mid S_{k+1} \subseteq S] &\leq \mathbf{E}^*[\mathscr{E} \mid S_k \subseteq S] + \Delta_k \\ &< \tfrac{1}{2}\left(1 + \frac{k}{r}\right) + \frac{1}{c_0^2 r^{\lfloor d/2 \rfloor / c_1^{2/3} + 1}} \left(\sum_{s \in \mathscr{T}(R^\cap)} |S_{k|s}|^{dc_1^{2/3}} \right)^{1/c_1^{2/3}}.\end{aligned}$$

From the induction hypothesis we find that

$$\mathbf{E}[\mathscr{E}_1 \mid S_k \subseteq S] \leq \mathbf{E}^*[\mathscr{E} \mid S_k \subseteq S] < 1,$$

therefore

$$\sum_{s \in \mathscr{T}(R^\cap)} |S_{k|s}|^{dc_1^{2/3}} < c_0^2 r^{\lfloor d/2 \rfloor}$$

and

$$\begin{aligned}\mathbf{E}^*[\mathscr{E} \mid S_{k+1} \subseteq S] &< \tfrac{1}{2}\left(1 + \frac{k}{r}\right) + \frac{(c_0^2 r^{\lfloor d/2 \rfloor})^{1/c_1^{2/3}}}{c_0^2 r^{\lfloor d/2 \rfloor / c_1^{2/3} + 1}} \\ &< \tfrac{1}{2}\left(1 + \frac{k+1}{r}\right),\end{aligned}$$

which completes the induction. □

References

1. Agarwal, P. K. Partitioning arrangements of lines, I: An efficient deterministic algorithm, *Discrete Comput. Geom.* **5** (1990), 449–483.
2. Brown, K. Q. Voronoi diagrams from convex hulls, *Inform. Process. Lett.* **9** (1979), 223–228.
3. Chazelle, B. Cutting hyperplanes for divide-and-conquer, *Discrete Comput. Geom.* **9** (1993), 145–158.
4. Chazelle, B., Friedman, J. A deterministic view of random sampling and its use in geometry, *Combinatorica* **10** (1990), 229–249.
5. Chazelle, B., Matoušek, J. Derandomizing an output-sensitive convex hull algorithm in three dimensions (submitted for publication).

6. Clarkson, K. L. A randomized algorithm for closest-point queries, *SIAM J. Comput.* **17** (1988), 830–847.
7. Clarkson, K. L. Randomized geometric algorithms, in *Euclidean Geometry and Computers*, D. Z. Du and F. K. Hwang, eds., World Scientific, to appear.
8. Clarkson, K. L., Shor, P. W. Applications of random sampling in computational geometry, II, *Discrete Comput. Geom.* **4** (1989), 387–421.
9. Edelsbrunner, H. *Algorithms in Combinatorial Geometry*, Springer-Verlag, Heidelberg, 1987.
10. Edelsbrunner, H., Seidel, R. Voronoi diagrams and arrangements, *Discrete Comput. Geom.* **1** (1986), 25–44.
11. Graham, R. L. An efficient algorithm for determining the convex hull of a planar point set, *Inform. Process. Lett.* **1** (1972), 132–133.
12. Kirkpatrick, D. G., Seidel R. The ultimate planar convex hull algorithm? *SIAM J. Comput.* **15** (1986), 287–299.
13. Matoušek, J. Approximations and optimal geometric divide-and-conquer, *Proc. 23rd Annual ACM Symp. on Theory of Computing*, 1991, pp. 505–511.
14 Matoušek, J. Efficient partition trees, *Proc. 7th Annual ACM Symp. on Computational Geometry*, 1991, pp. 1–9.
15. Matoušek, J. Cutting hyperplane arrangements, *Discrete Comput. Geom.* **6** (1991), 385–406.
16. Matoušek, J. Linear optimization queries, *J. Algorithms*, to appear.
17. Preparata, F. P., Hong, S. J. Convex hulls of finite sets of points in two and three dimensions, *Comm. ACM* **20** (1977), 87–93.
18. Raghavan, P. Probabilistic construction of deterministic algorithms: approximating packing integer programs, *J. Comput. System Sci.* **37** (1988), 130–143.
19. Seidel, R. A convex hull algorithm optimal for point sets in even dimensions, Technical Report 81-14, University of British Columbia, 1981.
20. Seidel, R. Constructing higher-dimensional convex hulls at logarithmic cost per face, *Proc. 18th Annual ACM Symp. on Theory of Computing*, 1986, pp. 404–413.
21. Seidel, R. Small-dimensional linear programming and convex hulls made easy, *Discrete Comput. Geom.* **6** (1991), 423–434.
22. Spencer, J. *Ten Lectures on the Probabilistic Method*, CBMS–NSF, SIAM, Philadelphia, PA, 1987.
23. Vapnik, V. N., Chervonenkis, A. Ya. On the uniform convergence of relative frequencies of events to their probabilities, *Theory Probab. Appl.* **16** (1971), 264–280.

Received July 9, 1991, and in revised form May 28, 1992, and March 8, 1993.