

How Hard Is Half-Space Range Searching?*

Hervé Brönnimann,¹ Bernard Chazelle,¹ and János Pach²

¹ Department of Computer Science, Princeton University,
 35 Olden Street, Princeton, NJ 08544-2087, USA
 hbr@cs.princeton.edu
 chazelle@cs.princeton.edu

² Department of Computer Science, City College, C.U.N.Y.,
 New York, NY 10031, USA, and
 Courant Institute, N.Y.U., 251 Mercer Street,
 New York, NY 10012, USA
 pach@cims6.nyu.edu

Abstract. We investigate the complexity of *half-space range searching*: given n points in d -space, build a data structure that allows us to determine efficiently how many points lie in a query half-space. We establish a tradeoff between the storage m and the worst-case query time t in the Fredman/Yao arithmetic model of computation. We show that t must be at least on the order of

$$\frac{(n/\log n)^{1-(d-1)d(d+1)}}{m^{1/d}}.$$

Although the bound is unlikely to be optimal, it falls reasonably close to the recent upper bound of $O(n/m^{1/d})$ established by Matoušek. We also show that it is possible to devise a sequence of n inserts and half-space range queries that require a total time of $n^{2-O(1/d)}$. Our results imply the first nontrivial lower bounds for spherical range searching in any fixed dimension. For example, they show that, with linear storage, circular range queries in the plane require $\Omega(n^{1/3})$ time (modulo a logarithmic factor).

1. Introduction

A considerable amount of attention has been given to *simplex range searching* in the last few years [1], [4]–[13], [16], [19], [21], [22], [26], [27], [29]. This is the

* Work by Hervé Brönnimann and Bernard Chazelle has been supported in part by NSF Grant CCR-90-02352 and the Geometry Center. Janos Pach was supported by NSF Grant CCR-91-22103 and OTKA-4269.

problem of preprocessing a set P of n points in Euclidean d -space so that, given an arbitrary “query” simplex s , which points lie in s can be found efficiently. The problem comes under various guises, depending on whether it is wished to enumerate the points or simply count them. All these variants can be unified into a common formulation by attaching weights to the points and endowing the weights with a semigroup structure [3], [4], [15], [16], [28].

If m is the amount of storage, a lower bound on the query time of $\Omega(n/m^{1/2})$, if $d = 2$, and $\Omega((n/\log n)/m^{1/d})$, if $d > 2$, was established by Chazelle [4] in the *arithmetic* model of computation. The model, which is due to Fredman and Yao [3], [15], [16], [28], is general enough to encompass all known algorithms for the counting problem. Later, Chazelle *et al.* [9] provided a quasi-matching upper bound (up to within a factor of n^ϵ , for any fixed $\epsilon > 0$). More recently, Matoušek [22] lowered the multiplicative factor to polylogarithmic. It thus appears that—at least theoretically—simplex range searching is a solved problem.

However, what happens if we restrict the query simplex to be a half-space? (Note that any simplex large enough acts as a half-space with respect to the n points.) *Half-space range searching*, as the problem is known, is particularly interesting because, as was observed in [29], just about any form of algebraic range searching can be “lifted” into such a problem. All kinds of intriguing questions immediately arise. If we insist on *reporting* the points, then the half-space problem is considerably easier than the simplex version. For example, if we wish to achieve optimal query time, the former can be solved using only $O(n^{\lfloor d/2 \rfloor + \epsilon})$ space, for any fixed $\epsilon > 0$ [11], [21]. By contrast, simplex range reporting provably requires $\Omega(n^{d-\epsilon})$ storage (on a pointer machine) [8].

In the counting version of the problem, however, no one has yet been able to take (significant) advantage of the fact that a half-space is a restricted form of simplex to derive a faster algorithm. Indeed, the upper bound of $O(n/m^{1/d})$ for half-space range searching established by Matoušek [22] is only a polylogarithmic improvement over the best bound for simplex range searching. This suggests that the two problems might have similar intrinsic complexity, but unfortunately no lower bound has been established for half-space range searching. This paper partially fills this gap by establishing a lower bound, which although nonoptimal, is probably not too far from the truth (at least in higher dimensions). We prove that if m units of storage are available, then the worst-case query time must be on the order of

$$\frac{(n/\log n)^{1-(d-1)/d(d+1)}}{m^{1/d}},$$

which is

$$\frac{n^{1-(1/d)-O(1/d^2)}}{m^{1/d}}.$$

We also show that it is possible to devise a sequence of n inserts and half-space

range queries, so that, starting with the empty set, carrying out the sequence of operations on-line requires time proportional to

$$\frac{n^{2-(3d+1)/(d+1)^2}}{(\log n)^{(d^2+1)/(d+1)^2}},$$

which is $n^{2-O(1/d)}$. The only related result previously known is for the two-dimensional case and requires deletions as well as insertions: it was shown by Fredman [16] that a mixed sequence of insertions, deletions, and queries takes $\Omega(n^{4/3})$ time in the worst case. It is interesting to notice that because half-spaces are special cases of d -balls, our lower bounds hold for spherical range searching in any fixed dimension. For example, we derive that, with linear storage, circular range queries in the plane require $\Omega(n^{1/3})$ time (modulo a logarithmic factor).

A close examination of the lower bound proof for simplex range searching given in [4] shows that although some tools can be salvaged, the basic technique cannot be extended to the half-space case and new ideas are needed. Let us briefly outline our line of attack in an informal, intuitive fashion.

In the arithmetic model of computation, we can think of each point $p_i \in P$ as a variable x_i , and a query τ as a linear form $\sum_{i \in J(\tau)} x_i$, where $J(\tau) = \{i: p_i \in \tau\}$. The data structure can be modeled as a collection of m precomputed linear forms called *generators*, g_1, \dots, g_m . The time t to answer a query τ is given by the smallest number of generators needed to express the linear form associated with τ , i.e., the smallest subset $\{g_{i_1}, \dots, g_{i_t}\}$, such that

$$\sum_{i \in J(\tau)} x_i = \alpha_1 g_{i_1} + \dots + \alpha_t g_{i_t}$$

for integral $\alpha_i > 0$.

Consider the bipartite graph $G = (V, W, E)$, where V is the set of variables, W is the set of linear forms associated with all possible queries, and the edges of E connect the variables to the linear forms in W where they appear. The starting point of the lower-bound proof for simplex range searching [4] is the observation that efficient query-answering is related to the existence of large complete bipartite subgraphs (c.b.s) in G , where “large” refers to the number of edges in the c.b.s. Proving that G does not have any large c.b.s. immediately implies a lower bound on the maximum value of t as a function of m . To prove the absence of large c.b.s., a geometric analog of the graph is built (whereby generators map to certain convex polytopes) and the problem reduces to proving an isoperimetric inequality in integral geometry.

The trouble in pursuing this line of attack is that in the case of half-space range searching the graph G *does* have very large c.b.s., so that the basic approach is doomed. We go around this difficulty by using the following trick: we weight each edge (x_i, τ) of the graph as a function of how far the point p_i lies from the boundary of the half-space τ . The intuition behind this weighting strategy is that what makes a query hard are the points near the boundary and not so much those squarely

in the middle. Our lucky break is that it is precisely the points in the middle that create large c.b.s. Indeed, with an appropriate weighting function designed to emphasize the points near the boundary we can make all large (weighted) c.b.s. magically disappear. To prove this we must establish a more complex isoperimetric inequality than the one proven in [4]. This is done by using some of the same integral-geometric tools, as well as a useful geometric fact of independent interest, which we call the *slicing lemma*: it asserts that a compact convex body can be peeled into few chunks, so that any big enough slice of the body fully contains at least one chunk of roughly the same volume.

2. Preliminaries

In this section we briefly review the arithmetic model of computation. We introduce our notation and present useful facts from discrepancy theory and integral geometry. Finally, we state and prove the slicing lemma. Throughout this paper we assume that $d > 1$.

2.1. The Model of Computation

Let $(S, +)$ be a faithful commutative semigroup; recall that a semigroup is called faithful [28] if any two identically equal linear forms have the same set of variables. Let $P = \{p_1, \dots, p_n\}$ be a set of n points in $[0, 1]^d$, and let w (the weight function) be an assignment of each point p_i to a semigroup value in S . We define a function *answ* mapping half-spaces τ to S :

$$\text{answ}(\tau) = \sum_{p \in P \cap \tau} w(p).$$

Let x_1, \dots, x_n be n variables over S . A *generator* $g(x_1, \dots, x_n)$ is a linear form $\sum_{1 \leq i \leq n} \alpha_i x_i$, where the α_i 's are nonnegative integers. A *storage scheme* Γ of size m is a collection of m generators $\{g_1, \dots, g_m\}$ satisfying the following property: for any closed half-space τ ($P \cap \tau \neq \emptyset$), there exist $J \subseteq [1 \dots m]$ and a set of labeled integers $\{\beta_i > 0 \mid i \in J\}$ such that the relation

$$\text{answ}(\tau) = \sum_{i \in J} \beta_i g_i(w(p_1), \dots, w(p_n))$$

holds for any weight function w over P . The size of the smallest such J is the query time for τ . Note that a storage scheme can be dependent on the particular semigroup under consideration and also take advantage of any property which P may enjoy: however, it must hold for any assignment of semigroup values to P .

Given a linear form $\sum_{1 \leq i \leq n} \alpha_i x_i$, the set of points $\{p_i \mid \alpha_i \neq 0\}$ is called its *cluster*. Let C_1, \dots, C_m be the clusters associated with the generators of the storage scheme. Given any closed half-space τ , let A_τ be the smallest set of indices such that

$P \cap \tau = \bigcup_{i \in A_\tau} C_i$. It is shown in [4] that, given P and m units of storage, the worst-case query time is at least equal to the minimum, over all set-systems C_1, \dots, C_m , of $\max_\tau |A_\tau|$.

2.2. Three Geometric Lemmas

We need to build a number of technical tools. First, by relying on a random input distribution we are able to relate the number of input points in a given query to its volume. This allows us to approximate a discrete measure by a continuous one and thereby open the door to an analytical attack. Next, we introduce a well-behaved probability measure over the set of queries and estimate the probability that a random query contains a convex body of fixed volume. Finally, we prove the slicing lemma mentioned earlier.

Let P be a set of n points in $[0, 1]^d$ and let vol denote the Lebesgue measure over E^d . We say that P is *scattered* if, given any convex set $C \subseteq [0, 1]^d$,

$$\frac{1}{a} n \text{vol } C - \log n \leq |P \cap C| \leq an \text{vol } C + \log n$$

for some constant $a > 1$ that depends only on d .

Lemma 2.1. *A random set of n points chosen uniformly and independently in $[0, 1]^d$ is scattered with probability $1 - o(1)$.*

Proof. It is shown in [4] that, with probability at least $1 - 1/n$, every convex set $C \subseteq [0, 1]^d$ satisfies $|P \cap C| \leq an \text{vol } C + \log n$. We establish the other inequality by using the following well-known result:

Lemma 2.2. *Given any $0 < \rho < 1$, there exists a collection of $m \leq a_1/\rho^{a_2}$ convex bodies K_1, \dots, K_m , where a_1, a_2 are positive constants that depend only on d , such that any convex set $K \subseteq [0, 1]^d$ of volume ρ encloses some K_i , where $\text{vol } K_i \geq (1/a_2) \text{vol } K$.*

Proof. A straightforward variation of the proof of Lemma 4.8 [4]. □

Set $\rho = a_3(\log n)/n$, where a_3 is some appropriate constant. We can assume that, for all i , $K_i \subseteq [0, 1]^d$ and $\text{vol } K_i \geq \rho/a_2$. By Chernoff's bound [25] we know that, for small enough $\varepsilon > 0$ and any fixed i ,

$$\text{Prob}[|P \cap K_i| < n(\text{vol } K_i)(1 - \varepsilon)] < \exp(-\frac{1}{2}\varepsilon^2 n \text{vol } K_i).$$

Set $\varepsilon = 1 - a_2/a_3$; by choosing a_3 large enough, we verify that

$$\text{Prob}[|P \cap K_i| < \log n] < 1 - \rho^{a_2+1}.$$

Therefore, with probability $1 - o(1)$, each K_i contains at least $\log n$ points of P , and therefore, by Lemma 2.2, this is also true of any convex body in $[0, 1]^d$ of volume ρ . By cutting up any bigger convex body into convex slices of volume ρ or less, we complete the proof of Lemma 2.1. \square

Next, we define a probability measure over the set of queries. Let \mathcal{H}^+ be the set of half-spaces of the form $\tau(q) = \{p \in E^d: \langle p, q \rangle \leq |q|^2\}$, where $q \in E^d \setminus \{O\}$. We denote by \mathcal{H}^- the set of half-spaces obtained by reversing the inequality, i.e., setting $\langle p, q \rangle \geq |q|^2$. We can define a differential structure over \mathcal{H}^+ and \mathcal{H}^- that is invariant under isometries [4], [24]. The differential volume element is uniquely defined (up to scaling) as

$$d\tau = \frac{dx_1 \wedge \cdots \wedge dx_d}{|q|^{d-1}},$$

where $q = (x_1, \dots, x_d)$. Given $\tau \in \mathcal{H}^+$ and $\alpha > 0$, let $\tau[\alpha]$ be the slab obtained by slicing a sliver of width α off the boundary of τ . Specifically, $\tau[\alpha]$ is the set of points p satisfying

$$\left| \langle p, q \rangle - \left(1 - \frac{\alpha}{2|q|}\right)|q|^2 \right| \leq \frac{\alpha}{2}|q|.$$

The key inequality below relates the volume of a convex body with the probability that a random slab $\tau[\alpha]$ encloses it.

Lemma 2.3 [4]. *Given any compact convex body $K \subseteq [0, 1]^d$, we have*

$$\text{vol } K \times \int_{\tau[\alpha] \supseteq K} d\tau < b\alpha^{d+1},$$

where b is a constant that depends only on d .

We close this section with the slicing lemma. This is a useful tool in the spirit of Lemma 2.2. It allows us to break up a big cluster into a small number of pieces that collectively account for the entire action of the cluster near the boundary of a query half-space.

Lemma 2.4. *Given a compact convex body $K \subset E^d$ of unit volume and $0 < \varepsilon < 1/(2d)^{2d}$, there exists a collection of at most $c(1/\varepsilon)^{1-2/(d+1)}$ convex bodies, $K_1, K_2, \dots \subseteq K$, satisfying the following condition: for any half-space $\tau \in \mathcal{H}^-$ with $\text{vol}(K \cap \tau) \geq \varepsilon$, there exists $K_i \subseteq K \cap \tau$ such that*

$$\text{vol } K_i \geq \frac{1}{(6d)^{3d}} \text{vol}(K \cap \tau).$$

The constant c depends only on d .

Given a compact convex body $K \subseteq E^d$, a point $x \in K$, and a real number λ , we call the set

$$M(x, \lambda) = x + \lambda\{(K - x) \cap (x - K)\}$$

a *Macbeath* region [20]. Obviously,

$$M(x, \lambda) \subseteq M(x, \lambda')$$

for all $0 < \lambda \leq \lambda'$ and

$$\text{vol } M(x, \lambda) = \left(\frac{\lambda}{\lambda'}\right)^d \text{vol } M(x, \lambda').$$

We need two additional properties of Macbeath regions. The first one is a variation of Lemma 1 [14] while the second one is derived from Lemma 2 [14].

Lemma 2.5. *If $M(x, \frac{1}{3}) \cap M(y, \frac{1}{3}) \neq \emptyset$ for some $x, y \in K$, then $M(x, \frac{1}{3}) \subseteq M(y, 1)$.*

Proof. As $M(x, \frac{1}{3}) \cap M(y, \frac{1}{3}) \neq \emptyset$, there is a point z in this set that can be written in the form

$$z = x + \frac{1}{3}(k_1 - x) = y + \frac{1}{3}(y - k_2)$$

for some $k_1, k_2 \in K$. Thus,

$$x = \frac{3}{2}y - \frac{1}{4}(k_1 + k_2).$$

We have to show that every point $v \in M(x, \frac{1}{3})$ belongs to the set $M(y, 1) = K \cap (2y - K)$. Obviously,

$$v \in M(x, \frac{1}{3}) \subseteq M(x, 1) \subseteq K.$$

On the other hand, v can be written in the form $v = x + (x - k_3)/5$ for some $k_3 \in K$. It follows that

$$v = \frac{6}{5}x - \frac{1}{5}k_3 = \frac{6}{5}\left(\frac{3}{2}y - \frac{1}{4}k_1 - \frac{1}{4}k_2\right) - \frac{1}{5}k_3 = 2y - \left(\frac{1}{5}y + \frac{3}{10}k_1 + \frac{3}{10}k_2 + \frac{1}{5}k_3\right) \in 2y - K,$$

by convexity of K . □

Lemma 2.6. *Let $K \subset E^d$ be a compact convex body containing the ball $B(O, r)$ of radius r centered at the origin. Let τ be a half-space not containing the origin and assume that the distance between $\partial\tau$ and one of the supporting hyperplanes of K*

parallel to $\partial\tau$ is at most $r/3$. Then

$$K \cap \tau \subseteq M(c, 3d),$$

where c is the center of gravity of the section $K \cap \partial\tau$.

Proof. The statement of Lemma 2 [14] is identical, except that it requires that τ should not cut $B(O, r)$, and that one of the supporting hyperplanes of K parallel to $\partial\tau$ should be within distance $r/2$ from $\partial\tau$. We notice that under our assumptions K contains the ball $B(O, 2r/3)$ and τ does not cut this ball, so that our claim is identical to Lemma 2 [14], with r replaced by $2r/3$. \square

It is well known that, by using a volume-preserving affine transformation of E^d , any compact convex body K can be brought into a form such that it contains a ball of radius r and is contained in a concentric ball of radius R with $R \leq dr$ (see, e.g., [18]). From now on we always assume that $B(O, r) \subseteq K \subseteq B(O, R)$, with $R = dr$ and $\text{vol } K = 1$.

Given any $\varepsilon > 0$, let K_ε denote the set of all points of K not contained in any half-space $\tau \in \mathcal{H}^-$ such that $\text{vol}(K \cap \tau) = \varepsilon$. The set K_ε is usually called a *floating body*. With these conventions, we can easily prove the following statement, which is a strengthening of Theorem 6 from [2].

Lemma 2.7. *For any $\varepsilon < 1/(2d)^{2d}$ there exist pairwise disjoint Macbeath regions $M(x_i, \frac{1}{3}) \subseteq K \setminus K_\varepsilon$, such that*

- (i) $(30d)^{-d}\varepsilon \leq \text{vol } M(x_i, \frac{1}{3}) \leq \varepsilon$ for all i , and
- (ii) any half-space $\tau \in \mathcal{H}^-$ with $\text{vol}(K \cap \tau) \geq \varepsilon$ fully contains at least one of the regions $M(x_i, \frac{1}{3})$.

Proof. The assertion is obviously true for $d = 1$, so let us assume that $d \geq 2$. Assume that x_1, \dots, x_j have already been determined so that (i) holds but (ii) does not. Then there exists a half-space $\tau \in \mathcal{H}^-$ such that $\text{vol}(K \cap \tau) = \varepsilon$, but $M(x_i, \frac{1}{3}) \not\subseteq K \cap \tau$, for all $i \leq j$. Let $h_0 \subseteq \tau$ denote a hyperplane parallel to $\partial\tau$ and supporting K at some point c_0 .

Observe that the distance between $\partial\tau$ and h_0 is at most $r/3$. Otherwise,

$$\text{vol}(K \cap \tau) \geq \text{vol}(\text{conv}\{B(O, r) \cup c_0\}) \cap \tau,$$

and therefore, using ω_d to denote the volume of the unit d -ball, and using the fact that $\omega_d \leq (4/3)^d \omega_{d-1}$,

$$\begin{aligned} \text{vol}(K \cap \tau) &> \left(\frac{r/3}{R}\right)^d \left(\frac{Rr^{d-1}\omega_{d-1}}{d}\right) = \left(\frac{dr}{R}\right)^{2d-1} (R^d \omega_d) \left(\frac{\omega_{d-1}}{3^d d^{2d} \omega_d}\right) \\ &\geq \frac{\omega_{d-1}}{3^d d^{2d} \omega_d} \geq (2d)^{-2d} > \varepsilon, \end{aligned}$$

contradicting our assumption. Therefore, we can apply Lemma 2.6 to derive that $K \cap \tau \subseteq M(c, 3d)$, where c is the center of gravity of $K \cap \partial\tau$.

Letting $y = (c + c_0)/2$, since y lies halfway between $\partial\tau$ and h_0 , we have

$$M(y, \frac{1}{3}) \subseteq M(y, 1) \subseteq K \cap \tau.$$

On the other hand, $M(y, 1) \supseteq M(c, \frac{1}{2}) + (y - c)$. Thus,

$$\begin{aligned} \varepsilon &> \text{vol } M(y, \frac{1}{3}) = \frac{1}{5^d} \text{vol } M(y, 1) \geq \frac{1}{5^d} \text{vol } M(c, \frac{1}{2}) = \frac{1}{(30d)^d} \text{vol } M(c, 3d) \\ &\geq \frac{1}{(30d)^d} \text{vol}(K \cap \tau) = \frac{1}{(30d)^d} \varepsilon. \end{aligned}$$

We claim that $M(x_i, \frac{1}{3}) \cap M(y, \frac{1}{3}) = \emptyset$ for all $i \leq j$. Assuming the opposite, Lemma 2.5 would imply that $M(x_i, \frac{1}{3}) \subseteq M(y, 1) \subseteq K \cap \tau$, which contradicts the choice of τ . So we can set $x_{j+1} = y$ and repeat the whole procedure, if necessary. The algorithm will terminate in finitely many steps. \square

We need the following well-known relation [2], [17]:

Lemma 2.8. *There exists a constant $c_d > 0$ such that, for any compact convex body $K \subset E^d$ of unit volume, and for any $\varepsilon > 0$,*

$$\text{vol}(K \setminus K_\varepsilon) \leq c_d \varepsilon^{2/(d+1)}.$$

In fact, it is also known that $\text{vol}(K \setminus K_\varepsilon)$ attains its maximum for ellipsoids. However, we do not need this stronger assertion. Now, apply Lemma 2.7 with $\varepsilon_j = 2^j \varepsilon$ for $0 \leq j < \log(\varepsilon^{-1} (2d)^{-2d})$. In view of Lemma 2.8, for each j we obtain a collection of at most

$$\frac{\text{vol}(K \setminus K_{\varepsilon_j})}{\varepsilon_j / (30d)^d} \leq \frac{c_d (30d)^d (1/\varepsilon)^{1-2/(d+1)}}{2^{(1-2/(d+1))j}}$$

Macbeath regions, and the union of these collections meets the requirements. In particular, for any $\tau \in \mathcal{H}^-$ such that $\text{vol}(K \cap \tau) \geq \varepsilon$ we can choose j so that

$$\varepsilon_j \leq \text{vol}(K \cap \tau) \leq 2(2d)^{2d} \varepsilon_j,$$

and, hence, we can find a Macbeath region $M(x_i, \frac{1}{3}) \subseteq K \cap \tau$ in the corresponding collection with

$$\text{vol } M(x_i, \frac{1}{3}) \geq \frac{\varepsilon_j}{(30d)^d} \geq \frac{1}{2(2d)^{2d}(30d)^d} \text{vol}(K \cap \tau) > \frac{1}{(6d)^{3d}} \text{vol}(K \cap \tau).$$

This completes the proof of Lemma 2.4. \square

3. The Lower Bound

Let P be a scattered set of n points in $[0, 1]^d$. Assume that C_1, \dots, C_m are the m clusters derived from the storage scheme and that any query can be answered in time at most t . Recall that given a query half-space $\tau \in \mathcal{H}^+$, A_τ is a minimum set of indices such that $P \cap \tau = \bigcup_{i \in A_\tau} C_i$ and that $t \geq \max_\tau |A_\tau|$. Let $\alpha = c_1(t \log n)/n$ be a real parameter, where $c_1 > 0$ is some appropriate constant. Given a query τ , our weighting strategy is meant to filter out any point of τ further from the bounding hyperplane by more than α . To do that, we clip the query by $\tau[\alpha]$. We restrict ourselves to the subset \mathcal{H}_0^+ of half-spaces $\tau = \tau(q) \in \mathcal{H}^+$ such that $q \in (0, +\infty)^d$ and $|P \cap \tau[\alpha]| > 3at \log n$. (Recall that a is the constant implicit in Lemma 2.1.) Let

$$\Phi = \int_{\mathcal{H}_0^+} |P \cap \tau[\alpha]| d\tau.$$

We easily derive a lower bound on Φ by virtue of the fact that P is scattered. On the other hand, the ability of answering any query in time t implies an upper bound on Φ . The desired space–time tradeoff follows from these two bounds.

Since we can assume that t does not exceed, say, $n/\log^2 n$, we easily check that (for n large enough) the measure of the set of $\tau[\alpha]$ ($\tau = \tau(q) \in \mathcal{H}^+$ and $q \in (0, +\infty)^d$) whose intersection with the unit cube has volume exceeding $c_2\alpha$ is at least c_2 , for some small enough constant $c_2 > 0$. Because P is scattered, it follows that if $c_1 > 6a^2/c_2$, then the measure of \mathcal{H}_0^+ is at least c_2 , and hence

$$\Phi > 3ac_2t \log n. \quad (3.1)$$

We now need an upper bound on Φ . Let A_τ^+ be the set of indices $i \in A_\tau$ such that $|C_i \cap \tau[\alpha]| > 2a \log n$. Since $t \geq |A_\tau|$ for any $\tau \in \mathcal{H}_0^+$ and

$$P \cap \tau[\alpha] = \bigcup_{i \in A_\tau} C_i \cap \tau[\alpha],$$

we have

$$|P \cap \tau[\alpha]| \leq \sum_{i \in A_\tau} |C_i \cap \tau[\alpha]| \leq 3 \sum_{i \in A_\tau^+} |C_i \cap \tau[\alpha]|.$$

It follows that

$$\Phi \leq 3 \int_{\mathcal{H}_0^+} \sum_{i \in A_\tau^+} |C_i \cap \tau[\alpha]| d\tau \leq 3 \sum_{1 \leq i \leq m} \int_{\Delta_i} |C_i \cap \tau[\alpha]| d\tau,$$

where the integration domain Δ_i is $\{\tau \in \mathcal{H}_0^+ : |C_i \cap \tau[\alpha]| > 2a \log n\}$. Because P is

scattered and $C_i \subseteq P$, we have

$$\Phi \leq 3 \sum_{1 \leq i \leq m} \int_{\Delta_i} (an \operatorname{vol} B_i + \log n) d\tau \leq 6a \sum_{1 \leq i \leq m} \int_{\Delta_i} n \operatorname{vol} B_i(\tau) d\tau, \quad (3.2)$$

where $B_i(\tau) = \operatorname{conv}(C_i) \cap \tau[\alpha]$ is the portion of the convex hull of C_i within $\tau[\alpha]$.

Let i_0 be the index i that maximizes the rightmost summand, and let

$$\varepsilon = \frac{1}{a(2d)^{2d}} \frac{\log n}{n}.$$

Since P is scattered, we have

$$\frac{\varepsilon}{\operatorname{vol}(\operatorname{conv} C_{i_0})} \leq \frac{\varepsilon}{\operatorname{vol} B_{i_0}(\tau)} < \frac{1}{(2d)^{2d}},$$

and therefore we can apply Lemma 2.4 to $\operatorname{conv} C_{i_0}$. This gives a collection of at most $c(1/\varepsilon)^{1-2/(d+1)}$ convex bodies, $K_1, K_2, \dots \subseteq \operatorname{conv} C_{i_0}$, such that

$$\int_{\Delta_{i_0}} n \operatorname{vol} B_{i_0}(\tau) d\tau \leq (6d)^{3d} \sum_j \int_{\tau[\alpha] \ni K_j} n \operatorname{vol} K_j d\tau.$$

It follows from Lemma 2.3 that

$$\int_{\Delta_{i_0}} n \operatorname{vol} B_{i_0}(\tau) d\tau \leq bc(6d)^{3d} \left(\frac{1}{\varepsilon}\right)^{1-2/(d+1)} n\alpha^{d+1},$$

and from (3.2) that

$$\Phi \leq 6abc(6d)^{3d} \left(\frac{1}{\varepsilon}\right)^{1-2/(d+1)} mn\alpha^{d+1}.$$

Combining this inequality with (3.1) establishes the following.

Theorem 3.1. *Half-space range searching among n points in E^d , using m units of storage, requires query time on the order of*

$$\frac{(n/\log n)^{1-(d-1)/d(d+1)}}{m^{1/d}}.$$

Note that the same lower bound holds for spherical range searching, since a large ball acts as a half-space. We also derive a lower bound for the dynamic on-line case.

Theorem 3.2. *There exists a sequence of n inserts followed by n half-space range searching queries in E^d whose execution requires time on the order of*

$$\frac{n^{2 - (3d+1)/(d+1)^2}}{(\log n)^{(d^2+1)/(d+1)^2}},$$

which is $n^{2-O(1/d)}$.

Proof. Insert the n points that provide the input for Theorem 3.1, then keep asking the hardest query n times. Notice that although any amount of reorganization in the data structure is allowed between queries, no new query may be asked until the answer to the previous one has been provided. The maximum size of the data structure at any stage is at most proportional to the time t required to process the whole sequence. Therefore, the hardest query requires time proportional to

$$\frac{(n/\log n)^{1 - (d-1)/d(d+1)}}{t^{1/d}}.$$

Thus, up to within a constant factor the following inequality holds,

$$t > \frac{n(n/\log n)^{1 - (d-1)/d(d+1)}}{t^{1/d}},$$

from which the theorem follows. □

References

1. Agarwal, P. K., Sharir, M. Applications of a new partitioning scheme, *Discrete Comput. Geom.* **9** (1993), 13–38.
2. Barany, I., Larman, D. Convex bodies, economic cap coverings, random polytopes, *Mathematika* **35** (1988), 274–291.
3. Burkhard, W. A., Fredman, M. L., Kleitman, D. J. Inherent complexity trade-offs for range query problems, *Theoret. Comput. Sci.* **16** (1981), 279–290.
4. Chazelle, B. Lower bounds on the complexity of polytope range searching, *J. Amer. Math. Soc.* **2** (1989), 637–666.
5. Chazelle, B. Lower bounds for orthogonal range searching: I. The reporting case, *J. Assoc. Comput. Mach.* **37** (1990), 200–212.
6. Chazelle, B. Lower bounds for orthogonal range searching: II. The arithmetic model, *J. Assoc. Comput. Mach.* **37** (1990), 439–463.
7. Chazelle, B., Rosenberg, B. The complexity of computing partial sums off-line, *Internat. J. Comput. Geom. Applic.* **1** (1991), 33–45.
8. Chazelle, B., Rosenberg, B. Lower bounds on the complexity of simplex range reporting on a pointer machine, *Proc. 19th ICALP*, LNCS 623, Springer-Verlag, Berlin, July 1992, pp. 439–449.
9. Chazelle, B., Sharir, M., Welzl, E. Quasi-optimal upper bounds for simplex range searching and new zone theorems, *Algorithmica* **8** (1992), 407–429.
10. Chazelle, B., Welzl, E. Quasi-optimal range searching in spaces of finite VC-dimension, *Discrete Comput. Geom.* **4** (1989), 467–489.

11. Clarkson, K., Shor, P. Applications of random sampling to computational geometry, II, *Discrete Comput. Geom.* **4** (1989), 387–421.
12. Cole, R., Yap, C. K. Geometric retrieval problems, *Inform. Control* **63** (1984), 39–57.
13. Edelsbrunner, H., Welzl, E. Halfplanar range search in linear space and $O(n^{0.695})$ query time, *Inform. Process. Lett.* **23** (1986), 289–293.
14. Ewald, G., Larman, D. G., Rogers, C. A. The directions of the line segments and of the r -dimensional balls on the boundary of a convex body in Euclidean space, *Mathematika* **17** (1970), 1–20.
15. Fredman, M. L. A lower bound on the complexity of orthogonal range queries, *J. Assoc. Comput. Mach.* **28** (1981), 696–705.
16. Fredman, M. L. Lower bounds on the complexity of some optimal data structures, *SIAM J. Comput.* **10** (1981), 1–10.
17. Groemer, H. On the mean value of the volume of a random polytope in a convex set, *Arch. Math.* **25** (1974), 86–90.
18. Grotschel, M., Lovasz, L., Schrijver, A. *Geometric Algorithms and Combinatorial Optimization*, Springer-Verlag, Berlin, 1988.
19. Haussler, D., Welzl, E. Epsilon-nets and simplex range queries, *Discrete Comput. Geom.* **2** (1987), 127–151.
20. Macbeath, A. M. A theorem on non-homogeneous lattices, *Ann. of Math. (2)* **56**, (1952), 269–293.
21. Matoušek, J. Reporting points in halfspaces, *Proc. 32nd Ann. Symp. Found. Comput. Sci.*, 1991, pp. 207–215.
22. Matoušek, J. Range searching with efficient hierarchical cuttings, *Proc. 8th Ann. ACM Symp. Comput. Geom.*, 1992, pp. 276–285.
23. Mehlhorn, K. *Data Structures and Algorithms 3: Multidimensional Searching and Computational Geometry*, Springer-Verlag, Berlin, 1984.
24. Santaló, L. A. *Integral Geometry and Geometric Probability*, Encyclopedia of Mathematics and Its Applications, Vol. 1 (Gian-Carlo Rota, ed.), Addison-Wesley, Reading, MA, 1976.
25. Spencer, J. *Ten Lectures on the Probabilistic Method*, CBMS–NSF, SIAM, Philadelphia, PA, 1987.
26. Willard, D. E. Polygon retrieval, *SIAM J. Comput.* **11** (1982), 149–165.
27. Yao, F. F. A 3-space partition and its applications, *Proc. 15th Ann. ACM Symp. Theory Comput.*, 1983, pp. 258–263.
28. Yao, A. C. On the complexity of maintaining partial sums, *SIAM J. Comput.* **14** (1985), 277–288.
29. Yao, A. C., Yao, F. F. A general approach to d -dimensional geometric queries, *Proc. 17th Ann. ACM Symp. Theory Comput.*, 1985, pp. 163–168.

Received September 28, 1992, and in revised form February 3, 1993.