Lower Bounds for Orthogonal Range Searching: II. The Arithmetic Model

BERNARD CHAZELLE

Princeton University, Princeton, New Jersey

Abstract. Lower bounds on the complexity of orthogonal range searching in the static case are established. Specifically, we consider the following *dominance search* problem: Given a collection of n weighted points in d-space and a query point q, compute the cumulative weight of the points dominated (in all coordinates) by q. It is assumed that the weights are chosen in a commutative semigroup and that the query time measures only the number of arithmetic operations needed to compute the answer. It is proved that if m units of storage are available, then the query time is at least proportional to $(\log n/\log(2m/n))^{d-1}$ in both the worst and average cases. This lower bound is provably tight for $m = \Omega(n(\log n)^{d-1+\epsilon})$ and any fixed $\epsilon > 0$. A lower bound of $\Omega(n(\log n/\log\log n)^d)$ on the time required for executing n inserts and queries is also established.

Categories and Subject Descriptors: E.1 **[Data Structures]**; F.2.2 **[Analysis of Algorithms and Problem Complexity]**: Nonnumerical Algorithms and Problems

General Terms: Algorithm, Theory

1. *Introduction*

Whereas searching a linearly ordered set is relatively well understood, the complexity of multidimensional searching is far from being elucidated. The vast amount of literature on this topic witnesses its central location in the study of data structures as well as its relevance to many practical areas (e.g., database, computer graphics). If many ingenious data structures have been discovered, however, only few of them have been given lower bounds matching their performance. An interesting model of computation was proposed by Fredman, along with a powerful technique for proving lower bounds [5, 6]. Unfortunately, deletions (or related operations like updates) play an essential part in Fredman's framework, and results on static problems or on problems allowing only inserts and queries are thereby excluded. This should not come as a total surprise; recent work on dynamization suggests that the coexistence of inserts and deletes often causes a severe increase in complexity [8].

A preliminary version of this work has appeared in *Proceedings of the 27th Annual IEEE Symposium on Foundations of Computer Science*. IEEE, New York, 1986, pp. 87–96.

Part of this research was done while the author was a visiting professor at École Normale Supérieure, Paris, France. This research was supported in part by the National Science Foundation under grant CCR 87-00917.

Author's address: Department of Computer Science, Princeton University, Princeton, NJ 08544.

As regards the static case, one of the most interesting results to date is a lower bound of Yao [14] on the complexity of the following problem, which we call *dominance searching*. Let $\{(p_i, w_i) \mid 1 \leq i \leq n\}$ be a file of n records, where $p_i \in E^d$ and w_i belongs to a commutative semigroup. Given a query point $q \in E^d$, compute $\sum_{p_i \preceq q} w_i$, where $p_i \preceq q$ means that p_i does not dominate q in any coordinate. This problem is fundamental because most *rectangle problems* are in one way or the other equivalent to it [7]. Yao looked at the two-dimensional case and proved that if only m units of storage are available, answering a query requires $\Omega(\log n/\log((m/n)\log n))$ in the worst case.[1] The underlying model of computation—the *arithmetic model*—is very general and is particularly well suited for lower bounds. It regards a data structure as a set of precomputed sums in the semigroup, and charges only for the number of arithmetic operations needed to answer a query. The model makes minimal assumptions about the real costs of an algorithm, which is what makes it so attractive for proving lower bounds.

Another lower bound for the static case was obtained by Vaidya [9], who proved that the worst-case query time t for orthogonal range searching in d-space is at least proportional to $((n/m)\log_t^{d-\theta} n)$, where $\theta = 1$ if $d = 2, 3$, and $\theta = 2$ if $d > 3$. (Orthogonal range searching is a little more general than dominance searching: Queries are of the form $\sum_{q' \preceq p_i \preceq q} w_i = ?$.) Vaidya's result might not be as strong as Yao's (numerically and also because it deals with a more general problem), but it has the advantage to hold in arbitrary dimension.

The main contribution of this paper is to strengthen both sets of lower bounds by presenting a space–time trade-off for dominance searching that is optimal (almost) across the board. Interestingly, the techniques we use are completely different from both Yao's and Vaidya's. Our main result says that the query time is at least proportional to $(\log n/\log(2m/n))^{d-1}$ in the worst case. Actually, we prove the stronger result that under a uniform distribution this bound holds for a random query with probability arbitrarily close to 1; moreover, this is also true if the point-set is random. Consequently, the lower bound holds in the worst case as well as on the average. We can prove that the bound is tight for any $m = \Omega(n(\log n)^{d-1+\epsilon})$ and any $\epsilon > 0$.

We also establish a lower bound of $\Omega(n(\log n/\log\log n)^d)$ on the time required for executing n inserts and queries. A similar result by Fredman [5] says that if deletions are allowed, then $\Omega(n \log^d n)$ is a lower bound. Recent work on dynamization (e.g., [8]) suggests that deletions are often hard to accommodate. Intuitively, what makes a deletion costly is that a single one may invalidate large portions of the data structure (because a semigroup has no inverse operation). Fredman's proof technique rests crucially on that fact and therefore does not generalize to the case of inserts-only. Interestingly, our result says that even without deletions the problem still remains almost as difficult. The lower bound was already established by Yao [14] for the case $d = 1$, but it is new for all $d > 1$.

Most of the proofs in this paper use probabilistic arguments in the sense of [4] and [13]. This departs from previous methods that were (by and large) constructive. For the problems considered in this paper, one advantage of probabilistic methods is to yield at no extra cost additional information on the distribution of inputs that achieve the lower bounds. In several cases, this allows us to strengthen the results by providing bounds for both the worst and average cases.

Section 2 lays down the foundations and recasts the problem in purely combinatorial terms. Section 3 establishes a weaker lower bound meant to illustrate some

[1] All logarithms in this paper are taken to the base 2, unless specified otherwise.

of the ideas used in the main proof, which is given in Section 4. In Section 5, we discuss the optimality of the lower bound. Section 6 treats the dynamic case. Finally, Section 7 concludes with closing remarks and open problems.

2. *The Combinatorial Backdrop*

We set our notation (2.1) and describe the problem (2.2) and the model of computation (2.3). Then we prove a crucial reduction lemma (2.4) to strip the problem of its computational aspect and cast it as a combinatorial problem in discrete geometry.

2.1 SOME NOTATION AND TERMINOLOGY. A *random point* in $[0, 1]^d$ refers to a point $(x_1, \ldots, x_d) \in E^d$, where each x_i is drawn randomly and independently from the uniform distribution in $[0, 1]$. A *random set of points* in $[0, 1]^d$ is obtained by drawing random points in $[0, 1]^d$ independently. We shall use this abbreviated terminology throughout the paper. If $p = (x_1, \ldots, x_d)$ and $q = (y_1, \ldots, y_d)$ are two points in E^d, we say that q *dominates* p, denoted $p \preceq q$, if $x_i \leq y_i$ for each i $(1 \leq i \leq d)$. We define $\hat{p} = \{q \in [0, 1]^d \mid q \preceq p\}$ and $\tilde{p} = \{q \in [0, 1]^d \mid p \preceq q\}$. Finally, let us recall some miscellaneous terminology. We define a *d-range* as the region of E^d formed by the Cartesian product of d closed intervals over the real line. We let λ_d designate the Lebesgue measure in E^d. The cardinality of a finite set X is denoted $|X|$. Let $X_1, \ldots, X_m$ be m collections of closed intervals in $\mathscr{R}$; then $\prod_{1 \leq i \leq m} X_i$ denotes the set

$$\{I_1 \times \cdots \times I_m \mid I_i \in X_i \ (1 \leq i \leq m)\}.$$

Finally, we put $[1 \cdots n] = \{1, 2, \ldots, n\}$ and $\mathscr{N} = \{0, 1, 2, \ldots\}$.

2.2 PRELIMINARIES. Let $(\mathscr{S}, +)$ be a commutative semigroup and let w be a weight function mapping every point of E^d to an element of $\mathscr{S}$. Let $P = \{p_1, \ldots, p_n\} \subseteq E^d$ be a set of n points in E^d. We assume that either w can be computed in constant time or the pairs $\{(p_i, w(p_i)) \mid 1 \leq i \leq n\}$ are given as part of the input. Dominance searching is the problem of answering queries of the form $\sum_{p_i \preceq q} w(p_i) = ?$, given an arbitrary point $q \in E^d$. If no p_i is dominated by q, we expect the answer *null*, which is a special symbol that does not belong to the semigroup. In order to obtain lower bounds which hold on *any* sequential machine, we define a complexity measure which charges only for the arithmetic part of the computation.

Before defining the model formally, let us introduce some of the ideas on a concrete example. We first describe a tentative model. Then we point out its weaknesses and discuss what we can do to fix them. Suppose that the semigroup is $(\mathscr{N}, +)$. If the answer to a query is *null*, no semigroup element is needed for the answer, so we say that the query time is 0. (Note that this discounts the time required to find out that the answer is indeed *null*.) If the answer to a query is, say, 17, and the data structure happens to store 17 somewhere, then we say that the query time is 1, regardless of how long it takes for the algorithm to reach the memory location where the answer is stored. If the database does not store 17 but, say, 10 and 7, the query time will be 2, simply by virtue of the fact that 17 can be computed as 10 + 7. Once again, the time to find these values in memory is not charged. In general, we say that the query time is equal to the smallest number of stored operands needed to form 17, using the semigroup operation as the only arithmetic capability available. Similarly, the size of the data structure is defined to be the number of semigroup elements that it stores. We do not charge for

auxiliary data, pointers, or any other information not directly related to the semigroup.

This model is almost what we want, but not quite. One problem with it is that too many important (nontrivial) semigroups have trivial solutions in that model. If the semigroup is finite, for example, we can encode all the elements in a constant amount of space and have constant-time answers. Also, in that model the semigroup, $(\mathcal{N}, \max)$ always admits of a linear-size solution with a worst-case query time of 1. Since range searching over those semigroups is by no means trivial, we must revise the model to avoid such pitfalls.

Consider the semigroup $(P^*, \cup)$, formed by the power-set of P (minus the empty set), denoted P^*, with set-union as the semigroup operation. None of the problems mentioned above apply in this case. A stored value is a certain nonempty subset of points that can be used only when the output is a superset of it. There is no room here for cheap tricks and nontrivial lower bounds can be expected. To extend this nice situation to other semigroups—beyond those mapping homomorphically onto $(P^*, \cup)$—we modify the model by requiring that each precomputed sum stored in memory should be associated with a certain nonempty subset of P and be equal to the cumulative weight of that subset. This means that if we choose to reassign weights to the points of P, all we have to do is to reevaluate the cumulative weight of each subset under the new assignment, and the same data structure will work just as before. This situation actually reflects what happens most often in practice. That is, most of the data structures for range searching proposed in the literature are specified completely once the points are known, and do not try to take advantage of the weight assignments. (There are very few exceptions.) In this light, a data structure should be regarded not merely as a collection of precomputed values, but as a collection of linear forms. To recreate the situation of our "ideal" semigroup $(P^*, \cup)$, we need not worry so much about the semigroups themselves as about the semigroup of linear forms defined over them. All we need to ensure is that two linear forms with distinct variable sets must be different: Translated in terms of the semigroup, this is a fairly weak property, called *faithfulness*, which is enjoyed by most of the semigroups arising in practice, for example, $(\mathcal{N}, +)$, $(\mathcal{N}, \max)$, $(\{0, 1\}, \text{or})$.

2.3 The Arithmetic Model of Computation. We now set out to specify the computational model precisely as well as its complexity measures. We also state our main lower bound and discuss what happens if we change the model somewhat. To each p_i we assign a distinct variable s_i with values in $\mathcal{S}$. A *generator* $g(s_1, \ldots, s_n)$ is a linear form $\sum_{1 \le i \le n} \alpha_i s_i$, where the α_i's are nonnegative integers (not all 0). Since $\mathcal{S}$ need not be a monoid (i.e., it may not have an identity), we make the notational convention that $\alpha_i = 0$ means that the variable s_i *does not* appear in the linear form. So, for example, $3s_1 + 0s_2 + 2s_3$ stands for $s_1 + s_1 + s_1 + s_3 + s_3$. Similarly, if all the α_i's are 0, the linear form is not defined. In practice (i.e., in all the algorithms described in the literature), $(\alpha_1, \ldots, \alpha_n)$ is a 0/1 vector that characterizes the subset of points associated with the generator (but this does not have to be the case). A *storage scheme* Γ for P of size m is a collection of m generators $\{g_1, \ldots, g_m\}$ satisfying the following property: Given any point q such that $\{p_i \in P \mid p_i \preceq q\} \neq \emptyset$, there exist $K \subseteq \{1, \ldots, m\}$ and a set of labeled integers $\{\beta_k > 0 \mid k \in K\}$ such that the relation

$$\sum_{p_i \preceq q} s_i = \sum_{k \in K} \beta_k g_k(s_1, \ldots, s_n) \tag{1}$$

is an identity. The storage cost is measured by the number of generators used (and not by the size of the explicit description of each of them). The reason for this is obvious. The data structure itself need not store the generators of the storage scheme: It should store their values when each s_i is assigned the value $w(p_i)$. Thus, the answer $\sum_{p_i \preceq q} w(p_i)$ to a query q is obtained by computing the sum $\sum_{k \in K} \beta_k g_k(w(p_1), \ldots, w(p_n))$, which is said to take time $|K|$.

As was hinted to earlier, a data structure is more than a mere collection of semigroup elements. It also specifies, for each memory cell, which linear form is used to compute its contents. A change in weight assignment can be simply effected by reevaluating the linear forms of the storage scheme. The query algorithm remains just the same. Now, to avoid a number of difficulties, we wish to assume that generators are identical if and only if they are formally identical. Actually, we require something a little weaker, called *faithfulness*. We say that the semigroup $(\mathscr{S}, +)$ is *faithful* if for each $n > 0$, $\varnothing \subset T_1$, $T_2 \subseteq [1 \cdots n]$, $T_1 \neq T_2$, and every sequence of integers $\alpha_i, \beta_j > 0$ $(i \in T_1, j \in T_2)$, the equation

$$\sum_{i \in T_1} \alpha_i s_i = \sum_{j \in T_2} \beta_j s_j$$

is not an identity (i.e., cannot be satisfied for all assignments of the variables $s_1, \ldots, s_n$). Note that this definition does not prohibit idempotence $(x = 2x)$. For example, $(\mathscr{N}, +)$, $(\mathscr{N}, \max)$, and $(\{0, 1\}, \textit{or})$ are faithful, but $(\{0\}, \textit{or})$ and $(\{0, 1\}, \textit{exclusive or})$ are not. In summary, a storage scheme must work for *any* weight assignment (this is what we call the *storage scheme requirement*), but it can take advantage of any property that P may enjoy. Because of the bijection between the variables s_i and the points of P, we can (abusively) regard a generator $\sum_{1 \le i \le n} \alpha_i s_i$ as a certain subset of P: $\{p_i \mid \alpha_i > 0\}$. Given faithfulness, the meaning of (1) is that any set of the form $\{p_i \mid p_i \preceq q\}$ can be written as a union of generators. Note that this union need not be disjoint.

Next, we define the complexity of a storage scheme Γ. Given a point $q \in E^d$, let K be the smallest set such that (1) is true: we define $t(P, \Gamma, q) = |K|$. As always, we are interested in complexity measures as functions of the number n of input points and the size m of the data structure. For this purpose we define the function $t(n, m)$, where

$$t(n, m) = \max_{|P|=n} \min_{|\Gamma|=m} \max_{q} t(P, \Gamma, q).$$

Assuming the probability distribution discussed in (2.1), we define the average-case time complexity

$$\bar{t}(n, m) = E_{|P|=n} \min_{|\Gamma|=m} E_q t(P, \Gamma, q).$$

We shall now state the result whose proof will occupy us for the next four sections.

THEOREM 2.1. *Let $\mathscr{S}$ be a faithful commutative semigroup; let d be any positive integer and ϵ any fixed real $(0 < \epsilon < 1)$. There exists a constant $c > 0$ (dependent on ϵ) such that the following is true. Let P be a random set of n points in $[0, 1]^d$ and let Γ be any storage scheme for P of size m. If p is a random point in $[0, 1]^d$, then with probability greater than $1 - \epsilon$ the time complexity of the dominance search problem satisfies*

$$t(P, \Gamma, p) \geq c\left(\frac{\log n}{\log(2m/n)}\right)^{d-1}.$$

As a corollary, the worst-case and average-case times satisfy

$$t(n, m) \geq \bar{t}(n, m) = \Omega\left(\left(\frac{\log n}{\log(2m/n)}\right)^{d-1}\right).$$

Remark. Intuitively, we have the chronological sequence: Pick P, then set Γ, and finally choose p. Rigorously, the statement is to be understood as follows: Given any function mapping each P into a storage scheme $\Gamma(P)$, the stated lower bound on $t(P, \Gamma(P), p)$ holds with probability $> 1 - \epsilon$, if (P, p) is a random point in $[0, 1]^{d(n+1)}$. To say "let Γ be any storage scheme ..." in Theorem 2.1 is a shorthand for saying that the previous statement is true for an *arbitrary* choice of the function Γ.

What would happen to Theorem 2.1 if we were to drop the storage scheme requirement. Let us change the model of computation (just for now) and relax the requirement that the preprocessing should work for all weight assignments. Instead, let us think of a data structure as a collection of precomputed sums with no special assumption other than the ability to express every non-*null* answer as a sum of stored values. From our previous discussion, it appears that Theorem 2.1 will hold for any semigroup that is rich enough to simulate all nonempty subsets of an n-element set. Certain faithful semigroups, such as $(\mathcal{N}, \max)$, were rich enough to obey the lower bound in the previous model but are now in jeopardy. This is not the case of $(\mathcal{N}, \times)$, however. Let β_i be the ith largest prime: assign the weight β_i to each p_i. Obviously, we can always assume that the data structure does not store any integer which is a multiple of a prime exceeding β_n. Again, let $(P^*, \cup)$ be the semigroup formed by the set of all nonempty subsets of the point-set P under the set-union operation. The function that maps any integer to its set of prime divisors carries the semigroup generated by $\beta_1, \ldots, \beta_n$ homomorphically onto the semigroup $(P^*, \cup)$. This proves that $(\mathcal{N}, \times)$ is rich enough to obey the lower bound of Theorem 2.1, even though the storage scheme requirement has been lifted.

2.4 The Dominance Lemma. The title of this paper refers to orthogonal range searching. The fact that we restrict ourselves to dominance searching may have appeared to the reader as a courageous decision, since proving lower bounds for a subproblem can only be more difficult. As it turns out, this will allow us to clear out the landscape and simplify matters a great deal. The reason is that in this way the points associated with a generator can always be chosen to be of the form $P \cap \hat{p}$, for some $p \in [0, 1]^d$. The net result is that both queries and generators can be interpreted as points in d-space. This remarkable property will play a crucial role in the following:

Let M be a finite set of points in $[0, 1]^d$. We say that M is a *P-cover* if, for each point $p \in [0, 1]^d$, there exists a subset Q of $M \cap \hat{p}$, such that

$$P \cap \hat{p} \subseteq \bigcup_{q \in Q} \hat{q}. \tag{2}$$

We define $c(P, M, p) = |Q|$, where Q is the smallest set such that (2) is true. If M is not a P-cover, we have $c(P, M, p) = +\infty$. Our next result states that each generator can be normalized by considering the smallest $\hat{p}$ that contains all the points associated with it.

Lemma 2.2. *Let $\mathcal{S}$ be a faithful commutative semigroup and let P be a finite subset of $[0, 1]^d$. Given any storage scheme Γ for P of size m, there exists a P-cover M of size m such that, for each point $p \in [0, 1]^d$, $c(P, M, p) \leq t(P, \Gamma, p)$.*

PROOF. Let $P = \{p_1, \ldots, p_n\}$ and let $\Gamma = \{g_1, \ldots, g_m\}$ be a storage scheme for P. If

$$g_k(s_1, \ldots, s_n) = \sum_{1 \le i \le n} \alpha_{k,i} s_i,$$

let $C_k = \{p_i \mid \alpha_{k,i} > 0\}$. We define the set $M = \{q_1, \ldots, q_m\}$, where each q_k is the corner of the smallest octant containing the points of C_k. Formally, q_k is the unique point such that

$$\hat{q}_k = \bigcap_{C_k \subseteq \hat{p}} \hat{p}.$$

We can prove that M is a P-cover of complexity no greater than Γ. Let p be a point of $[0, 1]^d$. From (1), we have

$$\sum_{p_i \preceq p} s_i = \sum_{k \in K} \beta_k g_k(s_1, \ldots, s_n),$$

for some $K \subseteq \{1, \ldots, m\}$. Using the associativity and commutativity of $\mathscr{S}$ we have

$$\sum_{p_i \preceq p} s_i = \sum_{p_i \in C^+} \gamma_i s_i,$$

where $C^+ = \bigcup_{k \in K} C_k$ and $\gamma_i > 0$. Let $Q = \{q_k \mid k \in K\}$. By faithfulness, we have $C^+ = P \cap \hat{p}$, which implies that $P \cap \hat{p} \subseteq \bigcup_{k \in K} \hat{q}_k$. Since each C_k is a subset of $\hat{p}$, each q_k lies in $\hat{p}$. The proof is complete. Note that the maxima of Q do the covering job all by themselves. □

3. *A Weaker Result to Illustrate the Main Ideas*

What was originally a fairly intricate problem, the space/time complexity of orthogonal range searching, has now become, courtesy of Lemma 2.2, a rather simple problem to state. Let us reword it as a game played between two protagonists. Alice (the lower-bound prover) places n green points in $[0, 1]^d$. Then, Bob (Nature) places m red points to *shield* the green points. What *shield* means is that for any choice of p in $[0, 1]^d$, the set of green points dominated by p is itself *globally dominated* by a subset of at most t red points; themselves, dominated by p. Alice wants to prove that if m is small then t must be large. If m is at least n^2, Bob can always form a grid by drawing vertical and horizontal lines through each green point and then place one red point per grid cell. This will make $t = 1$. On the other hand, if $m = n$, Bob has no choice but to place his troops right on the green points. But clever Alice will have arranged her points on the line $y = 1 - x$. Thus, we get $t = n$ in the worst case.

These extreme situations are fairly trivial. Let us now consider intermediate scenarios. Alice will try to disperse her green points as much as possible in order to limit the power of any red point. To do this, she will just throw her points at random. Obviously, Bob must place one red point on top of each green point. This is actually necessary and sufficient to ensure that t remains finite. With his remaining $m - n$ red points, he will try to be clever enough to get a small t. Bob knows that Alice has thrown the green points at random. For a given p, it makes sense to attribute a *quality rating* to the red points inside $\hat{p}$. Intuitively, the closer a red point is to p, the greater is its contribution to the global domination of $P \cap \hat{p}$. Of course, we are not referring here to the Euclidean metric but to the range-distance discussed in Part I [3]. Recall that the *range-distance* between two

points $(x_1, \ldots, x_d)$ and $(y_1, \ldots, y_d)$ in E^d is defined as $\prod_{1 \le k \le d} |x_k - y_k|$: it is the Lebesgue measure of the smallest d-range enclosing the two points.

The proofs will get a bit technical, so we choose to begin with a toned-down version of the proof technique in two dimensions. This will give us a chance to introduce the main ideas in a simpler context. We shall prove that $t(m, n) = \Omega(\log n / \log(m/n) \log n))$, in the case $d = 2$. This result is a little weak because, as we shall see in the next section, the second $\log n$ term need not be there. We begin with a few definitions. We say that a set of n points in $[0, 1]^2$ is *weakly uniform* if the points are in general position and any 2-range in $[0, 1]^2$ of measure $64(\log n)/n$ contains at least one point of the set. Our next result says that to be weakly uniform is nothing out of the ordinary.

LEMMA 3.1. *A random set of n points is weakly uniform with probability greater than* $1 - 1/n^4$, *for any n large enough.*

PROOF. We shall always assume that n is larger than any appropriate constant, whatever the case might be. We define a set of cells $\mathscr{T} = \{r \in \mathscr{S}^2 \mid n\lambda_2(r) \ge 4 \log n\}$, where

$$\mathscr{S} = \left\{ \left[\frac{i}{2^k}, \frac{i+1}{2^k} \right] \middle| 0 \le k \le \lceil \log n \rceil \text{ and } 0 \le i < 2^k \right\}.$$

Let $R = [x_1, x_2] \times [y_1, y_2] \subseteq [0, 1]^2$ be a 2-range of measure $64(\log n)/n$, and let

$$k = \left\lceil \log \frac{1}{x_2 - x_1} \right\rceil + 1$$

and $i = \lceil x_1 2^k \rceil$. We have

$$2^k(x_2 - x_1) = 2^{1 + \lceil \log 1/(x_2 - x_1) \rceil}(x_2 - x_1) \ge 2;$$

therefore, $2^k x_2 \ge 2 + 2^k x_1 > i + 1$, which implies that $[i/2^k, (i+1)/2^k] \subseteq [x_1, x_2]$. We easily verify that $0 \le k \le \lceil \log n \rceil$ and $0 \le i < 2^k$, therefore, $r_1 = [i/2^k, (i+1)/2^k]$ lies in $[x_1, x_2]$ and is a member of $\mathscr{S}$. Similarly, we derive the existence of $r_2 \in \mathscr{S}$ such that $r_2 \subseteq [y_1, y_2]$. We have

$$\lambda_1(r_1) = \frac{1}{2^k} \ge \frac{x_2 - x_1}{4},$$

and similarly, $\lambda_1(r_2) \ge (y_2 - y_1)/4$. We can then conclude to the existence of a 2-range $r \subseteq R$, with $r \in \mathscr{S}^2$ and $\lambda_2(r) \ge \lambda_2(R)/16 = 4(\log n)/n$, and hence $r \in \mathscr{T}$. To summarize, any 2-range in $[0, 1]^2$ of measure $64(\log n)/n$ contains a 2-range of $\mathscr{T}$. To complete the proof, we must show that if P is a random set of n points in $[0, 1]^2$, then with probability $> 1 - 1/n^4$ every 2-range of $\mathscr{T}$ contains a point of P. Let $n(k, l)$ be the number of elements of $\mathscr{T}$ of the form $[i/2^k, (i+1)/2^k] \times [j/2^l, (j+1)/2^l]$: we have $n(k, l) = 0$ if $n < 2^{k+l+2} \log n$, and $n(k, l) \le 2^{k+l}$ in general. Therefore,

$$|\mathscr{T}| \le \sum_{k, l \mid 2^{k+l} \log n \le n/4} n(k, l) \le \sum_{i \le \log n - \log\log n - 2} 2^i \log n \le n.$$

Now, the probability that no 2-range of $\mathscr{T}$ is empty is at least

$$1 - |\mathscr{T}|(1 - 4(\log n)/n)^n > 1 - |\mathscr{T}| e^{-4 \log n},$$

which is at least $1 - n/e^{4 \log n} \ge 1 - 1/n^4$. □

Let P be a weakly uniform set of n points in $[0, 1]^2$ and let M be a P-cover of size $m \leq n^2$. For clarity, we introduce some notation: $a = 1/\sqrt{16m\log n}$, $b = \frac{1}{16}$, $c = 3000m(\log n)^2/n$, and $\delta = \lfloor \frac{1}{5}(\log n)/\log((m/n)\log n) \rfloor$. Let

$$\mathscr{B} = ([0, b] \times [0, a]) \cap \{(x, y) \mid xy \leq a^2\}.$$

As usual, we assume that n is large enough so that, in particular, we have $a < b$. Let x and x' be two reals ($a \leq x < x' \leq b$) and let $\chi_1 = (x, a^2/x')$ and $\chi_2 = (x', a^2/x)$: χ_1 and χ_2 are two points on each side of the hyperbola delimiting $\mathscr{B}$. We define $b(x, x') = \tilde{\chi}_1 \cap \hat{\chi}_2$. Any 2-range of the form $b(x, x')$ is called a *box* (Figure 1). A box is called *valid* if its measure is equal to $64(\log n)/n$. A set $b_1, \ldots, b_k$ forms a *chain* of boxes if each b_i is a box and the intersection of any two is empty ($b_i \cap b_j = \emptyset$, if $i \neq j$).

Here is what we are trying to do: We want to grab the gun-like device $\mathscr{B}$ at the origin, rotate it by 180 degrees and translate it some place. The corner of the device specifies a query $\hat{p}$, as illustrated in Figure 2. Because of the weak uniformity of P, each box of the chain will contain a green point. We then argue that, no matter how Bob arranges his red points, it is always possible to find a placement of the device that avoids all the red points. In these conditions, no red point can be used to dominate more than one box of the chain. The length of the chain will then give us a lower bound on the query time. Intuitively, the choice of a hyperbola as the delimiting curve of the device is a compromise between two desires: A flat curve would allow us to place more boxes but would made it impossible to avoid the red points. Conversely, a very curvy shape would make it easy to avoid the red stuff but would make room for only a short chain of boxes. Of course, the fact that a sphere in the range-distance metric has a hyperbolic shape is not alien to our choice. We now turn these ideas into reality, beginning with a few technical lemmas.

LEMMA 3.2. *For any n large enough, there exists a chain of δ valid boxes.*

PROOF. Let $k = \lfloor (\log(b/a))/\log c \rfloor$ and let $b_i = b(ac^i, ac^{i+1})$, for each i such that $0 \leq i < k$. Since $m \geq n$, we have $c > 1$ and $ac^i \geq a$, for n large enough. The inequality $ac^k \leq b$ implies that each b_i is a box. We have

$$\lambda_2(b_i) = \frac{a^2(c-1)^2}{c} > \frac{a^2 c}{2} > \frac{64\log n}{n};$$

therefore, b_i contains a valid box b_i' in its interior. The set $\{b_0', \ldots, b_{k-1}'\}$ forms a chain of valid boxes. There are k of them, with

$$k = \left\lfloor \frac{\log((16m\log n)^{1/2}/16)}{\log(3000m\log^2 n/n)} \right\rfloor \geq \left\lfloor \frac{\frac{1}{2}\log n + \frac{1}{2}\log\log n - 2}{2\log((m/n)\log n) + 12} \right\rfloor > \delta,$$

for n large enough. □

Let $p = (p_x, p_y) \in [0, 1]^2$ and let $C(p)$ denote the device we were talking about earlier:

$$C(p) = \{(p_x - x, p_y - y) \mid (x, y) \in \mathscr{B}\}.$$

We say that a point p is *clear* if $C(p) \subseteq [0, 1]^2$ and $C(p) \cap M = \emptyset$. With each point p the region $C(p)$ associates a chain of boxes that lie between two pieces of hyperbola ($xy = a^2c$ and $xy = a^2/c$, up to rotation). Each box is valid and therefore

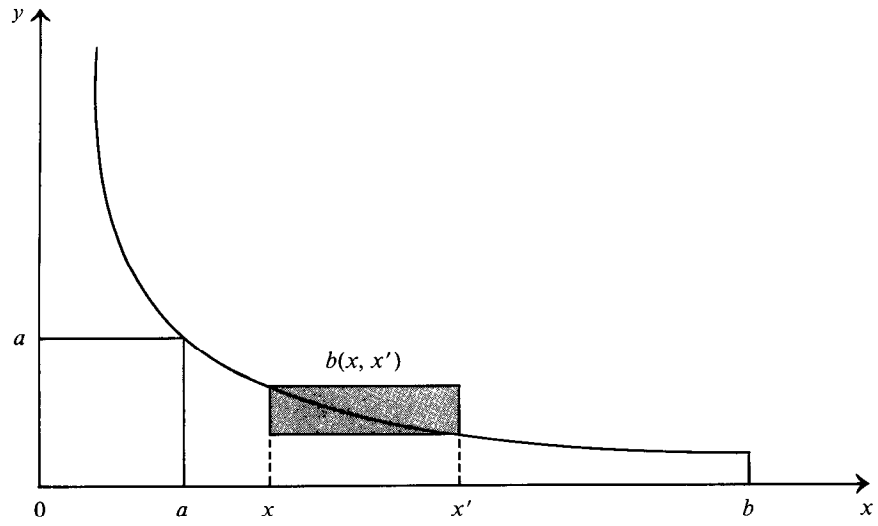

FIGURE 1

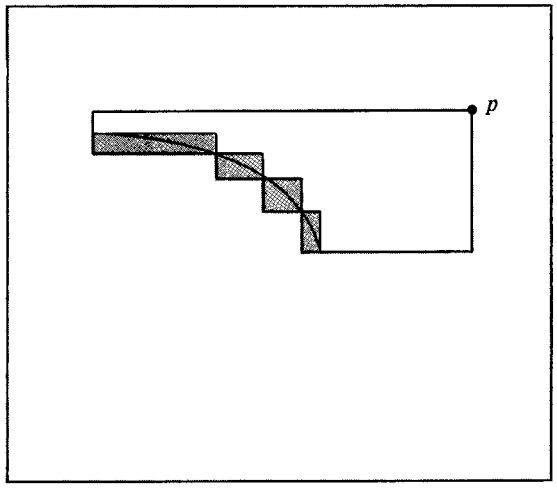

FIGURE 2

contains green points (weak uniformity). If p is clear, it will take at least one red point per box to ensure the global domination of the green points (Figure 2).

LEMMA 3.3. *A random point of* $[0, 1]^2$ *is clear with probability greater than* $\frac{3}{4}$, *for any n large enough.*

PROOF. Let μ be the probability that p is not clear. The condition $C(p) \subseteq [0, 1]^2$ contributes $1 - (1 - a)(1 - b)$ to μ. As to the condition $C(p) \cap M = \emptyset$, each point of M contributes at most the measure of $\mathscr{B}$, that is,

$$a^2 + \int_a^b \frac{a^2}{x}\, dx = a^2 \left(1 + \ln \frac{b}{a}\right);$$

therefore,

$$\mu \leq 1 - (1 - a)(1 - b) + a^2 m \left(1 + \ln \frac{b}{a}\right) \leq a + b + a^2 m \left(1 + \log \frac{b}{a}\right),$$

and $\mu \leq \frac{1}{8} + (\log m)/(32 \log n)$, for n large enough. Since we have assumed that $m \leq n^2$, we have $\mu \leq \frac{3}{16}$. □

If P is weakly uniform and p is clear, then each of the valid boxes of $C(p)$ provided by Lemma 3.2 contains at least one point of P (note that these boxes lie entirely in $[0, 1]^2$). Since the boxes are pairwise disjoint and $C(p) \cap M = \emptyset$, no point of M in $\hat{p}$ can dominate two points of P that lie in distinct boxes. It follows that $c(P, M, p) \geq \delta$. The inequality remains true if $m > n^2$; therefore,

$$t(n, m) = \Omega\left(\frac{\log n}{\log((m/n)\log n)}\right)$$

in two dimensions.

The relative weakness of this result is owed to two factors. As its name suggests, the weak uniformity criterion is too weak, indeed. We would like to strengthen it by requiring that *any* 2-range of measure inversely proportional to the density of P should intersect P. Unfortunately, this does not seem so easy to achieve in higher dimensions. Therefore, we replace *any* by *most*. Also, it appears that the *clearness* condition is a tad too strong. We weaken it by requiring that $C(p)$ should not contain too much red stuff (whatever that means). There now remains for us the arduous task of making these modifications precise and proving their utility.

4. *The Proof of Theorem* 2.1

We are now ready to prove our general lower bound. We assume in the following that $d > 1$. Let ϵ be an arbitrary fixed real $(0 < \epsilon < 1)$. As we did in the previous section, we introduce a few parameters to facilitate our discussion.

$$\begin{cases} \alpha = \dfrac{1}{n^{1/d}} \\ \beta = \dfrac{\epsilon}{5 \log(m/n)} \\ h = \dfrac{5}{(\epsilon n)} \end{cases}$$

Until specified otherwise, it will be assumed that the integers n and m satisfy the inequalities

$$0 < d^2 n < m < n^{1+\epsilon^2/5^{d+5}}. \tag{3}$$

These are technical restrictions to make our calculations easier. We relax them later. We continue with a string of definitions that, as we will try hard to show, are not nearly as tedious as they look. Let $p = (x_1, \ldots, x_d) \in [0, 1]^d$ and $\tilde{p} = (x_1, \ldots, x_{d-1})$. For each k $(0 < k < d)$ and $i \geq 0$, we define

$$J_{k,i} = [x_k - u_{i+1}, x_k - u_i],$$

where $u_0 = 0$ and for $i > 0$, $u_i = u_{i-1} + \alpha 2^{(i-1)/\beta}$. (Note that $u_i = \alpha((2^{i/\beta} - 1)/(2^{1/\beta} - 1))$.) We define the *logarithmic lattice*

$$\mathscr{L}(\bar{p}) = \prod_{0<k<d} \left(\bigcup_{i \geq 0} \{J_{k,i}\} \right).$$

This lattice is called logarithmic because a regular lattice can be derived from it by a logarithmic mapping. The lattice consists of $(d-1)$-ranges $r(\bar{p}, j) = \prod_{0<k<d} J_{k,i_k}$, where $j = (i_1, \ldots, i_{d-1}) \in \mathscr{N}^{d-1}$. The point $\bar{p}$ specifies the origin of the lattice in E^{d-1}; $i_1, \ldots, i_{d-1}$ are the coordinates of the lattice point j in its local system of reference. Let $z = h/\lambda_{d-1}(r(\bar{p}, j))$ and

$$a = (x_1 - u_{i_1+1}, \ldots, x_{d-1} - u_{i_{d-1}+1}, x_d - z).$$

We define

$$\begin{cases} v(p, j) = r(\bar{p}, j) \times [x_d - z, x_d], \\ v^+(p, j) = r(\bar{p}, j) \times \left[x_d - z, x_d - \dfrac{z}{2^{1/\beta}} \right], \\ w(p, j) = (\bar{a} \cap \hat{p}) \backslash v^+(p, j). \end{cases}$$

Note that since $2^{1/\beta} = (m/n)^{5/\epsilon} > d^{10/\epsilon} > 1$ (from (3)), the interval $[x_d - z, x_d - z/2^{1/\beta}]$, and hence $v^+(p, j)$, are well defined.

Figures 3 and 4 illustrate these notions in the cases $d = 2, 3$. Figure 3 shows clearly what $v(p, j)$ and $w(p, j)$ are. The lattice is one-dimensional, extending horizontally from p to the left. The corner a lies on a hyperbolic curve. In Figure 4, the logarithmic lattice is two-dimensional: (i) $v(p, j)$ is the box between the two shaded rectangles with corners a and U; the volume of the box is invariant (equal to h regardless of the position of j in the lattice); (ii) $w(p, j)$ is the box between the corners p and a minus the box between the two shaded rectangles with corners a and T.

Our strategy now is to find a placement of p that maximizes the number of v-boxes that contain green points and at the same time maximizes the number of w-boxes that are free of red points. In the previous section, we required that *all* v- and w-boxes satisfy these properties. Now we simply require that *many* of them should. We define the notion of a ν-exposed point p (one that has more than ν v-boxes with green points inside) and that of a ν-isolated point p (one that has more than ν w-boxes free of red points). We estimate the probability that a random throw of n green points followed by a random pick of p gives us a point that is both ν-exposed and ν-isolated, a property that we call ν-hyperbolic. Finally, we argue that if ν is big enough, a point p that is ν-hyperbolic contains many v-boxes, say ν' of them, which contain green stuff and whose corresponding w-boxes are free of red points. Obviously, what we have in mind is to claim that ν' is indeed a lower bound on the query time. These are the basic ideas of the proof. The rest is just a lot of technical approximations necessary for the probabilistic analysis.

We introduce two collections of sets (the v-boxes and w-boxes of the previous discussion):

$$\mathscr{V}(p) = \{v(p, j) \subseteq [0, 1]^d \mid j \in \mathscr{N}^{d-1}\}$$

and

$$\mathscr{W}(p) = \{w(p, j) \mid v(p, j) \in \mathscr{V}(p)\}.$$

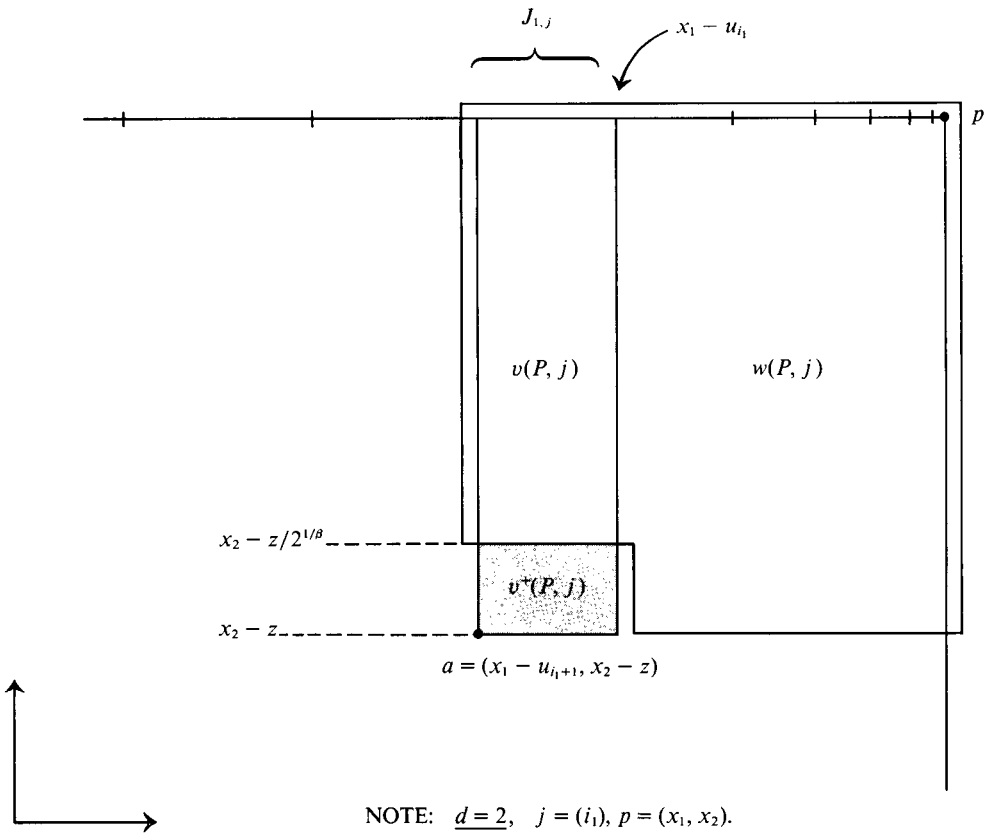

FIGURE 3

Let $P = \{p_1, \ldots, p_n\}$ and $M = \{q_1, \ldots, q_m\}$ be two sets of points in $[0, 1]^d$; P gives the green points and M the red points. Given P, M, and a placement p of the lattice, we wish to count how many v-boxes contain green points and how many w-boxes are free of red points. For this purpose, we introduce the functions $\pi(P, p)$ and $\mu(M, p)$, defined for each $p \in [0, 1]^d$:

$$\pi(P, p) = |\{r \in \mathscr{V}(p) \mid P \cap r \neq \emptyset\}|$$

and

$$\mu(M, p) = |\{r \in \mathscr{W}(p) \mid M \cap r = \emptyset\}|.$$

Let ν be a positive real. We say that $p \in [0, 1]^d$ is *ν-exposed* if $\pi(P, p) > \nu$, and *ν-isolated* if $\mu(M, p) > \nu$. If p is both ν-exposed and ν-isolated, then it is called *ν-hyperbolic*. Note that P and M are understood. These definitions find their justification in the following lemma.

LEMMA 4.1. *Let P be a finite set of points in $[0, 1]^d$ and M be a P-cover. Let $p \in [0, 1]^d$ and ν be a real > 0. If p is ν-hyperbolic, then $c(P, M, p) > 2\nu - |\mathscr{V}(p)|$.*

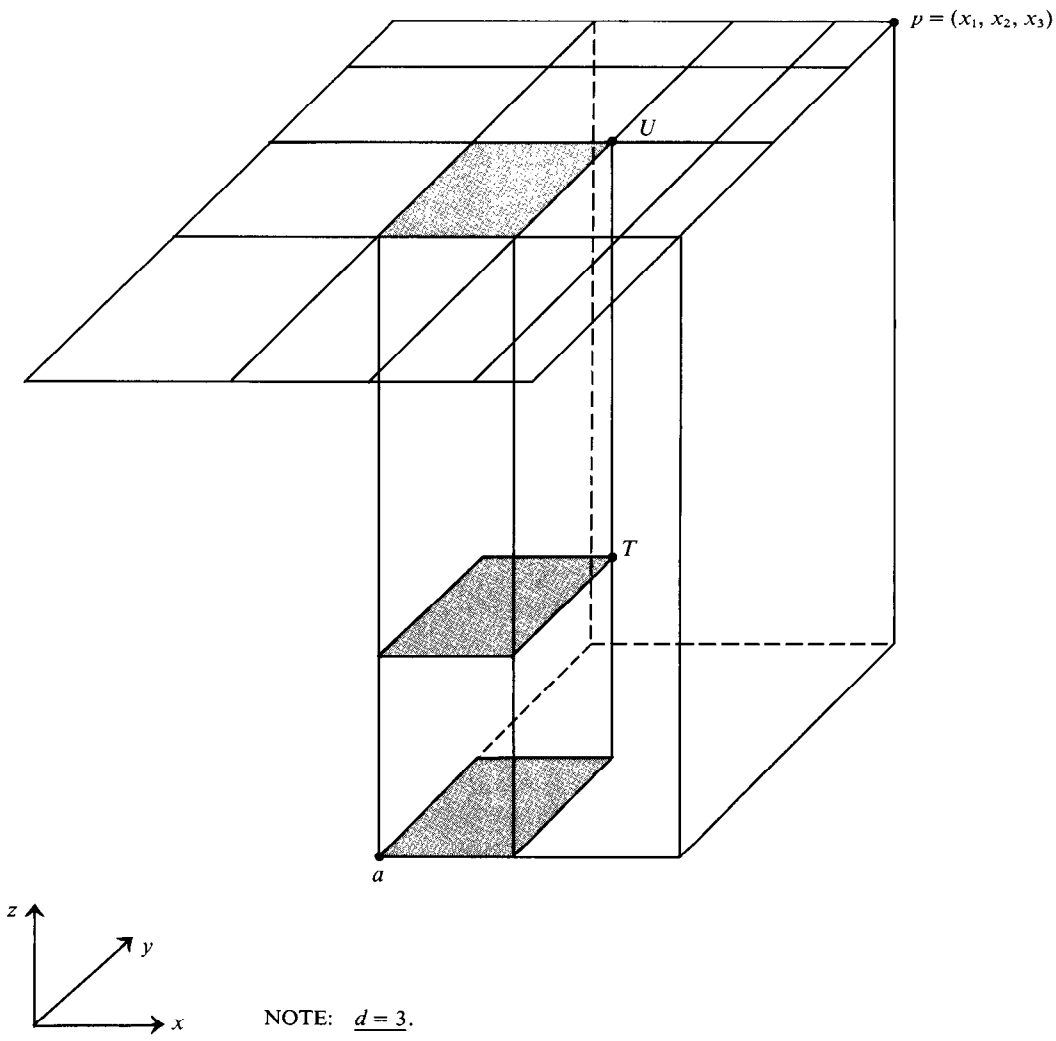

FIGURE 4

PROOF. Let

$$L_1 = \{j \in \mathcal{N}^{d-1} \mid v(p, j) \in \mathcal{V}(p) \text{ and } v(p, j) \cap P \neq \emptyset\}$$

and

$$L_2 = \{j \in \mathcal{N}^{d-1} \mid w(p, j) \in \mathcal{W}(p) \text{ and } w(p, j) \cap M = \emptyset\},$$

and let $L = L_1 \cap L_2$. If $j \in L$, $v^+(p, j)$ contains a point of P since $P \subseteq M$ and $w(p, j) \cap M = \emptyset$. This point is not shared by any other $v^+(p, j')$ $(j' \neq j)$. But since $w(p, j) \cap M = \emptyset$, the only points of $M \cap \hat{p}$ that can dominate this point must also lie in $v^+(p, j)$. This proves that if we have $Q \subseteq M \cap \hat{p}$ and $P \cap \hat{p} \subseteq \bigcup_{q \in Q} \hat{q}$, then $|Q| \geq |L|$. But we have

$$|L| = \pi(P, p) + \mu(M, p) - |L_1 \cup L_2| > 2\nu - |\mathcal{V}(p)|;$$

therefore, the proof is complete. □

Our work is cut out for us. We need to show that, even when ν is large, a random point p is ν-hyperbolic with high probability. Luckily, first-moment methods are powerful enough for this purpose. In other words, the arguments come down to estimating the expectation Φ_1 (resp., Φ_2) of the number of v-boxes that contain green points (resp., w-boxes free of red points). The only difficulty is technical: It comes from the fact that p is not free to be placed anywhere in the unit cube, since a substantial number of v- and w-boxes must also lie in the cube.

4.1 Measuring the Set of Exposed Points. Let $j = (i_1, \ldots, i_{d-1}) \in \mathcal{N}^{d-1}$. We define a characteristic function $f_j(P, p)$ as follows:

$$f_j(P, p) = \begin{cases} 1, & \text{if } v(p, j) \in \mathscr{V}(p) \text{ and } v(p, j) \cap P \neq \emptyset; \\ 0, & \text{otherwise.} \end{cases}$$

Put

$$t_j = \left(u_{i_1+1}, \ldots, u_{i_{d-1}+1}, \frac{h}{\alpha^{d-1} 2^{(i_1+\cdots+i_{d-1})/\beta}}\right),$$

and

$$\Phi_1 = \int_{[0,1]^{dn}} \int_{[0,1]^d} \pi(P, p)\, dp\, dP.$$

Note that we treat P as a point in $[0, 1]^{dn}$. Observing that $v(p, j) \in \mathscr{V}(p)$ if and only if $p \in \tilde{t}_j$, we can write

$$\begin{aligned} \Phi_1 &= \int_{[0,1]^{dn}} \int_{[0,1]^d} \sum_{j \in \mathcal{N}^{d-1}} f_j(P, p)\, dp\, dP \\ &= \sum_{j \in \mathcal{N}^{d-1}} \int_{[0,1]^d} \int_{[0,1]^{dn}} f_j(P, p)\, dP\, dp \\ &= \sum_{j \in \mathcal{N}^{d-1}} \int_{\tilde{t}_j} (1 - (1 - h)^n)\, dp; \end{aligned}$$

therefore,

$$\Phi_1 = (1 - (1 - h)^n) \sum_{j \in \mathcal{N}^{d-1}} \lambda_d(\tilde{t}_j). \tag{4}$$

The derivations above are valid only if $h < 1$, which is true for n large enough. We continue with a technical result. Let $\alpha' = \alpha^{1/\sqrt{\log(1/\alpha)}}$ and $\alpha_d = (\alpha', \ldots, \alpha') \in E^d$ and let $\Delta = \{j \in \mathcal{N}^{d-1} \mid t_j \in \hat{\alpha}_d\}$. As long as we confine the point p to the region $\tilde{\alpha}_d$, we can use all the v-boxes $v(p, j)$ $(j \in \Delta)$, since they lie entirely in the unit cube. Although we can use more v-boxes when p is close to $(1, 1, \ldots, 1)$, it is convenient to use the same number for all the points p. Since α_d converges towards the origin O as n goes to infinity, confining p to $\tilde{\alpha}_d$ is not a big imposition. Of course, one might be worried that this does not leave us much room to place v-boxes. The best thing about the logarithmic lattice is that this is not the case. To get one more v-box requires an exponentially larger chunk of the unit cube, so what we obtain from Δ is actually quite large. Since

$$u_i = \alpha \frac{2^{i/\beta} - 1}{2^{1/\beta} - 1},$$

even under the best of circumstances no point p can claim more than roughly

$(\beta \log(1/\alpha))^{d-1}$ v-boxes and w-boxes. Our next result shows that restricting ourselves to the lattice coordinates of Δ yields just about the same number of boxes.

LEMMA 4.2. *For any n sufficiently large, we have*

$$|\Delta| > \left(1 - \frac{d}{(\log 1/\alpha)^{1/2}}\right)\left(\beta \log\frac{1}{\alpha} - 2\right)^{d-1}.$$

PROOF. We can easily verify that for n large enough we have

$$\frac{h}{\alpha^{d-1}2^{(i_1+\cdots+i_{d-1})/\beta}} \leq \alpha'.$$

Since $2^{1/\beta} - 1 > 1$, it follows that

$$|\Delta| \geq \left|\left\{(i_1, \ldots, i_{d-1}) \in \mathscr{N}^{d-1} \;\middle|\; i_k \leq \beta\left(1 - \frac{1}{(\log 1/\alpha)^{1/2}}\right)\log\frac{1}{\alpha} - 1; 0 < k < d\right\}\right|,$$

hence,

$$|\Delta| \geq \left(\beta\left(1 - \frac{1}{(\log 1/\alpha)^{1/2}}\right)\log\frac{1}{\alpha} - 1\right)^{d-1}.$$

Using (3) to show that $\beta \log(1/\alpha) \geq 2$, if n is large enough, we have

$$|\Delta| \geq \left(1 - \frac{1}{(\log 1/\alpha)^{1/2}}\right)^{d-1} \times \left(\beta \log\frac{1}{\alpha} - 2\right)^{d-1}.$$

For any real x $(0 < x < 2)$ we have $(1-x)^{d-1} \geq 1 - (d-1)x$. (This inequality will be used repeatedly later on without further mention.) Since $1/(\log 1/\alpha)^{1/2} < 2$ for n large enough, the lemma follows readily. □

For n large enough, we have $\log n < 2^{\sqrt{(\log n)/d}}$; therefore, $d\alpha^{1/\sqrt{\log(1/\alpha)}} < 1/\log(1/\alpha)$. Since $t_j \in \hat{\alpha}_d$ if $j \in \Delta$, we have, for such a j,

$$\lambda_d(\tilde{t}_j) \geq \left(1 - \alpha^{1/\sqrt{\log(1/\alpha)}}\right)^d \geq 1 - d\alpha^{1/\sqrt{\log(1/\alpha)}} > 1 - \frac{1}{\log(1/\alpha)}. \tag{5}$$

From (4) we derive that $\Phi_1 \geq (1 - (1-h)^n) \sum_{j\in\Delta} \lambda_d(\tilde{t}_j)$ and therefore, if n is large enough,

$$\Phi_1 > (1 - (1-h)^n)\left(1 - \frac{1}{\log(1/\alpha)}\right)|\Delta|. \tag{6}$$

We now introduce an important quantity,

$$\#\mathscr{V} = \max\{|\mathscr{V}(p)| : p \in [0, 1]^d\},$$

which we estimate below. Lemma 4.4 summarizes our findings about exposed points.

LEMMA 4.3. *For any n large enough,* $(\lfloor \beta \log 1/\alpha \rfloor)^{d-1} \leq \#\mathscr{V} \leq (1 + \beta \log 1/\alpha)^{d-1}$.

PROOF. Obviously, $\#\mathscr{V} = |\mathscr{V}(p)|$, where $p = (1, \ldots, 1) \in [0, 1]^d$. Let $j = (i_1, \ldots, i_{d-1}) \in \mathscr{N}^{d-1}$; since $v(p, j) \in \mathscr{V}(p)$ if and only if $p \in \tilde{t}_j$, an equivalent condition is that (i) for each k $(0 < k < d)$,

$$u_{i_k+1} = \alpha \frac{2^{(i_k+1)/\beta} - 1}{2^{1/\beta} - 1} \leq 1$$

and (ii) $h \leq \alpha^{d-1}2^{(i_1+\cdots+i_{d-1})/\beta}$. Note that for n large enough, (ii) is always satisfied.

Also, one can easily verify that because of (3) condition (i) implies $i_k \leq \beta \log(1/\alpha)$ and is satisfied for $i_k \leq \beta \log(1/\alpha) - 1$. □

LEMMA 4.4. *Let n and m be integers satisfying* (3), *with n large enough, and let ν be a real such that $0 < \nu < \#\mathscr{V}$. If P is a random set of n points in $[0, 1]^d$ and p is a random point in $[0, 1]^d$, then p is ν-exposed with probability greater than*

$$\frac{(1 - (1 - h)^n)(1 - 1/\log(1/\alpha))|\Delta| - \nu}{\#\mathscr{V} - \nu}.$$

PROOF. Let $\Gamma_1 = \{(P, p) \in [0, 1]^{d(n+1)} \mid \pi(P, p) > \nu\}$. Since $\pi(P, p) \leq |\mathscr{V}(p)|$, we have

$$\Phi_1 \leq (\#\mathscr{V})\lambda_{d(n+1)}(\Gamma_1) + \nu(1 - \lambda_{d(n+1)}(\Gamma_1)).$$

The lemma follows from (6) and the fact that $\lambda_{d(n+1)}(\Gamma_1)$ is precisely the probability that p is ν-exposed. □

4.2 MEASURING THE SET OF ISOLATED POINTS. We pursue the same line of reasoning. The difference is that unlike P the set M of red points is not random, but rather, it is specified by an adversary. As usual, M is a set of m points in $[0, 1]^d$, and n and m satisfy (3). Let $j = (i_1, \ldots, i_{d-1}) \in \mathscr{N}^{d-1}$ and $p \in [0, 1]^d$. We define a function $g_j(p)$ as follows:

$$g_j(p) = \begin{cases} 1, & \text{if } w(p, j) \in \mathscr{W}(p) \text{ and } w(p, j) \cap M = \varnothing; \\ 0, & \text{otherwise.} \end{cases}$$

Following the approach of the previous section, we define the expectation $\Phi_2 = \int_{[0,1]^d} \mu(M, p)\, dp$. We have

$$\Phi_2 = \sum_{j \in \mathscr{N}^{d-1}} \int_{[0,1]^d} g_j(p)\, dp = \sum_{j \in \mathscr{N}^{d-1}} \int_{\tilde{i}_j} g_j(p)\, dp \geq \sum_{j \in \Delta} \int_{\tilde{i}_j} g_j(p)\, dp. \tag{7}$$

Let $w(p, j) \in \mathscr{W}(p)$; by definition we have

$$\lambda_d(w(p, j)) = \lambda_d(\tilde{a} \cap \hat{p}) - \lambda_{d-1}(r(\tilde{p}, j))\left(1 - \frac{1}{2^{1/\beta}}\right)z,$$

where $z = h/\lambda_{d-1}(r(\tilde{p}, j))$. We derive

$$\begin{aligned} \lambda_d(w(p, j)) &= \frac{\prod_{0<k<d} u_{i_k+1}}{\lambda_{d-1}(r(\tilde{p}, j))} h - \left(1 - \frac{1}{2^{1/\beta}}\right)h \\ &= \frac{h}{(2^{1/\beta} - 1)^{d-1}} \prod_{0<k<d} \left(2^{1/\beta} - \frac{1}{2^{i_k/\beta}}\right) - \left(1 - \frac{1}{2^{1/\beta}}\right)h, \end{aligned}$$

therefore,

$$\lambda_d(w(p, j)) \leq \left(\left(\frac{1}{1 - 1/2^{1/\beta}}\right)^{d-1} + \frac{1}{2^{1/\beta}} - 1\right)h. \tag{8}$$

The derivative of $dx + 1 - 1/(1 - x)^{d-1}$ has a zero at $x = 1 - (1 - 1/d)^{1/d}$, so it is immediate that for all x satisfying $0 < x < 1 - (1 - 1/d)^{1/d}$,

$$\left(\frac{1}{1 - x}\right)^{d-1} < 1 + dx. \tag{9}$$

Because $d > 1$ and $0 < \epsilon < 1$, we have $(1 - 1/d^{10})^d > 1 - 1/d$; hence, $1/d^{10/\epsilon} < 1 - (1 - 1/d)^{1/d}$. From (3) we find that

$$\frac{1}{2^{1/\beta}} < \frac{1}{d^{10/\epsilon}} < 1 - \left(1 - \frac{1}{d}\right)^{1/d},$$

therefore from (8) and (9), we derive

$$\Lambda = \max_{p \in [0,1]^d} \max_{r \in \mathscr{V}(p)} \lambda_d(r) < \frac{(d+1)h}{2^{1/\beta}}$$

and from (7), $\Phi_2 \geq \sum_{j \in \Delta} (\lambda_d(\tilde{t}_j) - m\Lambda)$, hence from (5)

$$\Phi_2 > \left(1 - \frac{1}{\log(1/\alpha)} - \frac{(d+1)mh}{2^{1/\beta}}\right) |\Delta|. \tag{10}$$

LEMMA 4.5. *Let n and m be integers satisfying* (3), *with n large enough, and let ν be a real such that $0 < \nu < \#\mathscr{V}$. If M is an arbitrary set of m points in $[0, 1]^d$, then a random point in $[0, 1]^d$ is ν-isolated with probability greater than*

$$\frac{(1 - 1/\log(1/\alpha) - (d+1)mh/2^{1/\beta})|\Delta| - \nu}{\#\mathscr{V} - \nu}.$$

PROOF. Let $\Gamma_2 = \{p \in [0, 1]^d \mid \mu(M, p) > \nu\}$. For each $p \in [0, 1]^d$, $\mu(M, p) \leq |\mathscr{M}(p)| = |\mathscr{V}(p)|$; therefore, $\Phi_2 \leq (\#\mathscr{V})\lambda_d(\Gamma_2) + \nu(1 - \lambda_d(\Gamma_2))$, which because of (10) completes the proof. □

4.3 THE LOWER BOUND, AT LAST. Recall that ϵ is a real ($0 < \epsilon < 1$) and n and m satisfy (3). Let $\gamma = \beta \log(1/\alpha)$ and $\nu = \#\mathscr{V} - \gamma^{d-1}/3$. Let $M(P)$ be an arbitrary mapping of a set P of n points in $[0, 1]^d$ into a P-cover of size m. We define Π as the probability that if P is a random set of n points in $[0, 1]^d$ and if p is a random point of $[0, 1]^d$, then $c(P, M(P), p) > 2\nu - |\mathscr{V}(p)|$. Let Π_1 be the probability that given a random set P of n points in $[0, 1]^d$, a random point of $[0, 1]^d$ is ν-exposed with respect to P, and let $h_1(P)$ be the measure of the set of points in $[0, 1]^d$ that are ν-exposed. We define $\mathscr{M} = \{M \subseteq [0, 1]^d : |M| = m\}$ and $\mathscr{M}(P)$ as the set of all P-covers of size m, given $P \subseteq [0, 1]^d$ and $|P| = n$. Let $h_2(M)$ be the Lebesgue measure of the ν-isolated subset of $[0, 1]^d$, given $M \in \mathscr{M}$, and put $\Pi_2 = \min\{h_2(M) \mid M \in \mathscr{M}\}$. Using Lemma 4.1, we have

$$\begin{aligned}
\Pi &\geq \int_{[0,1]^{dn}} \min_{M \in \mathscr{M}(P)} \lambda_d(\{p \in [0, 1]^d \mid c(P, M, p) > 2\nu - |\mathscr{V}(p)|\})\, dP \\
&\geq \int_{[0,1]^{dn}} \min_{M \in \mathscr{M}(P)} \lambda_d(\{p \in [0, 1]^d \mid p \text{ is } \nu\text{-hyperbolic}\})\, dP \\
&\geq \int_{[0,1]^{dn}} (h_1(P) + \min\{h_2(M) \mid M \in \mathscr{M}\} - 1)\, dP,
\end{aligned}$$

therefore

$$\Pi \geq \Pi_1 + \Pi_2 - 1. \tag{11}$$

Applying Lemmas 4.4 and 4.5, we find that for n large enough we have $1 - \Pi < A/B$, where

$$A = 2(\#\mathscr{V}) - |\Delta| \left((2 - (1 - h)^n)\left(1 - \frac{1}{\log(1/\alpha)}\right) - \frac{(d+1)mh}{2^{1/\beta}}\right), \tag{12}$$

and

$$B = \frac{\gamma^{d-1}}{3}. \tag{13}$$

We derive an upper bound on A via several approximations. We begin with a technical result.

LEMMA 4.6. *For any reals* $x, y \geq 2$, *we have* $x^y > (x-2)y^2$.

PROOF. Let $\phi(x, y) = x^y - (x-2)y^2$ and $\psi(y) = 2^{y-1} - y$. Since $\psi'(y) = 0$ for $y = 1 - \log\ln 2 < 2$, we have $\psi(y) \geq 0$; hence, $y^{1/(y-1)} \leq 2$, for $y \geq 2$. But $\partial\phi(x, y)/\partial x = 0$ at $x = y^{1/(y-1)}$; therefore, $\phi(x, y) \geq \phi(2, y) > 0$, for $x, y \geq 2$. □

From (3) we find that

$$\gamma > \frac{5^{d+4}}{d\epsilon}. \tag{14}$$

Lemma 4.3 shows that $\#\mathscr{V} \leq \gamma^{d-1}(1 + 1/\gamma)^{d-1}$, which is easily shown to be at most

$$\gamma^{d-1}\left(1 + \frac{(d-1)2^{d-2}}{\gamma}\right),$$

because $\gamma > 1$. Using (14) and Lemma 4.6 we find that

$$\frac{2^{d-2}}{\gamma} < \frac{d\epsilon}{4^5(5/2)^d} < \frac{2\epsilon}{4^5 d};$$

therefore, we have

$$\#\mathscr{V} < \gamma^{d-1}\left(1 + \frac{\epsilon}{500}\right). \tag{15}$$

From (14), we know that $\gamma > 1$; therefore,

$$(\gamma - 2)^{d-1} = \gamma^{d-1}\left(1 - \frac{2}{\gamma}\right)^{d-1} > \gamma^{d-1}\left(1 - \frac{2d}{\gamma}\right).$$

Using (14) and Lemma 4.6, it follows that

$$(\gamma - 2)^{d-1} > \gamma^{d-1}\left(1 - \frac{\epsilon}{900}\right).$$

From Lemma 4.2, we then have

$$|\Delta| > \left(1 - \frac{d\sqrt{d}}{\sqrt{\log n}}\right)(\gamma - 2)^{d-1},$$

so for n large enough,

$$|\Delta| > \left(1 - \frac{\epsilon}{800}\right)\gamma^{d-1}. \tag{16}$$

If n is large enough,

$$\left(1 - \frac{5}{\epsilon n}\right)^n < \frac{1}{2^{5/\epsilon}} < \frac{1}{3^{3/\epsilon}},$$

so from Lemma 4.6, we have $(1 - 5/(\epsilon n))^n < \epsilon^2/9$, hence

$$(2 - (1 - h)^n)\left(1 - \frac{1}{\log(1/\alpha)}\right) = \left(2 - \left(1 - \frac{5}{\epsilon n}\right)^n\right)\left(1 - \frac{d}{\log n}\right) > 2 - \frac{\epsilon}{8}. \quad (17)$$

Next, we establish the relation

$$\frac{(d+1)mh}{2^{1/\beta}} < \frac{\epsilon}{5}. \quad (18)$$

From (3), we have

$$\frac{(d+1)mh}{2^{1/\beta}} = \frac{5(d+1)}{\epsilon(m/n)^{5/\epsilon - 1}} < \frac{5(d+1)d^2}{\epsilon d^{10/\epsilon}}. \quad (19)$$

We distinguish between two cases:

(i) $\epsilon < \frac{3}{5}$. From Lemma 4.6, we have $d^{10/\epsilon} > (d^8)^{6/(5\epsilon)} > (36/25)(d^8 - 2)/\epsilon^2 > (11/8)d^8/\epsilon^2$; therefore, $(d+1)mh/2^{1/\beta} < 60\epsilon/(11d^5) < \epsilon/5$, since $d \geq 2$.

(ii) $\epsilon \geq \frac{3}{5}$. From (19) we have $(d+1)mh/2^{1/\beta} < 25(d+1)/(3d^8) \leq 25/(2d^7) < \epsilon/5$.

Relation (18) is thus established. Putting together the inequalities (15–18), we derive from (12)

$$A < \frac{663}{2000}\gamma^{d-1}\epsilon < \gamma^{d-1}\frac{\epsilon}{3},$$

which, combined with (13), gives $\Pi > 1 - \epsilon$. Using the lower bound of Lemma 4.3, we conclude that with probability greater than $1 - \epsilon$ we have

$$c(P, M(P), p) > 2v - \#\mathscr{V} \geq \lfloor \gamma \rfloor^{d-1} - \frac{2\gamma^{d-1}}{3}.$$

To simplify this lower bound, we use Lemma 4.6 to derive $5^{d+4} > 1875d^2$. From (14) we easily find that

$$\left(1 - \frac{1}{\gamma}\right)^{d-1} > \left(1 - \frac{d}{5^{d+4}}\right)^{d-1} > 1 - \frac{d^2}{5^{d+4}} > \frac{5}{6};$$

therefore, with probability $> 1 - \epsilon$, we have

$$c(P, M(P), p) > \frac{\gamma^{d-1}}{6} \geq \left(\frac{\epsilon \log n}{30d \log(m/n)}\right)^{d-1};$$

this statement being true for any n large enough and m satisfying (3). Assume now that (3) does not hold. If $m \leq d^2 n$, then we augment $M(P)$ with dummy points so as to obtain a set M' with $1 + d^2 n$ points, hence satisfying (3) if n is large enough. Since $c(P, M(P), p) \geq c(P, M', p)$, we have $c(P, M(P), p) = \Omega(\log^{d-1} n)$ with probability greater than $1 - \epsilon$. If now $m \geq n^{1+\epsilon^2/5^{d+5}}$, all the inequality above says is that $c(P, M(P), p) = \Omega(1)$. Although, strictly speaking, the inequality is not applicable, it is still obviously correct. This shows that for some appropriate constant $c > 0$, we have

$$c(P, M(P), p) > c\left(\frac{\log n}{\log(2m/n)}\right)^{d-1}.$$

On the basis of Lemma 2.2, we have established the following result.

Let $\mathscr{S}$ be a faithful commutative semigroup; let d be any positive integer and ϵ any real $(0 < \epsilon < 1)$. There exists a real $c > 0$ (dependent on ϵ) such that the following is true. Let P be a random set of n points in $[0, 1]^d$ and let Γ be any storage scheme for P of size m. If p is a random point in $[0, 1]^d$, then with probability greater than $1 - \epsilon$,

$$t(P, \Gamma, p) \geq c\left(\frac{\log n}{\log(2m/n)}\right)^{d-1}.$$

This means that $t(n, m) = \Omega((\log n/\log(2m/n))^{d-1})$. Moreover, this lower bound is true on the average with respect to both P and p. The proof of Theorem 2.1 is now complete.

5. *Optimality Issues*

The lower bound is provably optimal for any m large enough (in the arithmetic model). Let ϵ be any fixed positive real and assume that $m = \Omega(n(\log n)^{d-1+\epsilon})$. We can show that the worst-case time complexity of dominance searching on n points in d-space satisfies $t(n, m) = \Theta((\log n/\log(m/n))^{d-1})$. We can obviously assume that $d > 1$. The data structure is inspired by a solution to orthogonal range searching proposed by Bentley and Maurer [2].

Let $k = \lfloor (m/n)^{1/(d-1)}/\log n \rfloor$; note that $k > 2$ for n large enough. Divide up the set of n points into subsets $P_1, \ldots, P_l$ of size $\lceil n/k \rceil$ (except possibly for the last one). This partition is to be carried along one coordinate, say, the first one. Then, construct a data structure of dimension d for each of $P_1, \ldots, P_l$ and a data structure of dimension $d - 1$ for each of the sets $\bigcup_{1 \leq i \leq j} P_i^*$ $(1 \leq j < l)$, where P_i^* is the set of points in E^{d-1} obtained by ignoring the first coordinate of each point in P_i. In all these definitions, we assume that the parameter k is *fixed* once and for all. That is, it need not be readjusted at each recursive call. The storage $S(d, n)$ and the query time $Q(d, n)$ obey the inequalities: $(n = n_1 + \cdots + n_l$ and $n_i \leq \lceil n/k \rceil)$

$$S(d, n) \leq \sum_{0<j<l} S(d - 1, n_1 + \cdots + n_j) + \sum_{0<j\leq l} S(d, n_j),$$

$$Q(d, n) \leq \max\left\{ Q\left(d - 1, \sum_{0<i<j} n_i\right) + Q(d, n_j) \,\middle|\, 0 < j \leq l \right\}$$

(sums over empty sets are null) and $S(1, n) = O(n)$ and $Q(1, n) = O(1)$. Let $\lambda = (\log n)/\log k$. We easily derive

$$\begin{aligned} S(d, n) &\leq l \times S(d - 1, n) + \sum_{0<j\leq l} S(d, n_j) \\ &= O(l\lambda S(d - 1, n)) \\ &= O((l\lambda)^{d-1} S(1, n)). \end{aligned}$$

Since $l \leq k$ we easily verify that for n large enough we have $S(d, n) \leq m$. Similarly, we find

$$\begin{aligned} Q(d, n) &\leq Q(d - 1, n) + Q\left(d, \left\lceil \frac{n}{k} \right\rceil\right) \\ &= O(\lambda Q(d - 1, n)) \\ &= O(\lambda^{d-1}). \end{aligned}$$

Since $m = \Omega(n(\log n)^{d-1+\epsilon})$, we have

$$\log k \geq \left(\frac{1}{d-1} - \frac{1}{d-1+\epsilon}\right)\log\frac{m}{n} - O(1) = \Omega\left(\log\frac{m}{n}\right),$$

which implies that $Q(d, n) = O((\log n/\log(m/n))^{d-1})$. This establishes our claim.

THEOREM 5.1. *Let m be the size of a data structure for dominance searching on n points in E^d. There exists a query that cannot be answered in fewer than $c(\log n/\log(2m/n))^{d-1}$ steps, for some fixed constant c. This lower bound is provably optimal in the arithmetic model, for any fixed $\epsilon > 0$ and any $m > n(\log n)^{d-1+\epsilon}$.*

Although the points and the queries used in the proof of Theorem 2.1 are defined with real coordinates, it is clear that the lower bound still holds if the point sets and the queries are required to have integer coordinates. This is an important observation since, after all, computers usually encode finite-precision numbers. Basically, all we need is the ability to encode arbitrary permutations with the coordinates of the points. This can be done if we have on the order of $\log n$ bits per coordinate.

Another question is how off Theorem 5.1 is from what can be done in a more realistic model of computation, such as a random access machine. Without getting into that issue, let us just say that using fast algorithms for successor searching [10], we can come within a factor of $\log\log n$ of the stated lower bound, for m large enough.

Proving that our lower bound is true for all values of m is left as an open problem. Actually, we should observe that the lower bound has little meaning if $m - n$ is not in $\Omega(n)$. Indeed, even for $d = 2$, we have a lower bound of $\Omega(n/(m - n + 1))$ on $t(n, m)$. This lower bound is vacuous for $m = n + \Omega(n)$, but far exceeds the bound of Theorem 2.1 when $m - n$ grows very slowly (e.g., as $\log n$). The claimed lower bound follows by reduction of another searching problem to dominance searching in two dimensions. If we place the points of P on the line $x + y = 1$, dominance searching becomes equivalent to summing up all the entries of an array of size n between query positions i and j. This problem has been studied by Yao [12], who derived an $\Omega(n/(m - n + 1) + \alpha(m, n))$ lower bound on its complexity, where α is a functional inverse of Ackermann's function.

It is interesting to compare this result with the fact, to be proven next, that for certain semigroups the average-case result of Theorem 2.1 is tight for $m = n$. (From the previous paragraph, this obviously is not true in the worst case). As we shall see, the moral of the story is: doing nothing is best! We choose $(N, \max)$ as our semigroup.

THEOREM 5.2. *There exist semigroups for which the expected-time complexity of dominance searching on n points in d-space satisfies $\bar{t}(n, m) = \Theta(\log^{d-1} n)$, for any m such that $n \leq m = O(n)$.*

PROOF. The case $d = 1$ is trivial, so let us assume that $d > 1$ from now on. The data structure is nothing more than the input to the problem; therefore, $m = n$. However, for each point p_i, instead of storing s_i, we shall store $\sum_{p_j \preceq p_i} s_j$. In this way, the average-case time complexity can be expressed as

$$A(n) = \int_{[0,1]^{dn}} \int_{[0,1]^d} m(P, p)\, dp\, dP,$$

where $m(P, p)$ is the number of maxima in $P \cap \hat{p}$. We can easily show that if

$q = (1, \ldots, 1) \in [0, 1]^d$, then $A(n)$ cannot exceed $\int_{[0,1]^{dn}} m(P, q)\, dP$, which from [1] we can show to be in $O(\log^{d-1} n)$. To prove our claim, observe that

$$A(n) = \int_{[0,1]^d} \left(\sum_{0 \le k \le n} \binom{n}{k} (1 - \lambda_d(\hat{p}))^{n-k} \lambda_d^k(\hat{p}) \int_{\substack{Q \subseteq \hat{p} \\ |Q| = k}} m(Q, \hat{p})\, dQ \right) dp$$

$$= \int_{[0,1]^d} \left(\sum_{0 \le k \le n} \binom{n}{k} (1 - \lambda_d(\hat{p}))^{n-k} M(k) \lambda_d^k(\hat{p}) \right) dp,$$

where $M(k)$ is the average number of maxima when k points are drawn uniformly and independently from a hypercube in E^d. Since obviously the number of maxima depends only on the d permutations of the points induced by their coordinates, and that for a given set of d permutations the set of points that realize them has the same measure, $M(k)$ can be obtained by assuming that the coordinates are permutations of $\{1, \ldots, k\}$. Then, we can use a result of Bentley et al. [1] and conclude that $M(k) = O(\log^{d-1} k)$, hence $A(n) = O(\log^{d-1} n)$, Optimality for $m = O(n)$ follows from Theorem 2.1. □

6. *The Dynamic Case*

As usual, let $(\mathcal{S}, +)$ be a faithful commutative semigroup, and let $P = \{p_1, \ldots, p_n\}$ be a set of points in E^d and w be the weight function associated with it. In the dynamic version of dominance searching, one wishes to process a sequence of instructions of the form: given $p \in E^d$, (1) insert $(p, w(p))$ into the database, or (2) compute $\sum_{p_i \le p} w(p_i)$. To do so, an algorithm must specify how to implement these instructions, using an infinite array of registers $z_1, z_2, \ldots$. For an insertion, the unit-time operations allowed are either of the form $z_i := s$, where $s \in S$, or $z_i := \alpha z_k + \beta z_l$ (α, β integer ≥ 0). Queries are answered as in the static case.

Yao [14] has shown that a dynamic algorithm in dimension d can be used to construct a static data structure in dimension $d + 1$. This will allow us to turn our lower bound for the static problem into one for its dynamic version. In the case in which deletions are allowed, Fredman [5] has been able to construct sequences of n instructions requiring $\Omega(n \log^d n)$ operations to be processed. The following result shows that disallowing deletions cannot improve the situation dramatically. As mentioned in the introduction, the result was already derived by Yao [14] for the case $d = 1$, but it is new for any $d > 1$.

THEOREM 6.1. *Consider dominance searching in d-space ($d > 0$) over a commutative faithful semigroup. For any $n > 2$, there exists a sequence of n instructions (inserts or queries) that requires $\Omega(n(\log n / \log\log n)^d)$ time to process.*

PROOF. Without loss of generality, assume that n is of the form $n = 3k > 6$. Because of Theorem 2.1, we know that there exists a constant $c > 0$ such that the following is true. For any $k > 0$, there exists a set of k points in $[0, 1]^{d+1}$ such that, for any storage scheme Γ for P of size m, we have $t(P, \Gamma, p) \ge c(\log k / \log(2m/k))^d$ for a random point $p \in [0, 1]^{d+1}$ with probability greater than one half. Let $\{p_1, \ldots, p_k\}$ be the points of P sorted by x_1-coordinates and let $p_i = (x_i, y_{i,1}, \ldots, y_{i,d})$ and $q_i = (y_{i,1}, \ldots, y_{i,d})$ $(1 \le i \le k)$. Let D be a data structure for dominance searching in d-space and let Γ be the storage scheme for P constructed as follows. Initially, Γ consists of the k semigroup values associated with the points of P. Then, as each q_i is inserted into D, for $i = 1, \ldots, k$, and as various queries are processed, Γ collects all the generators on the p_i's induced by the generators on the q_i's created by D in the process. If T is the time of execution

of a program on D, then certainly $|\Gamma| < c_1 T$, for some constant $c_1 > 0$ independent of k. Next, we define a language for specifying instructions to D.

(1) "$I(q)$" means "*insert q in D*".
(2) "Q" means "*ask the hardest query at current time*".

Let $K = \{v_1, \ldots, v_{2k}\}$ be the sequence formed by merging $\{x_1, \ldots, x_k\}$ and $\{i/k \mid 1 \le i \le k\}$. We form the program J by replacing in K each "x_i" by "$Q, I(q_i)$", and each "i/k" by "Q". Note that J consists precisely of n instructions. Let $r_0, \ldots, r_{2k-1}$ be the open intervals: $r_0 = (0, v_1)$ and $r_i = (v_i, v_{i+1})$ $(0 < i < 2k)$. Let Γ be the storage scheme formed by J, as described earlier. Put $\gamma = c(\log k/\log(2|\Gamma|/k))^d$. For each i $(0 \le i < 2k)$, mark r_i if $r_i \times [0, 1]^d$ contains a point p such that $t(P, \Gamma, p) \ge \gamma$. Since such points are to be found at random with probability greater than $\frac{1}{2}$ and the length of each r_i is $\le 1/k$, at least $\lceil k/2 \rceil$ intervals will be marked. But since Γ can only get richer and thus improve over time (like good wine), at least $\lceil k/2 \rceil$ queries in J take time $\ge c_2\gamma$, for some constant $c_2 > 0$. We immediately derive

$$T \ge \frac{c_2 k \gamma}{2} \ge \frac{1}{2} c_2 k c \left(\frac{\log k}{\log(c_1 T/k)} \right)^d,$$

from which it follows that $T \ge c_3 k(\log k/\log\log k)^d$, for some constant $c_3 > 0$. □.

7. *Conclusions and Open Problems*

Let us mention a few intriguing open problems. To begin with, is our lower bound on the complexity of dominance searching optimal when the storage is between, say, $2n$ and $O(n \log^{d-1} n)$? Is there a matching upper bound for the $\Omega(n(\log n/\log\log n)^d)$ lower bound given in this paper for the dynamic version of the problem? Also, what is the complexity of orthogonal range searching in the so-called *group model*, where we allow an inverse operation? Recently Willard has partially generalized Fredman's technique to the group model [11]. To our knowledge, however, nothing is known about the static case in higher dimensions.

Concerning the arithmetic model of computation, one must decide to what extent the storage scheme requirement can be relaxed. We saw in Section 2 that there is no problem doing so if the semigroup is rich enough, that is, sufficiently powerful to simulate set-union. But what about simpler semigroups? For example, how hard is it to determine whether a query d-range contains at least one point of P? Clearly, the complexity of this problem must have something to do with searching through the data structure and not so much with adding things up. So, it seems that a computational, rather than a combinatorial, model is needed.

ACKNOWLEDGMENTS. I wish to thank the referees for many useful comments that helped to improve the style and presentation of this paper.

REFERENCES

1. BENTLEY, J. L., KUNG, H. T., SCHKOLNICK, M., AND THOMPSON, C. D. On the average number of maxima in a set of vectors, and applications. *J. ACM 25*, 4 (Oct. 1978), 536–543.
2. BENTLEY, J. L., AND MAURER, H. A. Efficient worst-case data structures for range searching. *Acta Inf. 13* (1980), 155–168.
3. CHAZELLE, B. Lower bounds for orthogonal range searching: I. The reporting case. *J. ACM 37*, 2 (Apr. 1990), 200–212.
4. ERDÖS, P., AND SPENCER, J. *Probabilistic Methods in Combinatorics*. Academic Press, Orlando, Fla., 1974.

5. FREDMAN, M. L. A lower bound on the complexity of orthogonal range queries. *J. ACM 28*, 4 (Oct. 1981), 696–705.
6. FREDMAN, M. L. Lower bounds on the complexity of some optimal data structures. *SIAM J. Comput. 10* (1981), 1–10.
7. MEHLHORN, K. *Data structures and algorithms. 3: Multidimensional Searching and Computational Geometry.* Springer-Verlag, New York, 1984.
8. OVERMARS, M. H. The design of dynamic data structures. In Lecture Notes in Computer Science, vol. 156. Springer-Verlag, New York, 1983.
9. VAIDYA, P. M. Space-time tradeoffs for orthogonal range queries. In *Proceedings of the 17th Annual ACM Symposium on Theory of Computing* (Providence, R.I., May 6–8), ACM, New York, 1985, pp. 169–174.
10. WILLARD, D. E. Log-logarithmic worst-case range queries are possible in space $\Theta(n)$. *Inf. Process. Lett. 17* (1983), 81–84.
11. WILLARD, D. E. Lower bounds for dynamic range query problems that permit subtraction. In *Proceedings of the 13th International Colloquium on Automata, Languages, and Programming*, 1986.
12. YAO, A. C. Space-time tradeoff for answering range queries. In *Proceedings of the 14th Annual ACM Symposium on Theory of Computing* (San Francisco, Calif., May 5–7). ACM, New York, 1982, pp. 128–136.
13. YAO, A. C. Lower bounds by probabilistic arguments. In *Proceedings of the 24th Annual IEEE Symposium on Foundations of Computer Science.* IEEE, New York, pp. 420–428.
14. YAO, A. C. On the complexity of maintaining partial sums. *SIAM J. Comput. 14*, 2 (1985), 277–288.

RECEIVED AUGUST 1986; REVISED JUNE 1988; ACCEPTED JUNE 1989