LOWER BOUNDS ON THE COMPLEXITY OF MULTIDIMENSIONAL SEARCHING

Bernard Chazelle

CS-TR-055-86

October 1986

LOWER BOUNDS ON THE COMPLEXITY OF MULTIDIMENSIONAL SEARCHING

by

Bernard Chazelle

Department of Computer Science
Princeton University
Princeton, New Jersey 08544

Abstract:

We establish new lower bounds on the complexity of several searching problems. We show that the time for solving the *partial sum problem* on n points in d dimensions is at least proportional to $\left(\log n / \log \frac{2m}{n}\right)^{d-1}$ in both the worst and average cases; m denotes the amount of storage used. This bound is provably tight for $m = \Omega(n \log^c n)$ and any $c > d-1$. We also prove a lower bound of $\Omega\left(n(\log n / \log\log n)^d\right)$ on the time required for executing n inserts and queries. Other results include a lower bound on the complexity of *orthogonal range searching* in d dimensions (in report-mode). We show that on a pointer machine a query time of $O(s+\text{polylog}(n))$ time can only be achieved at the expense of $\Omega\left(n(\log n / \log\log n)^{d-1}\right)$ space, which is optimal; n and s denote respectively the input and output sizes.

This work was performed while the author was a visiting professor at Ecole Normale Supérieure, Paris, France.

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, 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 [F1] along with a powerful technique for proving lower bounds [F2]. 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 is not too surprising: recent work on dynamization suggests that the coexistence of inserts and deletes often causes a severe increase in complexity [O].

As regards the static case, one of the most interesting results to date is a lower bound of Yao [Y3] on the complexity of the *partial sum problem*: Let $\mathcal{F} = \{(p_i, s_i) \mid 1 \leq i \leq n\}$ be a file of n records, where $p_i \in \Re^2$ and s_i belongs to a semigroup; the problem is to precompute m values in the semigroup to facilitate the answering of questions such as $\sum_{p_i \preceq p} s_i = ?$, with p a query point in $\Re^2$. This problem is fundamental because most *rectangle problems* are in one form or another reducible to it [E]. Yao showed that in the worst case answering a partial sum query takes time $\Omega\big(\log n / \log(\frac{m}{n} \log n)\big)$. He posed as open problems deciding whether this bound was optimal and whether it could be extended to higher dimensions. In particular, he asked whether in $\Re^d$ the time complexity is in $\Omega(\log^{d-1} n)$, when $m = O(n)$.

We answer all these questions. Specifically, we show that in $\Re^d$ the hardest query takes time proportional to $\big(\log n / \log \frac{2m}{n}\big)^{d-1}$ to be answered. (Setting $d = 2$ gives an improvement on Yao's lower bound). Actually, we prove the stronger result that under a uniform distribution this bound is achieved for a random query with probability arbitrarily close to 1; moreover this is also true if the point-set is random. As an immediate consequence our lower bound holds in the worst case as well as on the average. We can prove that these bounds are tight for $m = \Omega\big(n(\log n)^{d-1+\epsilon}\big)$ and any $\epsilon > 0$.

The best previous result for $d = 2$ was the bound by Yao mentioned earlier. For $d > 2$ the best bound was due to Vaidya [V], who showed that the time T to answer an orthogonal range query is $\Omega\big((n/m) \log_T^{d-\theta} n\big)$, where $\theta = 1$ if $d = 3$, and $\theta = 2$ if $d > 3$. Our lower bound is stronger than Vaidya's for all values of d, n, m; also it applies to a more specific problem and is therefore more general.

All logarithms are taken to the base 2, unless specified otherwise.

The second result of this paper makes use of a clever idea of Yao concerning the "dynamization" of lower bounds. We establish the existence of a sequence of n inserts and queries that requires $\Omega\big(n(\log n/\log\log n)^d\big)$ time to be processed. A similar result by Fredman [F1] says that if deletions are allowed then $\Omega(n\log^d n)$ is a lower bound. Recent work on dynamization (e.g., [O]) 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. 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 [Y3] for the case $d = 1$, but it is new for all $d > 1$.

The third contribution of this paper concerns *orthogonal range reporting* in d dimensions, also known as orthogonal range searching in report-mode. We prove the rather surprising result that if a query time of the form $O(s+\text{polylog}(n))$ is sought on a pointer machine, then $\Omega\big(n(\log n/\log\log n)^{d-1}\big)$ storage is necessary (n and s are respectively the input and output sizes); moreover this is optimal. This shows, in particular, that the solution in $O(n\log n/\log\log n)$ space and $O(s+\log n)$ query time proposed in [C1] for the planar case is in fact optimal on a pointer machine.

Most of the proofs in this paper use probabilistic arguments in the sense of [ES,Y2]. This departs from previous methods which were mostly 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.

The next three sections address the three problems mentioned above in the same order. Section 5 closes with a few remarks and open questions.

2. The Partial Sum Problem: the Static Case

This section is fairly intricate so we shall take a number of preliminary steps. First, we outline our approach in the next few paragraphs. Then in §2.1 we define the problem and the model of computation; in §2.2 we establish a weaker result to provide a sense of the main theorem, which is then proven in §2.3. We conclude in §2.4 with the issue of optimality.

We follow Yao [Y3] as regards the problem statement, the model of computation, and a preliminary reduction. This leads us to a game played between Bob and Alice, which goes roughly as follows (we use a variant of Yao's formulation). Let C_d be a hypercube in $\Re^d$ containing n *green* points and m *red* points. We say that a red point *shields* any green point that it dominates (component-wise). Alice makes a move by picking a point p in C_d (preferably not colored) and asking the question: "How many of the red points dominated by p must be used in order to shield all the green points dominated by p?" The minimum number necessary constitutes Alice's score. Bob has control over the red points, but before the game starts it is Alice's job to place the green points in $\mathcal{C}_d$ the way she pleases. Then Bob and Alice play by taking turns: Alice picks p; then Bob places some red points in $\mathcal{C}_d$, which he also reveals to Alice, who then figures out her score. If she finds it acceptable, she can stop. Otherwise, she picks another point p and continues to play. As the game proceeds, Bob reveals more red points to Alice, who can then tune her strategy better each time. Obviously, time is on Alice's side since she can only hope to score higher at the next step. Actually, she should always try to score at least one extra point at each move. In this way, Bob's sole preoccupation will be to terminate the game as soon as possible. Of course, he should not be allowed to throw in any point that does not effectively counter Alice's latest move. Otherwise, Bob would immediately drop all his red points and the game would be over. (This can be achieved by having Bob respond to Alice's move by adding red points one at a time with the requirement that each new point should decrease the current score of Alice's last move).

To make Alice's strategy effective, Yao makes her play on several fronts (to disperse Bob's "troops"), but to keep her decision process tractable, at every move she gives up chunks of territory deemed worthless. This partly accounts for the term $\log\log n$ in Yao's $\Omega\big(\log n/(\log\log n + \log \frac{m}{n})\big)$ lower bound. The reason is that by playing cleverly Bob can then speed up the game and force Alice into a premature exhaustion of her strategy. To avoid this is possible as we shall see (although it will take a bit of effort).

First, we modify the rules and allow Alice to make non-deterministic moves. From a constructivist's viewpoint this is giving Alice the chance to say: "I was wrong in my last few choices and I wish to back up and undo them." Alice's strategy is now quite different. With each point p she considers a linear system of truncated hypersurfaces. Her goal is to find two surfaces in the system between which green points can be clustered together with their red "shielders", and partitioned into a large number of equivalence classes, each class adding 1 to Alice's score. The proper choice of hypersurfaces is the key to Alice's success. She must ensure that, no matter how clever Bob is, a random pick on her part will satisfy the conditions needed with high probability. For technical reasons, we choose hyperspheres in a metric space where the distance between two points is the Lebesgue measure of the smallest hyperrectangle containing them.

Returning to the game in action, we note that Alice's strategy is still incomplete since her choice of green points obviously cannot be left arbitrary. Yao introduces a pseudo-uniformity criterion and proceeds to construct sets of points that satisfy it. Unfortunately, his construction does not seem to extend to arbitrary dimensions. In a different context, Vaidya [V] uses probabilistic methods to prove the existence of point-sets in $\Re^d$ satisfying a weaker uniformity criterion. His criterion is too weak for our purposes, however, as it would enforce the aforementioned term $\log\log n$ which we are precisely seeking to eliminate. Instead, we keep Yao's pseudo-uniformity criterion, but we do not insist that it should hold everywhere in $\mathcal{C}_d$. We show that Alice can place the green points at random with a high probability of satisfying the criterion over a fixed fraction of $\mathcal{C}_d$. Of course, she will have to tune her choice of hypersurfaces later on quite carefully because she does not know which subset of $\mathcal{C}_d$ satisfies the uniformity criterion. For convenience, we shall rely on measure-theoretic notation in formalizing Alice's strategy, rather than using standard probabilistic language.

2.1. Preliminaries

A. The Notation. For $d > 0$, let $\mathcal{C}_d = [0,1]^d$. By a *random point in* $\mathcal{C}_d$ we mean a point drawn randomly from a uniform distribution in $\mathcal{C}_d$. A *random set of points in* $\mathcal{C}_d$ is a set obtained by drawing randomly and independently points from a uniform distribution in $\mathcal{C}_d$. We shall use this abbreviated terminology throughout the paper. We let λ_d designate the Lebesgue measure in $\Re^d$. If $p = (x_1, \ldots, x_d)$ and $q = (y_1, \ldots, y_d)$ are 2 points in $\Re^d$, we write $p \preceq q$ if $x_i \leq y_i$ for each i $(1 \leq i \leq d)$. We define $\hat{p} = \{q \in \mathcal{C}_d | \, q \preceq p\}$ and $\tilde{p} = \{q \in \mathcal{C}_d | \, p \preceq q\}$. A *rectangle* in $\Re^d$ is the Cartesian product of d closed intervals, i.e., $\prod_{1 \leq i \leq d} [a_i, b_i]$. The cardinality of a finite set X is denoted $|X|$. Let $X_1, \ldots, X_m$ be m collections of closed intervals in $\Re$; then $\prod_{1 \leq i \leq m} X_i$ denotes the set $\{I_1 \times \cdots \times I_m | \, I_i \in X_i (1 \leq i \leq m)\}$. Finally, we put $[1 \ldots n] = \{1, 2, \ldots, n\}$ and $N = \{0, 1, 2, \ldots\}$.

B. Semigroups. Let $(S, +)$ be a commutative semigroup with an operation denoted $+$. To rule out trivial semigroups over which complexity questions are vacuous (e.g., $S = \{b\}$ and $b + b = b$), Yao introduces the notion of faithfulness [Y3]. We say that $(S, +)$ is *faithful* if for every $n > 0$, $T_1, T_2 \subseteq [1 \ldots n]$, and every sequence of integers $\alpha_i, \beta_j > 0$ $(i \in T_1, \, j \in T_2)$, the equation

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

cannot be satisfied for all assignments of the variables $s_1, \ldots, s_n$ unless $T_1 = T_2$. Introducing the α_i, β_j's in the definition adds generality (but also makes proving lower bounds more difficult). For example, the semigroup $(N, +)$ is faithful and so is the seemingly "easier" one $(N, \max)$.

C. The Partial Sum Problem. Let $(S, +)$ be a faithful commutative semigroup. A *file* of size n is a collection $\mathcal{F} = \{(p_i, s_i) | \, 1 \leq i \leq n\}$, where $p_i \in \mathcal{C}_d$ and $s_i \in S$ for each $i = 1, \ldots, n$. We define a function $s \colon \mathcal{C}_d \to S$ such that $s(p) = \sum_{p_i \preceq p} s_i$ (we set $s(p) = \emptyset$ if $\{p_i | \, p_i \preceq p\} = \emptyset$). The partial sum problem is computing $s(p)$ efficiently for any $p \in \mathcal{C}_d$.

D. The Model of Computation. Let V be a subset of $[1 \dots n]$ and let $\{\alpha_i \mid i \in V\}$ be a collection of labelled positive integers. For any file $\mathcal{F}$ of size n, let $g(\mathcal{F}) = \sum_{i \in V} \alpha_i s_i$. The function g is called a *generator*: it is the elementary component of any data structure for the partial sum problem. Let $P = \{p_1, \dots, p_n\} \subseteq \mathcal{C}_d$; in the following S and P are *fixed*. A *storage scheme* Γ for (P, S) of size m is a collection of generators $\{g_1, \dots, g_m\}$ such that for *any* file $\mathcal{F}$ of the form $\{(p_i, s_i) \mid 1 \le i \le n\}$ $(s_i \in S)$ and any point $p \in \mathcal{C}_d$ we have

$$s(p) = \sum_{i \in W} \beta_i g_i(\mathcal{F}) \tag{1}$$

(or $= \emptyset$ if $W = \emptyset$), for some integers $\beta_i > 0$ and $W \subseteq [1 \dots m]$. Let $\mathcal{S}_m(P, S)$ designate the set of all storage schemes for (P, S) of size m. Note that Γ is defined for P and S fixed but for *any* assignments of the s_i's. This means that a storage scheme can take advantage of the particular semigroup under consideration as well as of any properties which P may enjoy: However, it must work for all possible assignments of values to the variables associated with the points of P. Next, we define the complexity of a storage scheme. Given $p \in \mathcal{C}_d$, let W be the smallest set such that (1) is true. We put $t(P, \Gamma, p) = |W|$ (defined only if Γ is a storage scheme for (P, S)). We also have

$$t(n, m) = \max_{|P| = n} \ \min_{\Gamma \in \mathcal{S}_m(P, S)} \ \max_{p \in \mathcal{C}_d} t(P, \Gamma, p).$$

Assuming the probability distribution discussed in (A), we define

$$\bar{t}(n, m) = E_{|P| = n} \min\{E_{p \in \mathcal{C}_d} \ t(P, \Gamma, p) \mid \Gamma \in \mathcal{S}_m(P, S)\}.$$

Note that in all these definitions the semigroup S is understood. We shall now state the main result of this section.

Theorem 1. Let S be a faithful commutative semigroup; let d be any positive integer and ϵ any real $(0 < \epsilon < 1)$. There exists a constant $c > 0$ such that the following is true. Let P be a random (nonempty) set of n points in $\mathcal{C}_d$ and let Γ be any storage scheme for (P, S) of size m. If p is a random point in $\mathcal{C}_d$, then with probability greater than $1 - \epsilon$ the time complexity of the partial sum problem satisfies

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

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

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

Remark: Intuitively there is 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 $\mathcal{C}_{d(n+1)}$. To say "let Γ be any storage scheme..." in Theorem 1 is a shorthand for saying that the previous statement is true for an *arbitrary* choice of the function Γ.

E. A Canonical Reduction. Let P and M be 2 finite sets of points in $\mathcal{C}_d$. We say that M is a *P-cover* if, for each point $p \in \mathcal{C}_d$, 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 and there is no such Q, we have $c(P, M, p) = +\infty$). Informally, P represents the green points mentioned earlier and M contains the red shielders. The next result states that $c(P, M, p)$ yields a lower bound on the complexity of the partial sum problem.

Lemma 1. Let S be a faithful commutative semigroup and let P be a finite subset of $\mathcal{C}_d$. Given any storage scheme Γ for (P, S), there exists a P-cover M of cardinality $|\Gamma|$ such that for each point $p \in \mathcal{C}_d$ we have $c(P, M, p) \leq t(P, \Gamma, p)$.

Proof: Let $\Gamma = \{g_1, \ldots, g_m\}$ be a storage scheme for (P, S) and let $P = \{p_1, \ldots, p_n\}$. Given any file $\mathcal{F} = \{(p_i, s_i) | 1 \leq i \leq n\}$ $(s_i \in S)$ we can write $g_i(\mathcal{F}) = \sum_{j \in V_i} \alpha_{i,j} s_j$, for some $V_i \subseteq [1 \ldots n]$ and integers $\alpha_{i,j} > 0$. Let $M = \{q_1, \ldots, q_m\}$ be a set of m points defined as follows: For each i $(1 \leq i \leq m)$

$$\hat{q}_i = \bigcap_{X_i \subseteq \hat{p}} \hat{p},$$

with $X_i = \{p_j | j \in V_i\}$. Next we show that M is a P-cover. Let p be a point of $\mathcal{C}_d$ and let $s(p) = \sum_{i \in W} \beta_i g_i(\mathcal{F})$. We have

$$s(p) = \sum_{i \in W} \beta_i \sum_{j \in V_i} \alpha_{i,j} s_j,$$

which because S is a commutative semigroup we can write as $\sum_{i \in W'} \gamma_i s_i$, with $W' = \bigcup_{i \in W} V_i$ and $\gamma_i = \sum_j \beta_j \alpha_{j,i}$ $(j \in W$ and $i \in V_j)$. It now suffices to show that $Q = \{q_i | i \in W\}$ satisfies (2) and the proof will be complete. By definition, we know that $s(p) = \sum_{p_i \in \hat{p}} s_i$. On the other hand, we have $s(p) = \sum_{i \in W'} \gamma_i s_i$, so by faithfulness we derive $P \cap \hat{p} = \{p_i | i \in W'\}$. This implies that for each $i \in W$ we have $X_i \subseteq \hat{p}$, therefore $\hat{q}_i \subseteq \hat{p}$, hence $Q \subseteq \hat{p}$. Also, since for each $i \in W$, $X_i \subseteq \hat{q}_i$, we have

$$P \cap \hat{p} = \{p_i | i \in W'\} = \bigcup_{i \in W} X_i \subseteq \bigcup_{i \in W} \hat{q}_i = \bigcup_{q \in Q} \hat{q},$$

which completes the proof. ∎

Remark: It is easy to show that a P-cover contains P, therefore $|M| \geq |P|$. We shall use this fact later on without further reference to it.

2.2. A Weaker Result

This section is included to illustrate in a simpler context some of the ideas used in the proof of the main theorem. The proof in question is fairly technical and involves a large number of quantities, some of which will be motivated in this preliminary digression. We shall rederive Yao's result [Y3] using our techniques on hypersurfaces. We shall show that in the case $d = 2$, $t(m, n) = \Omega\big(\log n / \log(\frac{m}{n} \log n)\big)$. This result will be improved later on, so technically speaking this section may be skipped entirely by the reader.

To begin with, we define a criterion of weak uniformity. We say that a set of n points in $\mathcal{C}_2$ is *weakly uniform* if the points are in general position and any rectangle in $\mathcal{C}_2$ of measure $64 \log n/n$ contains at least one point of the set. In the following we use the expression "sufficiently large" to mean "larger than some conveniently chosen constant".

Lemma 2. For any n sufficiently large, a random set of n points is weakly uniform with probability greater than $1 - 1/n^4$.

Proof: Let $I = \{[i/2^k, (i+1)/2^k] \,|\, 0 \leq k \leq \lceil \log n \rceil \text{ and } 0 \leq i < 2^k\}$ and $\mathcal{J} = \{r \in I^2 \,|\, n\lambda_2(r) \geq 4 \log n\}$. (The elements of $\mathcal{J}$ are somewhat similar to what Vaidya [V] calls canonical boxes). We assume throughout the proof that n is large enough. We shall successively show that $\mathcal{J}$ is linear in size, that "big" rectangles always contain members of $\mathcal{J}$, and finally that a random throw of n points in $\mathcal{C}_2$ will hit each rectangle of $\mathcal{J}$ with high probability. Let $n(k, l)$ be the number of elements of $\mathcal{J}$ 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 $n(k, l) \leq 2^{k+l}$ in general. Therefore,

$$|\mathcal{J}| \leq \sum_{k,l \,|\, 2^{k+l} \log n \leq n/4} n(k, l) \;\leq\; \left(\sum_{i \leq \log n - \log\log n - 2} 2^i \right) \log n \leq n.$$

Let $R = [x_1, x_2] \times [y_1, y_2] \subseteq \mathcal{C}_2$ be a rectangle of measure $64 \log n/n$, and let $k = \left\lceil \log \frac{1}{x_2 - x_1} \right\rceil + 1$ and $i = \lceil x_1 2^k \rceil$. To prove that $[i/2^k, (i+1)/2^k] \subseteq [x_1, x_2]$, it suffices to show that $i + 1 \leq 2^k x_2$. But this follows from the fact that

$$2^k (x_2 - x_1) = 2^{1 + \left\lceil \log \frac{1}{x_2 - x_1} \right\rceil} (x_2 - x_1) \geq 2,$$

hence $2^k x_2 \geq 2 + 2^k x_1 > i+1$. We easily verify that $0 \leq k \leq \lceil \log n \rceil$ and $0 \leq i < 2^k$, therefore $r_1 = [i/2^k, (i+1)/2^k]$ lies in $[x_1, x_2]$ and is a member of I. Similarly, we derive the existence of $r_2 \in I$ such that $r_2 \subseteq [y_1, y_2]$. Since

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

and a similar inequality holds for r_2, we conclude to the existence of a rectangle $r \subseteq R$, with $r \in I^2$ and

$$\lambda_2(r) \geq \frac{1}{16}\lambda_2(R) = 4\log n/n,$$

hence $r \in \mathcal{J}$. To summarize, any rectangle in $\mathcal{C}_2$ of measure $64\log n/n$ contains a rectangle of $\mathcal{J}$. To complete the proof, we shall show that if P is a random set of n points in $\mathcal{C}_2$, then with probability $> 1 - 1/n^4$ every rectangle of $\mathcal{J}$ contains a point of P. The probability that no rectangle of $\mathcal{J}$ is empty is at least

$$1 - |\mathcal{J}|(1 - 4\log n/n)^n > 1 - |\mathcal{J}|e^{-4\log n}$$

(for n large enough), which is at least $1 - n/e^{4\log n} \geq 1 - 1/n^4$. ∎

Let P be a weakly uniform set of n points in $\mathcal{C}_2$ and let M be a P-cover of size $m \leq n^2$. For clarity, we introduce some parameters: $a = 1/\sqrt{16m\log n}$, $b = 1/16$, $c = \frac{3000m\log^2 n}{n}$, and $\delta = \left\lfloor \frac{1}{5}\log n/\log\left(\frac{m}{n}\log n\right)\right\rfloor$. Let

$$\mathcal{B} = [0,b] \times [0,a] \bigcap \{(x,y) \,|\, xy \leq a^2\};$$

as usual, we assume that n is large enough, so in particular we have $a < b$. Let x and x' be 2 reals $(a \leq x < x' \leq b)$ and put $\chi_1 = (x, a^2/x')$ and $\chi_2 = (x', a^2/x)$: χ_1 and χ_2 are 2 points on each side of the hyperbolic curve delimiting $\mathcal{B}$. We define $b(x,x') = \tilde{\chi}_1 \cap \hat{\chi}_2$. Any rectangle of the form $b(x,x')$ is called a *box* (Fig.1). We say that a box is *valid* if its measure is equal to $64\log n/n$. Also, we say that the set $b_1, \ldots, b_k$ forms a *chain* of boxes if each b_i is a box and the intersection of any 2 is empty ($b_i \cap b_j = \emptyset$, if $i \neq j$). We need the following technical lemma.

Lemma 3. For any n large enough, there exists a chain of δ valid boxes.

Proof: Let $k = \lfloor \log \frac{b}{a}/\log c \rfloor$ and let $b_i = b(ac^i, ac^{i+1})$ for all i $(0 \leq i < k)$. Since $m \geq n$ and n is large enough, we have $c > 1$ and $ac^i \geq a$. With the inequality $ac^k \leq b$ we can conclude that each b_i is a box. We have

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

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

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

assuming as usual that n is sufficiently large. ∎

Let $p = (p_x, p_y) \in \mathcal{C}_2$ and $C(p) = \{(p_x - x, p_y - y) \mid (x, y) \in \mathcal{B}\}$. We say that a point p is *clear* if $C(p) \subseteq \mathcal{C}_2$ and $C(p) \cap M = \emptyset$. Some intuition might be helpful at this stage. With each point p the region $C(p)$ associates a chain of boxes (Lemma 3) that are confined between 2 hyperbolic curves ($xy = a^2c$ and $xy = a^2/c$, up to rotation). Each box being valid, it contains green points (Lemma 2), so it will add 1 to Alice's score provided that the point p is clear.

Lemma 4. For any n sufficiently large, a random point of $\mathcal{C}_2$ is clear with probability $> 3/4$.

Proof: Let μ be the probability that p is not clear. The condition $C(p) \subseteq \mathcal{C}_2$ contributes $1 - (1 - a)(1 - b)$ to μ. As to $C(p) \cap M = \emptyset$, each point of M contributes at most the measure of $\mathcal{B}$, that is,

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

therefore

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

hence $\mu \leq \frac{1}{8} + \frac{\log m}{32 \log n}$, for n large enough. Since we have assumed that $m \leq n^2$, this gives $\mu \leq 3/16$, hence the lemma. ∎

If P is weakly uniform and p is clear, then each of the valid boxes of $C(p)$ provided by Lemma 3 contains at least one point of P (note that these boxes lie entirely in $\mathcal{C}_2$). Since the boxes are pairwise disjoint and $C(p) \cap M = \emptyset$, no point of M in $\hat{p}$ can dominate 2 points of P in distinct boxes. Consequently, $c(P, M, p) \geq \delta$ (Fig.2). Since $\delta = 0$ if $m > n^2$, we conclude that in all cases $c(P, M, p) \geq \delta$, hence $t(n, m) = \Omega\left(\log n / \log\left(\frac{m}{n} \log n\right)\right)$ in the planar case. This is Yao's lower bound. Our result is actually slightly stronger. If we look at the pair (P, p) as a random point in $\mathcal{C}_{2n+2}$, then Lemmas 2 and 4 show that $c(P, M, p) \geq \delta$ with probability at least $\frac{3}{4}(1 - 1/n^4) \geq 1/2$, for $n > 1$. It follows that $\bar{t}(n, m) = \Omega\left(\log n / \log\left(\frac{m}{n} \log n\right)\right)$.

To improve on this result, we must deal primarily with two problems. First, the weak uniformity criterion is strengthened by requiring that any rectangle of measure inversely proportional to the density of P should intersect P. Unfortunately, this seems difficult to achieve (especially if it must be true on the average), so we replace "any" by "most". Secondly, the *clear*-ness condition is too strong. We weaken it by requiring that $C(p)$ should be free of points of M over many "large" subregions.

2.3. The Main Theorem

2.3.1. Introduction

We assume in the following that $d > 1$. Let ϵ be an arbitrary real $(0 < \epsilon < 1)$. For convenience we introduce some parameters

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

as well as a relation which, unless specified otherwise, is assumed from now on to be satisfied by the integers n and m:

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

Next we introduce a tool for "discretizing" the hypersurfaces used later on. Let $p = (x_1, \ldots, x_d) \in \mathcal{C}_d$ and $\bar{p} = (x_1, \ldots, x_{d-1})$. 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 $i > 0$, $u_i = u_{i-1} + \alpha 2^{\frac{i-1}{\beta}}$ (note that $u_i = \alpha \frac{2^{i/\beta}-1}{2^{1/\beta}-1}$). We define the *logarithmic lattice*

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

which consists of rectangles $r(\bar{p}, j) = \prod_{0<k<d} J_{k,i_k}$, where $j = (i_1, \ldots, i_{d-1}) \in N^{d-1}$. 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

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

Fig.3 illustrates these notions in the case $d = 2$. 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. We introduce 2 collections of rectangles:

$$\mathcal{V}(p) = \{v(p,j) \subseteq \mathcal{C}_d \,|\, j \in N^{d-1}\}$$

and

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

Let $P = \{p_1, \ldots, p_n\}$ and $M = \{q_1, \ldots, q_m\}$ be 2 sets of points in $\mathcal{C}_d$; we introduce the functions $\pi(P,p)$ and $\mu(M,p)$, defined for each $p \in \mathcal{C}_d$:

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

and

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

Let $\nu > 0$ be a real. We say that $p \in \mathcal{C}_d$ 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 in that terminology P and M are understood. These definitions find their justification in the following lemma.

Lemma 5. Let P be a finite set of points in $\mathcal{C}_d$ and M be a P-cover. Let $p \in \mathcal{C}_d$ and ν be a real > 0. If p is ν-hyperbolic then $c(P,M,p) > 2\nu - |\mathcal{V}(p)|$.

Proof: Let

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

and

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

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

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

so the proof is complete. ∎

2.3.2. Measuring the Set of Exposed Points

Let $j = (i_1, \ldots, i_{d-1}) \in N^{d-1}$. We define a characteristic function $f_j(P,p)$ as follows:

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

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

$$\Phi_1 = \int_{\mathcal{C}_{dn}} \int_{\mathcal{C}_d} \pi(P,p)\, dp\, dP.$$

Here, we interpret P as a point of $\mathcal{C}_{dn}$. Observing that $v(p,j) \in \mathcal{V}(p)$ if and only if $p \in \tilde{t}_j$, we can write

$$\begin{aligned}\Phi_1 &= \int_{\mathcal{C}_{dn}} \int_{\mathcal{C}_d} \sum_{j \in N^{d-1}} f_j(P,p)\,dp\,dP \\ &= \sum_{j \in N^{d-1}} \int_{\mathcal{C}_d} \int_{\mathcal{C}_{dn}} f_j(P,p)\,dP\,dp \\ &= \sum_{j \in N^{d-1}} \int_{\tilde{t}_j} \left(1-(1-h)^n\right) dp,\end{aligned}$$

therefore

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

The derivations above are valid only if $h < 1$, which is true for n large enough. Once again, throughout this section, we shall make use of the fact that n can be assumed to be larger than any conveniently chosen constant. We continue with a technical result.

Let $\alpha' = \alpha^{1/\sqrt{\log \frac{1}{\alpha}}}$ and $\alpha_d = (\alpha', \ldots, \alpha') \in \Re^d$ and let $\Delta = \{j \in N^{d-1} \,|\, t_j \in \hat{\alpha}_d\}$.

Lemma 6. For any n sufficiently large, we have

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

Proof: We can easily verify that for n large enough we have

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

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

$$|\Delta| \ge \left| \left\{ (i_1, \ldots, i_{d-1}) \in N^{d-1} \,|\, i_k \le \beta \left(1 - \frac{1}{\sqrt{\log \frac{1}{\alpha}}}\right) \log \frac{1}{\alpha} - 1 \ (0 < k < d) \right\} \right|,$$

hence

$$|\Delta| \ge \left(\beta \left(1 - \frac{1}{\sqrt{\log \frac{1}{\alpha}}}\right) \log \frac{1}{\alpha} - 1\right)^{d-1}.$$

Using (3) to show that $\beta \log \frac{1}{\alpha} \ge 2$, if n is large enough, we have

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

For any real x $(0 < x < 2)$ we have $(1-x)^{d-1} \ge 1 - (d-1)x$, and the lemma follows readily. (This last inequality will be used repeatedly later on without further mention). ∎

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

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

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

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

We now introduce an important quantity, $\#\mathcal{V} = \max\{|\mathcal{V}(p)|\colon p \in \mathcal{C}_d\}$, which we can estimate as follows:

Lemma 7. For any n large enough, $\left(\lfloor \beta \log \frac{1}{\alpha} \rfloor\right)^{d-1} \leq \#\mathcal{V} \leq \left(1 + \beta \log \frac{1}{\alpha}\right)^{d-1}$.

Proof: Obviously, $\#\mathcal{V} = |\mathcal{V}(p)|$, where $p = (1, \ldots, 1) \in \mathcal{C}_d$. Let $j = (i_1, \ldots, i_{d-1}) \in N^{d-1}$; since $v(p, j) \in \mathcal{V}(p)$ 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 $i_k \leq \beta \log \frac{1}{\alpha}$ and is satisfied for $i_k \leq \beta \log \frac{1}{\alpha} - 1$, which completes the proof. ∎

The next result concludes our study of exposed points.

Lemma 8. Let n and m be integers satisfying (3), with n large enough, and let ν be a real such that $0 < \nu < \#\mathcal{V}$. If P is a random set of n points in $\mathcal{C}_d$ and p is a random point in $\mathcal{C}_d$, then the point p is ν-exposed with probability greater than

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

Proof: Let $\Gamma_1 = \{(P, p) \in \mathcal{C}_{d(n+1)} \,|\, \pi(P, p) > \nu\}$. Since $\pi(P, p) \leq |\mathcal{V}(p)|$, we have

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

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

2.3.3. Measuring the Set of Isolated Points

As usual, M is a set of m points in $\mathcal{C}_d$, and n and m satisfy (3). This is understood throughout this section. Let $j = (i_1, \ldots, i_{d-1}) \in N^{d-1}$ and $p \in \mathcal{C}_d$. We define a function $g_j(p)$ as follows:

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

Following the approach of the previous section, we put $\Phi_2 = \int_{\mathcal{C}_d} \mu(M,p)dp$. We have

$$\Phi_2 = \sum_{j \in N^{d-1}} \int_{\mathcal{C}_d} g_j(p)dp = \sum_{j \in N^{d-1}} \int_{\tilde{t}_j} g_j(p)dp \geq \sum_{j \in \Delta} \int_{\tilde{t}_j} g_j(p)dp. \tag{7}$$

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

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

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

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

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 $(0 < x < 1 - (1-1/d)^{1/d})$

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

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 (8,9) we derive

$$\Lambda = \max_{p \in \mathcal{C}_d} \max_{r \in \mathcal{W}(p)} \lambda_d(r) < \frac{(d+1)h}{2^{1/\beta}}$$

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

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

Lemma 9. Let n and m be integers satisfying (3), with n large enough, and let ν be a real such that $0 < \nu < \#\mathcal{V}$. If M is an arbitrary set of m points in $\mathcal{C}_d$, then a random point in $\mathcal{C}_d$ is ν-isolated with probability greater than

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

Proof: Let $\Gamma_2 = \{p \in \mathcal{C}_d |\, \mu(M,p) > \nu\}$. For each $p \in \mathcal{C}_d$, $\mu(M,p) \leq |\mathcal{W}(p)| = |\mathcal{V}(p)|$, therefore $\Phi_2 \leq (\#\mathcal{V})\lambda_d(\Gamma_2) + \nu(1 - \lambda_d(\Gamma_2))$, which because of (10) completes the proof. ∎

2.3.4. The Lower Bound

Recall that ϵ is a real $(0 < \epsilon < 1)$ and n and m satisfy (3). Let $\gamma = \beta \log\frac{1}{\alpha}$ and $\nu = \#\mathcal{V} - \gamma^{d-1}/3$. Let $M(P)$ be an arbitrary mapping of a set P of n points in $\mathcal{C}_d$ into a P-cover of size m. We define Π as the probability that if P is a random set of n points in $\mathcal{C}_d$ and if p is a random point of $\mathcal{C}_d$, then $c(P, M(P), p) > 2\nu - |\mathcal{V}(p)|$. Let Π_1 be the probability that given a random set P of n points in $\mathcal{C}_d$, a random point of $\mathcal{C}_d$ is ν-exposed with respect to P, and let $h_1(P)$ be the measure of the set of points in $\mathcal{C}_d$ that are ν-exposed. We define $\mathcal{M} = \{M \subseteq \mathcal{C}_d : |M| = m\}$ and $\mathcal{M}(P)$ as the set of all P-covers of size m, given $P \subseteq \mathcal{C}_d$ $(|P| = n)$. Let $h_2(M)$ be the Lebesgue measure of the ν-isolated subset of $\mathcal{C}_d$, given $M \in \mathcal{M}$, and put $\Pi_2 = \min\{h_2(M) |\, M \in \mathcal{M}\}$. Using Lemma 5, we have

$$\begin{aligned}
\Pi &\geq \int_{\mathcal{C}_{dn}} \min_{M \in \mathcal{M}(P)} \lambda_d\left(\{p \in \mathcal{C}_d |\, c(P, M, p) > 2\nu - |\mathcal{V}(p)|\}\right) dP \\
&\geq \int_{\mathcal{C}_{dn}} \min_{M \in \mathcal{M}(P)} \lambda_d\left(\{p \in \mathcal{C}_d |\, p \text{ is } \nu\text{-hyperbolic }\}\right) dP \\
&\geq \int_{\mathcal{C}_{dn}} \left(h_1(P) + \min\{h_2(M) |\, M \in \mathcal{M}\} - 1\right) dP,
\end{aligned}$$

therefore

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

Applying Lemmas 8 and 9, we find that for n large enough we have $1 - \Pi < A/B$, where

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

and

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

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

Lemma 10. 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 $\frac{\partial\phi(x,y)}{\partial x} = 0$ 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 7 shows that $\#\mathcal{V} \leq \gamma^{d-1}(1+1/\gamma)^{d-1}$, which is easily shown to be at most $\gamma^{d-1}(1+(d-1)2^{d-2}/\gamma)$ (since $\gamma > 1$). Using (14) and Lemma 10 we find that

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

therefore we have

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

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

$$(\gamma-2)^{d-1} = \gamma^{d-1}(1-2/\gamma)^{d-1} > \gamma^{d-1}(1-2d/\gamma).$$

Using (14) and Lemma 10 it follows that

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

From Lemma 6, 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,

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

so from Lemma 10, we have $\left(1 - \frac{5}{\epsilon n}\right)^n < \epsilon^2/9$, hence

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

Next we establish the relation

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

From (3) we have

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

We distinguish between 2 cases:

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

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

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

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

which, combined with (13), gives $\Pi > 1-\epsilon$. Using the lower bound of Lemma 7, we conclude that with probability greater than $1-\epsilon$ we have

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

To simplify this lower bound, we use Lemma 10 to derive $5^{d+4} > 1875d^2$. From (14) we easily find that

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

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

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

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 $1+d^2n$ points, hence satisfying (3) if n is large enough. Since $c(P, M(P), p) \geq c(P, M', p)$, we have $c(P, M(P), p) = \Omega(\log^{d-1} n)$. If now $m \geq n^{1+\epsilon^2/5^{d+5}}$, our previous relation indicates that $c(P, M(P), p) = \Omega(1)$, which is of course always true. We thus have shown, using Lemma 1,

Theorem 1. Let S be a faithful commutative semigroup; let d be any positive integer and ϵ any real $(0 < \epsilon < 1)$. Then there exists a real $c > 0$ such that the following is true. Let P be a random (nonempty) set of n points in $\mathcal{C}_d$ and let Γ be any storage scheme for (P, S) of size m. If p is a random point in $\mathcal{C}_d$, then with probability greater than $1-\epsilon$ the time complexity of the partial sum problem satisfies

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

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

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

2.4. Optimality Issues

How good is the lower bound of Theorem 1? We can prove that it is optimal for "most" values of m/n, but we cannot conclude to its optimality in general. To begin with, let's observe that our bounds have little meaning if $m-n$ is not in $\Omega(n)$. Indeed, even for $d=2$, we have a lower bound of $\Omega(\frac{n}{m-n+1})$ on $t(n,m)$. This lower bound is vacuous for $m=n+\Omega(n)$, but far exceeds the bound of Theorem 1 when $m-n$ grows very slowly (e.g., as $\log n$). The claimed lower bound follows by reduction of another searching problem to the partial sum problem in 2 dimensions. If we place the points of P on the line $x+y=1$, computing partial sums 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 [Y1], who derived an $\Omega\left(\frac{n}{m-n+1}+\alpha(m,n)\right)$ lower bound on its complexity, where α is the 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 1 is tight for $m=O(n)$. (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 2. There exist semigroups for which the expected-time complexity of the partial sum problem on n points in d dimensions satisfies $\bar{t}(n,m)=\Theta(\log^{d-1} n)$, for any m such that $n\le m = O(n)$.

Proof: We can assume that $d>1$ without loss of generality. 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

$$A(n)=\int_{\mathcal{C}_{dn}}\int_{\mathcal{C}_d} m(P,p)\,dp\,dP,$$

where $m(P,p)$ is the number of maxima in $P\cap\hat{p}$. We can easily show that if $q=(1,\ldots,1)\in\mathcal{C}_d$, $A(n)$ cannot exceed $\int_{\mathcal{C}_{dn}} m(P,q)dP$, which from [BKST] we can show to be in $O(\log^{d-1} n)$. To prove our claim, observe that

$$\begin{aligned}A(n)&=\int_{\mathcal{C}_d}\left(\sum_{0\le k\le n}\binom{n}{k}(1-\lambda_d(\hat{p}))^{n-k}\int_{\substack{Q\subseteq\hat{p}\\|Q|=k}} m(Q,\hat{p})dQ\right)dp\\&=\int_{\mathcal{C}_d}\left(\sum_{0\le k\le n}\binom{n}{k}(1-\lambda_d(\hat{p}))^{n-k}M(k)\lambda_d^k(\hat{p})\right)dp,\end{aligned}$$

where $M(k)$ is the average number of maxima when k points are drawn uniformly and independently from a hypercube in $\Re^d$. 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 [BKST] 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 1. ∎

At this point, one should be reminded that the underlying model is not computational but combinatorial. In this regard, lower bounds are admittedly more meaningful than upper bounds. Next, we shall show that as regards the worst case Theorem 1 is optimal if $m = \Omega\left(n(\log n)^{d-1+\epsilon}\right)$.

Theorem 3. Let ϵ be any real > 0. The worst-case time complexity of the partial sum problem on n points in d dimensions satisfies $t(n,m) = \Theta\left(\left(\log n / \log \frac{m}{n}\right)^{d-1}\right)$, for any $m = \Omega\left(n(\log n)^{d-1+\epsilon}\right)$.

Proof: The theorem is clearly true for $d = 1$, so we assume that $d > 1$. The data structure is a straightforward modification of the solution to orthogonal range searching proposed by Bentley and Maurer [BM]. For this reason, we only give a brief sketch of the method. Since $m = \Omega\left(n(\log n)^{d-1+\epsilon}\right)$, we define

$$k = \left\lfloor \frac{(m/n)^{\frac{1}{d-1}}}{\log n} \right\rfloor > 2$$

for n large enough. Let $\lambda = \log n / \log k$. If $n = 1$ the data structure is trivial. If $n > 1$ then 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 we 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 $\bigcup_{1 \le i \le j} P_i^*$ $(1 \le j < l)$, where P_i^* is the set of points in $\Re^{d-1}$ obtained by ignoring the first coordinate of each point in P_i. The storage $S(d,n)$ and the query time $Q(d,n)$ follow the recurrences: $(n = n_1 + \cdots + n_l$ and $n_i \le \lceil n/k \rceil)$

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

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

(sums over empty sets are null) and $S(1,n) = O(n)$ and $Q(1,n) = O(1)$. We easily derive

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

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

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

Since $m = \Omega\left(n(\log n)^{d-1+\epsilon}\right)$, 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 $Q(d,n) = O\left((\log n / \log \frac{m}{n})^{d-1}\right)$. Optimality follows from Theorem 1. ∎

3. The Partial Sum Problem: the Dynamic Case

In the dynamic version of the partial sum problem, one wishes to process a sequence of instructions of the form: (1) insert(p,s) into the current file $\mathcal{F}$ ($p \in \Re^d$, $s \in S$), or (2) compute $s(p) = \sum_{p_i \preceq p} s_i$, for a given $p \in \Re^d$. 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, operations of the form $z_i := s$ or $z_i := \alpha z_k + \beta z_l$ (α, β integer ≥ 0) are allowed. Queries are answered as in the static case.

In [Y3] Yao cleverly observes that a dynamic algorithm in dimension d can be used to construct a static data structure in dimension $d+1$. This allows him to turn his static lower bound into a dynamic one. We follow a similar approach here. (Because we are using a continuous probabilistic model the proof is quite different, however).

In the case where deletions are allowed, Fredman [F1] 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 introduction, this result has already been obtained by Yao [Y3] for the case $d = 1$, but it is new for any $d > 1$.

Theorem 4. Consider the dynamic partial sum problem in $\Re^d$ ($d > 0$) over a commutative faithful semigroup S. 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 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 $\mathcal{C}_{d+1}$ such that, for any storage scheme Γ for (P,S) of size m, we have $t(P,\Gamma,p) \geq c\left(\log k / \log \frac{2m}{k}\right)^d$ for a random point $p \in \mathcal{C}_{d+1}$ with probability $> 1/2$. 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 $q_i = (y_{i,1}, \ldots, y_{i,d})$

$(1 \leq i \leq k)$. Let D be a data structure for the dynamic partial sum problem in $\Re^d$ and let Γ be the storage scheme for (P, S) 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 $|\Gamma| \leq c_1 T$, 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 $\{i/k | 1 \leq i \leq k\}$. 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 $\gamma = c\left(\log k / \log \frac{2|\Gamma|}{k}\right)^d$. For each i $(0 \leq i < 2k)$, mark r_i if $r_i \times C_d$ contains a point p such that $t(P, \Gamma, p) \geq \gamma$. Since such points are to be found at random with probability $> 1/2$ and the length of each r_i is $\leq 1/k$, at least $\lceil k/2 \rceil$ intervals will be marked. But since Γ can only "improve" over time, this means that at least $\lceil k/2 \rceil$ queries in J take time $\geq c_2 \gamma$, for some constant $c_2 > 0$. We derive

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

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

4. Orthogonal Range Reporting

4.1. Preliminaries

Let P be a set of n points in $\Re^d$ and let $Q = I^d$ be the query domain, where $I = \{[x, y] | x \leq y \in \Re\}$. *Orthogonal range reporting* in d dimensions refers to the problem of computing the function $q \in Q \rightarrow P \cap q$, using preprocessing. Throughout this section we assume that the underlying model of computation is a *pointer machine*, as defined by Tarjan [T]. Our main result is that a query time of the form $O(|P \cap q| + \text{polylog}(n))$ can only be achieved at the expense of $\Omega\left(n(\log n / \log \log n)^{d-1}\right)$ storage, and this is optimal.

This shows, rather surprisingly, that the solution in $O(n \log n / \log \log n)$ space and $O(|P \cap q| + \log n)$ query time, given by Chazelle [C1] for the planar case, is in fact optimal. It must be observed that this is not true in the random access machine model. Indeed, in that model, Chazelle [C2] has shown that it is possible to perform the computation in time $O(|P \cap q| + \log n)$ using $O(n \log^\epsilon n)$ storage, for any $\epsilon > 0$.

We begin by recalling a few basic facts about pointer machines. Observe that for the purpose of proving lower bounds one may use any model which is more powerful than the one in which the bounds are intended. This will simplify the description. Following Tarjan [T], a pointer machine consists of a finite number of registers and an unbounded (finite) amount of records: A record consists of a data field and a constant number of addresses (pointers); we can assume this number to be 2 without loss of generality. The memory can be modelled as a directed graph $G = (V, E)$ of outdegree at most 2, endowed with a source s from which every node can be reached. Roughly speaking, the execution of a program can be regarded as a sequential visit of nodes in G, starting at s, with various modifications of data and address fields along the way. The key requirement is that no node v ($\neq s$) can be visited unless a node w has already been visited and $(w, v) \in E$. New nodes can be added to G by requesting them from a pool of free nodes with empty fields — See [T] for details.

For the problem at hand we further assume that, aside from s, G has n distinguished nodes of outdegree 0: $v_1, \ldots, v_n$, where each v_i corresponds to a unique point p_i of P. Let $N(v) = \{w \mid (v, w) \in E\}$. Computing $P \cap q$ involves the execution of a program composed of instructions in the repertoire below: initially, $W = \{s\}$;

1. Pick any $v \in W$ and add $N(v)$ to W.
2. Request a new node v and add it to W $(N(v) = \emptyset)$.
3. Pick any $v, w \in W$ and add (v, w) to E (provided that the outdegree does not exceed 2).
4. Pick any $v, w \in W$ and remove the edge (v, w) from E if it exists.

At termination we must have $\{v_i \mid p_i \in P \cap q\} \subseteq W$. The time complexity is defined as $|W|$. Note that updates of data fields are not mentioned and that the actual implementation of "pick any" in steps 1,3,4 is hidden. In this sense our model is more powerful than an actual pointer machine. We leave it as an exercise to show that a lower bound in our model constitutes a valid lower bound (up to within a constant factor) on the complexity of orthogonal range reporting in the pointer machine model.

We can go even further and show that steps 2–4 can be ignored. Indeed, prior to the computation of $P \cap q$ there must exist a directed path from s to each v_i ($p_i \in P \cap q$), all of whose nodes will belong to W at the end of the computation (simple proof by induction). This implies the existence of a tree T rooted at s, whose nodes belong to W and whose leaves are precisely $\{v_i \mid p_i \in P \cap q\}$. (By a tree we mean a subgraph of G with a source s, whose undirected version is a tree). We use the notation $|T|$ to designate the number of nodes of T. Let $c(q)$ be the number of nodes of the Steiner minimal tree (SMT) of $\{s\} \bigcup \{v_i \mid p_i \in P \cap q\}$ in G. From the previous remark, it is clear that $c(q) \leq |W|$, so $c(q)$ constitutes a lower bound on the time needed to compute $P \cap q$, using G. Next, we introduce some terminology.

Let $a, b > 0$ be 2 real constants. We say that G is (a, b)-effective for P if for each $q \in Q$ we have

$$c(q) \leq a\left(|P \cap q| + \log^b n\right).$$

We assume in the following that G is (a, b)-effective for P. Given $v, w \in V$, let $p(v, w)$ be the number of edges on the shortest path in G from v to w; if there is no such path then $p(v, w) = +\infty$. We also define

$$d(v, w) = \min\{p(z, v) + p(z, w) \mid z \in V\}$$

and for any real $x > 0$,

$$M(x) = |\{(v, w) \in V^2 \mid d(v, w) \leq x\}|.$$

Note that $d(v, w) \leq p(s, v) + p(s, w) < +\infty$.

Lemma 11. For each $x > 0$, $M(x) \leq |V| 2^{2x+2}$.

Proof: For any $z \in V$, we have $|\{v \in V \mid p(z, v) \leq x\}| \leq 2^{x+1} - 1$, since each node has outdegree ≤ 2. This implies that for a given z,

$$|\{(v, w) \in V^2 \mid d(v, w) = p(z, v) + p(z, w) \leq x\}| < 2^{2x+2},$$

which completes the proof. ∎

Assume that $|P \cap q| \geq \log^b n > 4$ and let T be the *SMT* of $\{s\} \bigcup \{v_i \mid p_i \in P \cap q\}$ (i.e. $c(q) = |T|$). We transform T into another tree T' by applying the following steps as long as possible: Pick a node v with a single outgoing edge (v, w). If v has an incoming edge (z, v), remove v and replace the 2 edges (z, v) and (v, w) by (z, w). Otherwise, simply remove v and (v, w). It is clear that T' is a binary tree with $2|P \cap q| - 1$ nodes. Embed it in the plane and let $l_1, \ldots, l_m$ be its leaves from left to right. We have

$$\sum_{0<i<m} d(l_i, l_{i+1}) < 2(|T'| - 1) \leq 2(|T| - 1) < 2c(q).$$

Note that $m = |P \cap q|$. Since G is (a, b)-effective for P we have

$$\sum_{0<i<m} d(l_i, l_{i+1}) < 2a(m + \log^b n) \leq 4am.$$

Let $\Delta(q) = \{(l_i, l_{i+1}) \mid d(l_i, l_{i+1}) < 8a\}$; we immediately derive $|\Delta(q)| \geq m/4$. Suppose now that there exist $q_1, \ldots, q_k \in Q$ such that for each i, j $(i < j)$, $|P \cap q_i| \geq \log^b n$ and $\Delta(q_i) \cap \Delta(q_j) = \emptyset$; then

$$\left|\Delta = \bigcup_{1 \leq i \leq k} \Delta(q_i)\right| \geq km/4.$$

But on the other hand, we have $|\Delta| \leq M(8a) \leq |V|2^{16a+2}$ (Lemma 11). We conclude that

$$|V| \geq mk/2^{16a+4}. \tag{20}$$

We will use a slightly weaker version of this result. The idea is to find "many" rectangles with at least $\log^b n$ points of P in each of them, but with no 2 points belonging to 2 rectangles at the same time.

Given a query $q \in Q$, let $K(q) = \{(p_i, p_j) \in (P \cap q)^2 | i < j\}$. We say that a collection of queries $\{q_1, \ldots, q_k\}$ is *compatible* if for each i, j $(i < j)$ we have $|P \cap q_i| \geq \log^b n$ and $K(q_i) \cap K(q_j) = \emptyset$. From (20) we easily derive

Lemma 12. Let P be a set of n points in $\Re^d$ which admits of a compatible set of k queries. Then solving the orthogonal range reporting problem in $O(s + \log^b n)$ query time, where s denotes the output size, requires $\Omega(k \log^b n)$ storage.

Our next task is to prove the existence of a "large" compatible set of queries. We use probabilistic arguments: first we construct the rectangles, and then we show that with respect to random points these rectangles form a compatible set of queries with probability greater than 0. In the planar case, however, it is easier to give an explicit construction.

4.2. The Planar Case

Each rectangle will contain exactly m points of P. We define $m = \lfloor 2\log^b p \rfloor$ and $\lambda = \left\lfloor \frac{\log p}{1+b\log\log p} \right\rfloor$, where p is an integer large enough so that $m, \lambda > 4$. Let $n = m^\lambda$; for p large enough, it is easy to show that $m \geq \log^b n$ and

$$\lambda \geq \left\lfloor \frac{\log n}{1 + b\log\log n} \right\rfloor.$$

For any integer $i \geq 0$, let $m(i)$ be the integer obtained by writing i in base m $(i = m_1 m_2 \ldots m_\lambda)$ and reversing the order of the digits $(m(i) = m_\lambda \ldots m_2 m_1)$. Let $P = \{(m(i), i) | 0 \leq i < m^\lambda\}$. We construct an m-ary tree $\mathcal{T}$ as follows: Let $\{p_1, \ldots, p_n\}$ be the points of P from left to right, and let $z_1, \ldots, z_m$ be the children of the root z of $\mathcal{T}$ from left to right. With z we associate the list $L(z)$ of points in P sorted by y-coordinates. For each $i = 0, \ldots, m$, we define $L(z_i)$ as the y-sorted list of points in

$$\{p_{(i-1)m^{\lambda-1}+1}, p_{(i-1)m^{\lambda-1}+2}, \ldots, p_{im^{\lambda-1}}\}.$$

For each node of $\mathcal{T}$ we specify a certain number of queries $q_i \in Q$. For the root z, we caliber the queries q_i so that $P \cap q_i$ consists of m consecutive points of $L(z)$ (in x-order). A key observation is that within each $P \cap q_i$ the points appear in distinct lists $L(z_j)$'s. Applying this reasoning recursively leads to a compatible set of $\sum_{1 \leq i \leq \lambda} m^{i-1} \frac{n}{m^i} = \lambda n/m$ queries. From Lemma 12 it then follows that the storage is in $\Omega(n \log n / \log \log n)$. Taking into consideration the matching upper bound of [C1] we conclude with this preliminary result.

Theorem 5. Given n points in the plane, it is possible to solve the orthogonal range reporting problem in $O(n \log n / \log \log n)$ storage and $O(s + \log n)$ query time, where s is the size of the output. The algorithm is optimal in the pointer machine model.

In the next section we will prove the existence of a compatible set of queries of size

$$\Omega\left(\frac{n(\log n/ \log \log n)^{d-1}}{\log^b n}\right).$$

4.3. The General Case

In the following, we assume that $d > 1$. We also continue to use the notation of Section 2.1.A. For convenience we introduce a few parameters

$$\begin{cases} \alpha = \left(\dfrac{4 \log^b n}{n}\right)^{1/d} \\ \mu = 1/(\log n)^{b+d} \\ \delta = \dfrac{4}{n \log^d n}. \end{cases}$$

Whereas dealing with compatibility involves probabilistic arguments, defining the query rectangles can be done explicitly. (In two dimensions the rectangles are somewhat similar to those used in the proof of Lemma 2). The idea is to define several partitionings of $\mathcal{C}_d$, each corresponding to a distinct point with integer coordinates on the hyperplane $\sum_{1 \leq k \leq d} x_k = 0$. For this reason we introduce the set

$$\Im = \left\{ (z_1, \ldots, z_d) \in Z^d \mid \sum_{1 \leq k \leq d} z_k = 0 \right\}.$$

4.3.1. The Set of Query Rectangles

For each $z = (z_1, \ldots, z_d) \in \Im$ and $j = (j_1, \ldots, j_d) \in N^d$, we define the rectangle

$$v(z; j) = \prod_{1 \le k \le d} [\, j_k \alpha \mu^{z_k}, (j_k + 1)\alpha \mu^{z_k}]$$

and the collection

$$\mathcal{G} = \{v(z; j) \subseteq \mathcal{C}_d |\, (z, j) \in \Im \times N^d\}.$$

The elements of $\mathcal{G}$ are called *cells.* They all have the same measure, equal to α^d. The next lemma shows that two cells cannot overlap too much. This will be important later on to satisfy the intersection criterion of compatible sets.

Lemma 13. For any n large enough, the intersection of two distinct cells is either the empty set or a rectangle of measure at most δ.

Proof: Let $c = v(z; j)$ and $c' = v(z'; j')$ be 2 distinct cells. If $z = z'$, then $j \neq j'$ and obviously $\lambda_d(c \cap c') = 0$. If now $z \neq z'$, we can assume without loss of generality that $z_1 < z_1'$. Let c_k (resp. c_k') be the projection of c (resp. c') on the x_k-axis ($1 \le k \le d$). For n large enough we have the following derivations

$$\begin{aligned}
\lambda_d(c \cap c') &= \prod_{1 \le k \le d} \lambda_1(c_k \cap c_k') \\
&\le \prod_{1 \le k \le d} \min(\lambda_1(c_k), \lambda_1(c_k')) \\
&\le \mu^{z_1' - z_1} \lambda_1(c_1) \times \prod_{1 < k \le d} \min(\lambda_1(c_k), \lambda_1(c_k')) \\
&\le \mu \lambda_d(c),
\end{aligned}$$

therefore $\lambda_d(c \cap c') \le \mu \alpha^d$. ∎

Our next result is a lower bound on the size of $\mathcal{G}$.

Lemma 14. For any n large enough, we have

$$|\mathcal{G}| > \frac{n(\log n)^{d-b-1}}{(b+d)^{4d}(\log\log n)^{d-1}}.$$

Proof: We easily verify that for any $z_k \in Z$ such that

$$z_k \geq \frac{1}{b+d}\left(\frac{1+2/d}{\log\log n} + \frac{b}{d} - \frac{\log n}{d\log\log n}\right) \tag{21}$$

we have $\alpha\mu^{z_k} \leq 1/2$. This shows that for any $z = (z_1, \ldots, z_d) \in \Im$, such that (21) holds for $k = 1, \ldots, d$, the number of cells of the form $v(z; j)$ is at least

$$\prod_{1\leq k\leq d}\left\lfloor\frac{1}{\alpha\mu^{z_k}}\right\rfloor > \prod_{1\leq k\leq d}\left(\frac{1}{\alpha\mu^{z_k}} - 1\right) \geq 1/(2\alpha)^d = \frac{n}{2^{d+2}\log^b n}.$$

For n large enough, the r.h.s. of (21) is negative, therefore a simple lower bound on the number of $z \in \Im$ whose coordinates z_k's satisfy (21) is given by

$$\left(\left\lfloor\frac{1}{b+d}\left(\frac{\log n}{d\log\log n} - \frac{1+2/d}{\log\log n} - \frac{b}{d}\right)\right\rfloor + 1\right)^{d-1} > \left(\frac{\log n}{2d(b+d)\log\log n}\right)^{d-1}.$$

This implies that the number of cells in $\mathcal{G}$ exceeds

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

from which the lemma follows. ∎

We will also need an upper bound on the number of cells to which a given point can belong. A rough estimate will be sufficient.

Lemma 15. For any n large enough, no point can lie in the interior of more than $(d\log n/\log\log n)^{d-1}$ cells of $\mathcal{G}$.

Proof: If $z = (z_1, \ldots, z_d)$ is such that there exists $j \in N^d$ with $v(z; j) \subseteq C_d$, then for n large enough z_k exceeds $-\frac{1}{2}\log n/\log\log n$, for $k = 1, \ldots, d$. Since $z \in \Im$, we derive $z_k < \frac{d-1}{2}\log n/\log\log n$. Completing the proof is straightforward. ∎

4.3.2. Placing the Points: Heaviness and Diffusion

The first criterion of compatibility is that the query rectangles should contain at least $\log^b n$ points of P. We strengthen this requirement a little because "bad" points may have to be eliminated later on. We say that a cell is ***heavy*** if it contains in its interior more than $2\log^b n$ points of P.

Lemma 16. Let P be a random set of n points in C_d, with n large enough. Then, with probability greater than $1 - 2/\log^b n$, more than half the cells of $\mathcal{G}$ are heavy.

Proof: Let c be an arbitrary cell of $\mathcal{G}$ and let χ be the number of points of P in the interior of c. We also define π as the probability that c is heavy (i.e. Prob $(\chi > 2\log^b n)$). The mean and variance of χ are respectively $n\lambda_d(c)$ and $n\lambda_d(c)(1-\lambda_d(c))$. Using Chebyshev's inequality we derive

$$1-\pi \le \frac{n\lambda_d(c)(1-\lambda_d(c))}{(n\lambda_d(c)-2\log^b n)^2},$$

therefore $\pi > 1 - 1/\log^b n$. Let Π be the probability that more than half the cells of $\mathcal{G}$ are heavy. Since the expected number of heavy cells is equal to $\pi|\mathcal{G}|$, we have

$$(1-1/\log^b n)|\mathcal{G}| < \pi|\mathcal{G}| \le \frac{1}{2}(1-\Pi)|\mathcal{G}| + \Pi|\mathcal{G}|,$$

which completes the proof. ∎

Next we must ensure that points are not too close to each other, in order to satisfy the second criterion of a compatible set. To that effect, we define the *separation* between 2 points $(x_1, \ldots, x_d)$ and $(y_1, \ldots, y_d)$ in $\Re^d$ as the measure of the smallest rectangle containing them, that is, $\prod_{1\le k\le d}|x_k - y_k|$. Let P be a set of points in $\mathcal{C}_d$; we say that a point $p \in P$ is *stranded* if its separation to any point of $P \setminus \{p\}$ exceeds δ. Finally, we say that P is *ρ-diffuse* ($\rho \in [0,1]$) if it contains at least ρn stranded points.

Lemma 17. Let $\rho = 1 - 1/\sqrt{\log n}$. For any n large enough, a random set of n points in $\mathcal{C}_d$ is ρ-diffuse with probability greater than $1 - 2^{d+3}/\sqrt{\log n}$.

Proof: Let $L(d,y) = \lambda_d\left(\{(x_1,\ldots,x_d) \in \mathcal{C}_d \mid \prod_{1\le k\le d} x_k \le y\}\right)$, with $0 < y < 1$. We have the recurrence relation $L(0,y) = 0$ and, for $d \ge 1$,

$$L(d,y) = y + \int_y^1 L(d-1, y/x)\,dx.$$

Let $M(d,y) = L(d,y) - L(d-1,y)$, for $d > 0$. We derive the simpler recurrence: $M(1,y) = y$ and

$$M(d,y) = \int_y^1 M(d-1, y/x)\,dx,$$

for $d > 1$. This gives

$$M(d,y) = y\int_1^{1/y} M(d-1, 1/x)\,dx,$$

hence

$$M(d,y) = \int_1^{1/y}\int_1^{x_1}\cdots\int_1^{x_{d-2}} \frac{y}{x_1 \times \cdots \times x_{d-1}}\,dx_{d-1}\cdots dx_1.$$

We can evaluate this integral directly. This leads to

$$L(d,y) = \sum_{0 \le k < d} \frac{y}{k!} \left(\ln \frac{1}{y} \right)^k,$$

assuming that $0! = 1$. Since $1/\delta$ goes to infinity with n, we have

$$L(d,\delta) < \frac{\delta d}{(d-1)!} \left(\ln \frac{1}{\delta} \right)^{d-1} < 2\delta (\log n)^{d-1},$$

hence

$$L(d,\delta) < \frac{8}{n \log n}. \tag{22}$$

Consider the locus of points in $\mathcal{C}_d$ whose separation to a given point of $\mathcal{C}_d$ is at most δ. Let V be the largest measure of such a set. It is immediate that

$$V \le 2^d L(d,\delta). \tag{23}$$

Let ν be the expected number of stranded points in P and let π be the probability that P is ρ-diffuse. We have the relation

$$\nu \le (1-\pi)\rho n + \pi n. \tag{24}$$

The probability that a given point of P is stranded is at least equal to $(1-V)^{n-1}$, which exceeds $1-nV$ for n large enough (since V tends to 0 as n goes to infinity). This implies that $\nu > n(1-nV)$. Combining this inequality with (22–24) leads to the lemma. ∎

Lemmas 16 and 17 imply the existence of a set P which is ρ-diffuse and causes more than half the cells of $\mathcal{G}$ to be heavy. We form the set P^* by removing each point of P which is not stranded and replacing it by a point far outside of $\mathcal{C}_d$. In this way, each point of $P^* \cap \mathcal{C}_d$ is stranded.

Let Γ be the cells of $\mathcal{G}$ whose interior contains at least $\log^b n$ points of P^*.

Lemma 18. For n large enough, we have

$$|\Gamma| > \frac{n(\log n)^{d-b-1}}{(b+d)^{5d}(\log\log n)^{d-1}}.$$

Proof: Consider each heavy cell of $\mathcal{G}$ (with respect to P) and mark it if at least half the points it contains are stranded. Since P is ρ-diffuse and because of Lemma 15, the number of heavy cells left unmarked cannot exceed $d^{d-1} n(\log n)^{d-b-3/2}/(\log\log n)^{d-1}$. Since more than half the cells are heavy, Lemma 14 completes the proof. ∎

Since the points of $P^* \cap \mathcal{C}_d$ are stranded, it follows from Lemma 13 that no two cells of Γ can contain the same pair of points of P^*. We derive that Γ forms a compatible set of queries. From Lemmas 12 and 18, we conclude that to solve the orthogonal range reporting problem in $O(s+\text{polylog}(n))$ time, where s is the size of the output, requires $\Omega\big(n(\log n/\log\log n)^{d-1}\big)$ space. In [M, pp.47] Mehlhorn describes a solution to orthogonal range reporting whose performance depends on a slack parameter. When adjusted appropriately, this gives a data structure of size $O\big(n(\log n/\log\log n)^{d-1}\big)$, with $O(s+\text{polylog}(n))$ query time. Our lower bound is therefore optimal.

Theorem 6. On a pointer machine, to solve the orthogonal range reporting problem on n points in $\Re^d$ in $O(s+\text{polylog}(n))$ time, where s is the size of the output, requires $\Omega\big(n(\log n/\log\log n)^{d-1}\big)$ space. This lower bound is optimal.

We leave the converse question open: given $\Theta\big(n(\log n/\log\log n)^{d-1}\big)$ space, what query time can be achieved? We know that $O(s+\text{polylog}(n))$ is within reach, but what is the smallest exponent in the polylogarithmic term? Note that our results on the partial sum problem cannot be used here, or at least not directly, because in the reporting problem the query time is expressed as a function of both input and ouput sizes.

5. Concluding Remarks

It is important to observe that the lower bound proofs given in this paper assume real coordinates only for the sake of convenience. Since in this context point-sets differ combinatorially only if the permutations induced by the coordinates of the points also differ, all the lower bounds still hold if we restrict ourselves to points with integer coordinates.

In closing, we will mention some intriguing open problems. To begin with, is our lower bound on the complexity of the partial sum problem optimal when the storage is $O(n\log^{d-1} n)$? Is there a matching upper bound for the $\Omega\left(n(\log n/\log\log n)^d\right)$ lower bound given in this paper for the dynamic version of the problem? Also, what is the complexity of the partial sum problem in the so-called *group model*, where we allow an inverse operation? Recently Willard has generalized Fredman's technique to the group model [W2]. To our knowledge, however, nothing is known about the static case in higher dimensions. Finally, one should see whether the methods used here can be adapted to other range searching problems (e.g., polygonal/circular range search [EW,HW,W1,Y4,Y5]).

Fredman's arithmetic model of computation, in which our study of the partial sum problem is cast, is very general and any lower bound in that model can be trusted to hold in any "reasonable" sequential model as well. This is not always the case in the other direction, however. An optimal upper bound in the arithmetic model might no longer be tight in a more realistic model, because the cost of computing addresses is not included. Theorem 3 shows the existence of a solution for the partial sum problem on n points in $\Re^2$ requiring, for example, $O(n \log^2 n)$ storage and $O(\log n / \log \log n)$ time. One can easily show that these bounds cannot be achieved in any comparison-based model. But then what is the complexity of the problem in such models? Is there a natural mathematical model which brings about the distinction between the combinatorial and computational aspects of multidimensional searching?

Acknowledgment: I wish to thank Robert E. Tarjan for valuable comments on a draft of this paper.

REFERENCES

[BKST] Bentley, J.L., Kung, H.T., Schkolnick, M., Thompson, C.D. *On the average number of maxima in a set of vectors, and applications*, JACM, 25 (4), Oct. 1978, pp. 536–543.

[BM] Bentley, J.L., Maurer, H.A. *Efficient worst-case data structures for range searching*, Acta Informatica, 13, 1980, pp. 155–168.

[C1] Chazelle, B. *Filtering search: a new approach to query-answering*, Proc. 24th Annu. IEEE Symp. Found. Comput. Sci. (1983), pp. 122–132. To appear in SIAM J. Comput., 15 (3), Aug. 1986.

[C2] Chazelle, B. *A functional approach to data structures and its use in multidimensional searching*, Brown Univ. TR, No. CS–85–16, Sept. 1985. Preliminary version in Proc. 26th Annu. IEEE Symp. on Found. of Comp. Sci., Portland, Oct. 1985, pp. 165–174.

[E] Edelsbrunner, H. *Intersection problems in computational geometry*, PhD Thesis, Tech. Rep. F–93, Techn. Univ. Graz, 1982.

[EW] Edelsbrunner, H., Welzl, E. *Halfplanar range search in linear space and $O(n^{0.695})$ query time*, Rep. F111, Inst. Inform. Proc., Tech. Univ. Graz, Austria, 1983.

[ES] Erdös, P., Spencer, J. *Probabilistic methods in combinatorics*, Academic Press, New York, 1974.

[F1] Fredman, M.L. *A lower bound on the complexity of orthogonal range queries*, J. ACM, 28, 1981, pp. 696–705.

[F2] Fredman, M.L. *Lower bounds on the complexity of some optimal data structures*, SIAM J. Comput. 10 (1), 1981, pp. 1–10.

[HW] Haussler, D., Welzl, E. *Epsilon-nets and simplex range queries*, Proc. 2nd ACM Symp. on Computational Geometry, Yorktown Heights, NY, June 1986, pp. 61–71.

[M] Mehlhorn, K. *Data Structures and Algorithms: Vol. 3. Multi-dimensional searching and computational geometry*, Springer-Verlag, 1984.

[O] Overmars, M.H. *The design of dynamic data structures*, LNCS, Vol. 156, Springer-Verlag, 1983.

[T] Tarjan, R.E. *A class of algorithms which require nonlinear time to maintain disjoint sets*, J. Comput. System Sci., 18, 1979, pp. 110–127.

[V] Vaidya, P.M., *Space-time tradeoffs for orthogonal range queries*, Proc. 17th Annu. ACM Symp. on Theory of Comput., May 1985, pp. 169–174.

[W1] Willard, D.E. *Polygon retrieval*, SIAM J. Comput., 11, 1982, pp. 149–165.

[W2] Willard, D.E. *Lower bounds for dynamic range query problems that permit subtraction*, Proc. 13th ICALP, 1986.

[Y1] Yao, A.C. *Space-time tradeoff for answering range queries*, Proc. 14th Annu. ACM Symp. on Theory of Comput., 1982, pp. 128–136.

[Y2] Yao, A.C. *Lower bounds by probabilistic arguments*, Proc. 24th Annu. IEEE Symp. on Found. of Comp. Sci., Nov. 1983, pp. 420–428.

[Y3] Yao, A.C. *On the complexity of maintaining partial sums*, SIAM J. Comput. 14 (2), May 1985, pp. 277–288.

[Y4] Yao, A.C., Yao, F.F. *A general approach to d-dimensional geometric queries*, Proc. 17th Annu. ACM Symp. on Theory of Comput., May 1985, pp. 163–168.

[Y5] Yao, F.F. *A 3–space partition and its applications*, Proc. 15th Annu. ACM Symp. on Theory of Comput., 1983, pp. 258–263.

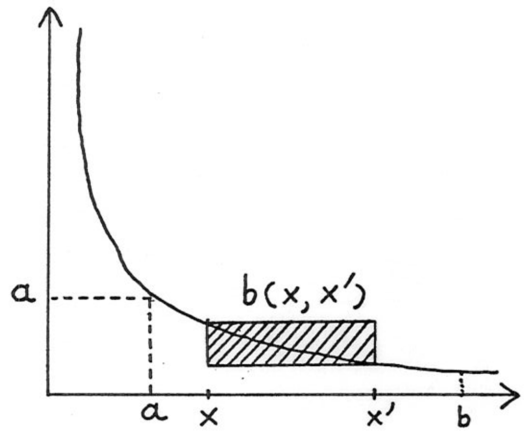

Figure 1

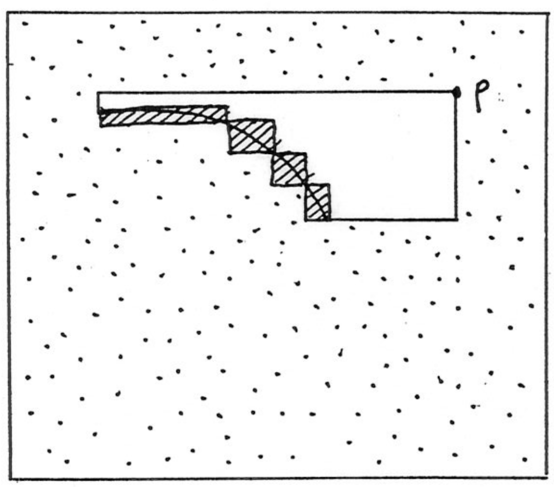

Figure 2

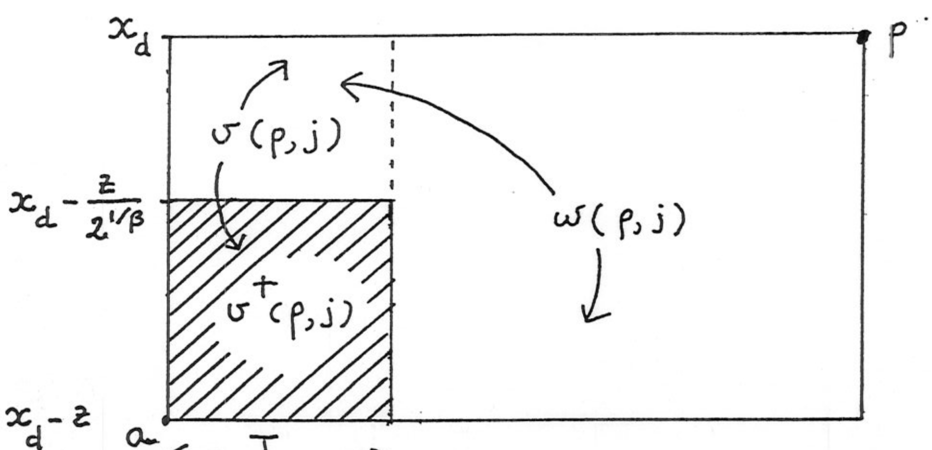

Figure 3